DESARROLLOS CIENTÍFICOS Y TECNOLÓGICOS EN LAS CIENCIAS COMPUTACIONALES

Gustavo Trinidad Rubín Linares

DESARROLLOS CIENTÍFICOS Y TECNOLÓGICOS EN LAS CIENCIAS COMPUTACIONALES

DESARROLLOS CIENTÍFICOS Y TECNOLÓGICOS EN LAS CIENCIAS COMPUTACIONALES

María del Carmen Santiago Díaz
Gustavo Trinidad Rubín Linares
María De Lourdes Sandoval Solís
Yeiny Romero Hernández
(Editores**)**

Gustavo Trinidad Rubín Linares
(Coordinador)

María del Carmen Santiago Díaz, Gustavo Trinidad Rubín Linares, María De Lourdes Sandoval Solís, Yeiny Romero Hernández
(editores BUAP)

Gustavo Trinidad Rubín Linares
(coordinador BUAP)

María del Carmen Santiago Díaz, Gustavo Trinidad Rubín Linares, Ana Claudia Zenteno Vázquez, Judith Pérez Marcial, Yeiny Romero Hernandez, José Luis Pérez Rendón, Nicolás Quiroz Hernández, Héctor David Ramírez Hernández, Gregorio Trinidad Garcia, María De Lourdes Sandoval Solís, Rogelio González Velázquez, Jose Luis Hernández Ameca, Armando Espíndola Pozos, María Blanca del Carmen Bermudez Juárez, José Martín Estrada Analco, Luz Del Carmen Reyes Garcés, Luis Enrique Colmenares Guillén, Elsa Chavira Martínez, Pedro García Juárez, Nelva Betzabel Espinoza Hernández, Roberto Contreras Juárez, Graciano Cruz Almanza, Maya Carrillo Ruiz, Bárbara Emma Sánchez Rinza, Beatriz Beltrán Martínez, Gabriel Juárez Díaz, José Andrés Vázquez Flores, Ivo Humberto Pineda Torres, Miguel Ángel Peña Azpiri, Hermes Moreno Álvarez, Raúl Antonio Aguilar Vera, Julio Cesar Diaz Mendoza, Francisco Marroquín Gutiérrez, Jéssica Nayeli López Espejel, Eden Belouadah
(revisores)

Primera edición: 2020
ISBN: 978-607-8728-36-7

Montiel & Soriano Editores S.A. de C.V.
15 sur 1103-6 Col. Santiago Puebla, Pue.

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA
Rector:
Dr. José Alfonso Esparza Ortiz

Secretario General:
Mtra. Guadalupe Grajales Y Porras

Vicerrector de Investigación y Estudios de Posgrado:
Dr. Ygnacio Martínez Laguna

Directora de la Facultad de Ciencias de la Computación:
M.I. María del Consuelo Molina García

Contenido

Prefacio

El avance vertiginoso de la ciencia y la tecnología han generado un gran abanico de soluciones a diversos problemas, sin embargo, en nuestra sociedad nos encontramos inmersos en un sistema que ya no brinda el soporte para una población que ha crecido de forma tan acelerada. Por ello es imprescindible resolver problemas propios de una sociedad en constante crecimiento, a fin de generar mejores condiciones de vida a nuestra población. Aunque hay identificados a nivel nacional y local los problemas que requieren solución inmediata, casi en cualquier área se requiere realizar innovaciones tecnológicas, algunas muy sofisticadas y complejas y otras no tanto, pero finalmente innovaciones, es decir, aplicar ideas y conceptos para solucionar problemas utilizando las ciencias computacionales, que aunque en muchos casos no se requiere una solución que implique años de investigación, si se requiere que la solución esté plenamente enfocada a un problema en particular. Muchas de estas soluciones en general no están a la vista sin embargo, resolver problemas ambientales, de vialidad, de producción de alimenticios, etc., representan en sí aplicar la tecnología de forma innovadora ya que aunque en nuestro país tenemos un enorme retraso tecnológico, lo que no se quiere en universidades e institutos de investigación es tener este retraso en la aplicación de la tecnología que se genera, por ello se cuenta con diversos programas internacionales, nacionales y locales para apoyar la innovación tecnológica. En nuestra universidad como en muchas otras en México y en el mundo se cuentan con programas específicos de innovación tecnológica para que laboratorios de investigación generen y apliquen tecnología propia. Pero estos esfuerzos no son suficientes, se requiere de una concientización colectiva para que desde el aula se socialice la necesidad de resolver toda clase de problemas aplicando el conocimiento impartido en clases, y no se debe esperar a una asignatura o catedra de emprendedurismo o innovación, sino desde cualquier tópico que se aborde en ciencias e ingeniería, ya que además de abstraer del mundo real el problema, se deben plantear las diversas alternativas de solución, las implicaciones tecnológicas para llevarlas a cabo, la importancia de la implementación, pero sobre todo, resolver el problema quizá por etapas o versiones hasta llegar a la solución óptima.

María del Carmen Santiago Díaz
Gustavo Trinidad Rubín Linares

Post-Quantum Cryptography for the Mexican Digital Invoices by Internet

Miguel Ángel León Chávez [1] and Francisco Rodríguez Henríquez[2]
[1] Facultad de Ciencias de la Computación
Benemérita Universidad Autónoma de Puebla, México
[2] Departamento de Computación
CINVESTAV - IPN
[1]mleon@cs.buap.mx, [2]francisco@cs.cinvestav.mx

Abstract This paper presents an analysis of the Post-Quantum Cryptography (PQC) digital signature algorithms accepted in the NIST third-round candidates. The digital signature is the core of the Mexican Digital Invoices by Internet (CFDI) and it is based on RSA digital signature but with the future arrival of Quantum Computers, it is vulnerable to cryptographic attacks. The paper points out the advantages and disadvantages for using the PQC digital signatures as the new core of CFDI.

Keywords: e-government; cryptography; post-quantum cryptography.

1 Introduction

El In 1976 [1] Whitfield Diffie and Martin Hellman (2015 Turing award winners) were the first to publicly expose the basic concepts of Public-Key Cryptography (PKC) and a key-exchange protocol.

In PKC, each user has two keys, one secret or private and the other publicly known by all the users in the system. Both keys are mathematically related in such a way that knowing both the public key and the encryption algorithm it is computationally infeasible to obtain the private key; but it is computationally easy to encrypt/decrypt the text using the keys, either of the two keys can be used for encryption while the other is used for decryption.

PKC is based on the computing problems difficult to solve, i.e., the algorithm solving the problem runs in a non-polynomial time. Two illustrious examples that have resisted the test of time are the integer factorization problem (IFP) and the discrete logarithm problem (DLP).

In 1977 [2] Ron Rivest, Adi Shamir and Len Adleman (2002 Turing award winners) proposed to use the exponentiation in a mathematical finite (Galois) field over integers modulo a large prime number; where the exponentiation is easy to compute but the integer factorization is very difficult to compute. Such proposal is known as the RSA Algorithm and it has been de facto standard of the PKC during the last four decades.

PKC together with Hash functions allow to implement the Digital Signature, which is the electronic analogous of the handwritten signature, where the signer agrees to respect a contract and his/her signature is valid for legal purposes.

A Hash function is a function that for any input text of arbitrary length produces an output of fixed bits length, known as the hash code or digest.

In 1994 Peter Shor [3] published two polynomial-time algorithms for solving both the IFP and the DLP on a quantum computer. Therefore, once that such quantum computers are available, the security of all applications based on these problems are vulnerable, e.g., the digital signatures based on IFP (RSA) and DLP (Digital Signature Algorithm (DSA) and Elliptic Curve DSA (ECDSA)).

In Mexico, the Secretary of Finance and Public Credit (SHCP) published in the official journal of the federation the Annex 20 of the fiscal miscellany resolution in the years

2006, 2010, 2012, 2014, and 2017 [4]. The Annex 20 is a computing standard that specifies the structure, form, syntax, format, and cryptographic of the digital invoices issued electronically.

The security requirements of the digital invoices by Internet (CFDI) are as follows: unforgeable, unique, protection against modification, protection against non-repudiation, and long-term storage. These requirements are met by means of two cryptographic digital signatures: one generated by the signer taxpayer, named digital stamp, and the other generated by the Tributary Administration System (SAT), named SAT's stamp.

SAT deploys a Public Key Infrastructure (PKI) in Mexico, where SAT is a Certification Authority (CA) that creates and signs two kinds of digital certificates for all the taxpayers in the country. One certificate for fiscal management purposes, named e-firma, and the other certificate for invoicing purposes, named digital stamp certificate. Both digital certificates are conformed to X.509 [5] standard and allow distributing the public keys. A taxpayer is responsible for both keeping secrete his/her private key and the long term storing of his/her issued digital invoices. SAT provides the software, named Certifica, in order to generate the taxpayer's private key and a requirements file that is verified at the SAT's office at the time of generation the digital certificates.

Annex 20 [4] specifies the usage of the following cryptographic functions for signing and verifying the CFDI:

- SHA-2 256, it is a Hash function producing an output of 256 bits length.
- RSAPrivateEncrypt, it is a function that uses the signer's taxpayer private key to encrypt the hash code of an invoice for generating the signature.
- RSAPublicDecrypt, it is a function that uses the signer's taxpayer public key to decrypt the signature of an invoice in order to verifying the signature.

Section III.A of the Annex 20 specifies the usage of RSA-2048 bits with the Hash function SHA2-256.

In [6,7,8] the authors have identified some security vulnerabilities of the Mexican digital invoices by Internet. And now the digital signature based on RSA is vulnerable to quantum computers.

On the other hand, in 2016 the National Institute of Standards and Technology (NIST) of the U.S. Department of Commerce published a request for nominations for Public-Key Post-Quantum Cryptographic (PQC) Algorithms [9] i.e., quantum-resistant and classical-resistant algorithms, that will be standardized and augmented to Digital Signature Standard (NIST FIPS 186-4) as well as recommendations for Pair-Wise Key Establishment Schemes using both DLP (NIST SP 800-56A) and IFP (NIST SP 800-56B).

In 2017, 82 candidate algorithms were submitted, among these 69 were accepted as the first-round candidates. In 2019 [10] 26 were accepted as the second-round candidates, where 17 candidates for public-key encryption and key-establishment algorithms, and 9 candidates for digital signatures.

In 2020, NIST published the status report on the second-round candidates and it announced the three third-round finalists as well as three alternate candidate algorithms for standardization at the end of the third round in the spring or summer od 2021 (NISTIR 8309).

The PQC digital signatures accepted in the third-round candidates are as follows: Crystals-Dilithium, Falcon (Fast-Fourier Lattice-based Compact Signature over NTRU), and Rainbow. In addition, GeMSS (Great Multivariate Short Signature), Picnic, and SPHINCS$^+$ will be considered as alternate candidates.

This paper analyses the PQC proposals accepted in the third-round candidates as digital signatures to generate the CFDI and points out their advantages and disadvantages. The rest of the paper is organized as follows: section 2 presents PQC digital signature proposals and their characteristics. Section 3 presents the performance of the PQC digital signatures proposals reported by the authors. Section 4 presents the performance reported by the project eBATS. Section 5 presents the results of NIST third-round candidates. Finally, the conclusions and the future research work are drawn.

2 PQC

Even if the IFP and the DLP are vulnerable to large quantum computers, there are other computing problems difficult to solve using quantum and classical computers, such as [11]: Secret-key, Hash-based, Lattice-based, Multivariate-polynomial, and Code-based, briefly described below.

Secret-key cryptographic, such as AES, is based on arithmetic operations on the Galois Field ($GF(2^8)$).

Hash-based cryptographic, such as SHA-2 and SHA-3, are based on the security properties of the Hash function, i.e., collision resistant, preimage resistant (one-way), and second preimage resistant.

Lattice-based cryptographic, such as NTRU, is based on the shortest vector problem with entails either approximating the minimal Euclidian length of a lattice vector or the short integer solution problem.

Multivariate-polynomial cryptographic, such as HFE^{v-}, is based on multivariate polynomial problem over finite fields.

Code-based cryptographic, such as McEliece's hidden Goppa-code, is based on error correcting codes.

These classes of cryptographic are the new cores of the NIST candidates for PQC digital signatures, as follows:

Crystals-Dilithium and Falcon are based on Lattice cryptographic.

GeMSS and Rainbow are based on Multivariate cryptographic.

Picnic, and $SPHINCS^+$ are based on Hash cryptographic.

2.1 PQC Security Strength

Instead of defining the strength of a submitted PQC algorithm using precise estimates of the number of bits of security. NIST defined a collection of security strength categories [9] as follows, listed in order of increasing strength:

1. Any attack that breaks the relevant security definition must require computational resources comparable to or greater than those required for key search on a block cipher with a 128-bit ley (e.g., AES128).
2. Any attack … for collision search on a 256-bit hash function (e.g., SHA256/SHA3-256).
3. Any attack … for key search on a block cipher with a 192-bit ley (e.g., AES192)
4. Any attack … for collision search on a 384-bit hash function (e.g., SHA384/SHA3-384).
5. Any attack … for key search on a block cipher with a 256-bit ley (e.g., AES256).

2.2 PQC Performance metrics

NIST is evaluating the submissions based on the following metrics [9]:

i) The sizes of the public keys, ciphertexts, and signatures that they produce.

ii) The computational efficiency (both on hardware and software) of the public keys (encryption, encapsulation, and signature verification) and private key (decryption, decapsulation, and signing) operations.

iii) The computational efficiency of their key generation operations.

iv) Decryption failures, some public-key encryption and key establishment mechanisms will occasionally produce ciphertext that cannot be decrypted/decapsulated, in such cases the submitters must provide the failure rate.

2.3 PQC Technical evaluation

NIST will test the submissions on the NIST PQC Reference Platform, an Intel x64 running Windows or Linux and supporting GCC compiler.

3 PQC Digital Signatures Performance

Table 1 summarizes the performance metrics of the candidates in terms of both the security strength and the length in Bytes (or kilo Bytes (kB) or bits) of the Public key (PK), Secret key (SK), and Signature (S) reported by the Authors.

3.1 Remarks

i) Crystals-Dilithium (2, 3, 4) provides the security strength 1, 2, and 3, respectively.

ii) Table 1 presents Falcon (512, 1024) dyn, according to the authors, the signature depends on whether (secret key) compression is used or not. Not using compression yields larger signatures, but with a fixed size. When using compression, signatures have a maximum theoretical size; the presented values correspond to the compressed and the average.

iii) The authors of Rainbow propose three variants, named classical, cyclic, and compressed. The cyclic Rainbow allows reducing the public key size of the classical scheme by up to 70% at a higher cost of signature verification. The compressed Rainbow stores the private key in the form of a 512 bit seed, thus enabling storing the private key easily on small devices at the cost of the efficiency of the signature generation process.

The only difference between the compressed Rainbow and the cyclic Rainbow is the use of the PRNG function during the signature process. Table 1 shows the parameter sets for the security level 1, 3, and 5, respectively.

iv) GeMMS (128, 192, 256) specifies the security strength 1, 2, and 3, respectively.

The authors proposed three sets of parameters, named: GeMSS, BlueGeMSS, and RedGeMSS. The first one corresponds to the parameters proposed for the first-round candidates, and the schemes Blue and Red were proposed for the second-round candidates.

v) The authors of the Picnic scheme propose the following three variants by varying the zero-knowledge protocol used and the transformation that is applied: Picnic-FS uses ZKB++ with the Fiat-Shamir transform, Picnic-UR uses ZKB++ with the Unruh transform, and Picnic2-FS uses a non-interactive zero-knowledge proof of knowledge with the Fiat-Shamir transform.

Table 1 shows these variants for the security strengths 1 (L1), 3 (L3), and 5 (L5), respectively. The presented values of the signature length are the maximum, but the authors provide also the average and the standard deviation computed over 100 signatures.

vi) The authors of SPHINCS$^+$ propose parameter sets achieving security levels 1, 3, and 5; for each of these levels they propose one size-optimized (ending on "s" for small) and one speed-optimized (ending on "f" for fast).

According to the authors the advantages and limitations of their proposal can be summarized in one sentence: On the one hand, it is probably the most conservative design of a post-quantum signature scheme, on the other hand it is rather inefficient in terms of signature size and speed.

3.2 PQC Computational efficiency

The computational efficiency is measured in terms of clock cycles or time in milliseconds of the key generation, signing, and signature verification processes reported by the authors. The key generation process takes into account the time to generate the key pair, a secret key and the corresponding public key.
Even if NIST tested the submissions on a PQC Reference Platform, each author measured the computational efficiency on his own platform, as follows:

1. Crystals-Dilithium (Skylake[1]) on an Intel Core-i7 6600U CPU at 2600 MHz using SHAKE as the XOF.
   The values of the cycles reported are the medians of 1000 execution each, and signing a text of 32 bytes length. The authors reported also optimized versions based on AVX2, and AVX2+AES.
2. Falcon (Skylake[2] core) on an Intel Core-i7 6567U CPU at 3.6 GHz with TurboBoost enabled [iacr, e-print: 2019-893].
   (ct) mean that constant-time hash-to-point and the fixed-size signature size are used, (fpemu) for generic integer emulation of floating-point operations.
   The authors do not indicate the length of the signing text.
3. Rainbow (Skylake[3]) on an Intel(R) Xeon(R) CPU E3-1225 v5 at 3.30 GHz.
   The reported values do not indicate the length of the signing text.

Table 1. Length in bytes (or kilo bytes (kB) or bits) of the Public Key (PK), Secret Key (SK), and Signature (S) reported by the Authors.

Digital Signature		AES 128	SHA3 256	AES 192	SHA3 384	AES 256
Crystals-Dilithium (2, 3, 4)	PK	1184	1472	1760		
	SK	2800	3504	3856		
	S	2044	2701	3366		
Falcon (512, 1024) dyn	PK	897				1793
	SK	1281				2305
	S	657.38				1273.31
Rainbow GF(16),32,32,32 GF(256),68,36,36 GF(256),92,48,48	PK (kB)	149		710.6		1705.5
	SK (kB)	93		511.4		1227.1
	S (bits)	512		1248		1632
Rainbow cyclic/ compressed GF(16),32,32,32 GF(256),68,36,36 GF(256),92,48,48	PK (kB)	58.1		206.7		491.9
	SK (kB)	93.0		511.4		1227.1
	S (bits)	512		1248		1632
GeMSS	PK	352.19		1237.9		3040.7

(128, 192, 256)	kB					
	SK (kB)	13.44		34.07		75.89
	S (bits)	258		411		576
BlueGeMSS (128, 192, 256)	PK kB	363.61		1264.1		3087.96
	SK (kB)	13.70		35.38		71.46
	S (bits)	270		423		588
RedGeMSS (128, 192, 256)	PK (kB)	375.21		1290.5		3135.59
	SK (kB)	13.10		34.79		71.89
	S (bits)	282		435		600
Picnic-FS (L1-L3-L5)	PK	32		48		64
	SK	16		24		32
	S	34032		76772		132856
Picnic-UR (L1-L3-L5)	PK	32		48		64
	SK	16		24		32
	S	53961		121845		209506
Picnic2-FS (L1-L3-L5)	PK	32		48		64
	SK	16		24		32
	S	13802		29750		54732
SPHINCS$^+$ s (small)	PK	32		48		64
	SK	64		96		128
	S	8080		17064		29792
SPHINCS$^+$ f (fast)	PK	32		48		64
	SK	64		96		128
	S	16976		35664		49216

4. GeMSS (Skylake[4]) LaptopS with an Intel Core-i7 6600U CPU at basic 2.6 GHz and max 3.4 GHz.
   The set of parameters of GeMSS is conservative in terms of security. The schemes Blue and Red are more aggressive in terms of security and more efficient regarding the signing timings, according to the authors. The reported values do not indicate the length of the signing text.
5. Picnic[5] on an Intel(R) Core(TM) i7-4790 CPU at 3.60 GHz, where:
   Picnic-FS: Picnic based on the Fiat-Shamir transform (FS)
   Picnic-UR: Picnic based on the Unruh transform (UR)
6. SPHINCS$^+$[6] on an Intel Core-i7 4770K CPU at 3.5 GHz.
   The authors report two implementations, one for the small (s) version, and the other for the fast (f) version. Both are implemented using Shake256 and SHA256, without indication of the length of the signing text.

   It can be noted that it is not possible to compare the computational efficiency of the proposals, in terms of cycles or time, since each was implemented in different CPUs at different frequencies. Fortunately, there exist the effort of multiple researchers to develop a fair evaluation, as it is discussed in the next section.

4 eBATS PQC Digital Signatures Performance

ECRYT Benchmarking of Asymmetric Systems (eBATS) [12] is a project to measure the performance of the public-key systems. It is based on the SUPERCOP's work, which is a toolkit for measuring the performance of cryptographic software.

With regard to the public-key signature systems it measures the cryptographic primitives according the following criteria:

Time (cycles) to generate a key pair (a secret key and the corresponding public key); time to sign a short message (59 bytes); time to open a signed short message, i.e., to verify a (larger) signed message and recover the original short message; space (bytes) for a secret key; space for a public key; space for a signature on a 0-bytes messages; space for a signature on a 23-bytes message, i.e., the signed-message length minus 23 bytes; and space for a signature on a long message, i.e., the signed-message length minus the message length.

eBATS has measured the PQC digital signature performance in both multiple CPU architectures and in multiple computers ranking from 1 up to 64 CPU cores at different frequencies.

We chose to present the PQC digital signature performance reported by eBATS in a slow computer using the Intel Core i3-2310M (2x2100 MHz) because most of the taxpayers in Mexico do not have high performance computers, nevertheless the performance analysis for servers with 64 CPU cores may be taken into account by the SAT.

Table 2 presents the computational efficiency of the first level of security (AES128) in terms of cycles (median) of the following processes: key generation, signing (59 bytes), and signature verification (verify) of 59 bytes reported by eBATS on an Intel Core i3-2310M (2x2100 MHz).

It can be noted in Table 2 that the rows list the median of many speed measurements. But measurements with large variance are indicated with question mark (?).

5 Conclusions

In the near future, the security of all applications based on the IFP and DLP problems are vulnerable to quantum computers, such as Mexican's CFDI.

This paper has presented a security analysis of the post-quantum digital algorithms of NIST third-round candidates based on both the performance reported by the authors and the reported by the project eBATS.

With regard to both Table 1 it can be noted the following:

i) Picnic and SPHINCS$^+$ provide the smallest secret keys in bytes

ii) Picnic and SPHINCS$^+$ provide the smallest public keys in bytes

Table 2. Computational Efficiency in Cycles (Median) of the Key Generation (Key gen), Sign (S), and Verify (V) processes of 59 bytes on an Intel Core i3-2310M (2x2100 MHz) reported by eBATS for the security level AES128.

Digital Signature (AES128)	Key Generation	Sign	Verify
Crystals-Dilithium	308,416	1,225,524?	342,748?
Falcon-512	26,292,540	1,276,148	114,816
Rainbow-binary GF(16)24,20,20	1,589,613,328	747,176	339,816
Rainbow6440 GF(31),26,20,20	6,292,390,620	2,767,088	1,948,804
Rainbow-binary GF(256),18,12,12	5,785,016,764	2,164,668	1,958,700

GeMSS	87,832,804	2,457,656,940?	281,952
Picnic-FS	13,228	10,644,980	8,951,196
Picnic-UR	13,200	12,841,472	10,914,748
Picnic2-FS	14,272	280,549,504	124,318,612
SPHINCS$^+$s SHA256	209,513,168	3,123,168,160	3,525,408
SPHINCS$^+$f SHA256	6,519,200	205,840,212	8,509,920

iii) Both GeMSS and Rainbow provide the smallest digital signatures in bits

With regard to both Table 2 it can be noted the following:

i) Both Picnic and Crystals-Dilithium have the best computing efficiency to compute the key generation process, however the signing process of Dilithium suffers of large variance.

ii) Rainbow binary has the best computing efficiency to sign and to verify 59 byes.

Crystals-Dilithium and Falcon are structured lattice schemes, from the point of view of NIST, these schemes appear to be the most promising general-purpose algorithms for public-key encryption and digital signature algorithms. Nevertheless, NIST believes it is prudent to continue to study other schemes against progress in cryptanalysis. Meanwhile, the computational efficiency of the finalist will be evaluated with regard to the Mexican's CFDI, as our future research work.

References

1. Diffie, W.; Hellman, M. E: New Directions in Cryptography, IEEE Transactions On Information Theory, Vol. IT-22, No. 6, pp. 644-654 (1976)
2. Rivest, R. L.; Shamir, A.; Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems, Communications of the ACM, Vol. 21, No. 2, pp. 120-126, (1978)
3. Shor, P.: Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, SIAM J. Computing, Vol. 26, No. 5, pp. 1484–1509 (1997)
4. Anexo 20 SHCP: Anexo 20 de la Resolución de la Miscelánea Fiscal para 2017, publicado el 23 de diciembre de 2016 en el diario oficial de la federación por el Poder Ejecutivo a través de la Secretaría de Hacienda y Crédito Público - Servicio de Administración Tributaria (2017)
5. ITU-T Recommendation X.509. Information technology – Open systems interconnection – The Directory: Public-key and attribute certificate frameworks (2008)
6. González-García, V.; Rodríguez-Henríquez, F.; Cruz-Cortés, N.: On the security of Mexican Digital Fiscal Documents. In Computación y Sistemas. Vol.12, No. 1, pp. 25-39 (2008)
7. León-Chávez, M.A.; Rodríguez-Henríquez, F.: Security Vulnerabilities of the Mexican Digital Invoices by Internet. In IEEE International Conference on Computing Systems and Telematics, Octubre, Xalapa, Ver. (2015)
8. Domínguez-Pérez, L.J.; Gómez-Trujillo, L.M.; Cruz-Cortés, N.; Rodríguez-Henríquez, F.: Sobre el impacto del colisionador SHA-1 en las firmas digitales mexicanas con valor legal. Computación y Sistemas, Vol. 23, No. 4, pp. 1181-1190 (2019)
9. NIST. Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process (2016)
10. NISTIR 8240. Status Report of the First Round of the NIST Post-Quantum Cryptography Standardization Process (2019)
11. Bernstein, D.; Buchmann, J.; Dahmen, E. (Ed.): Post-Quantum Cryptography. Springer, (2009)
12. eBATS (ECRYT Benchmarking of Asymmetric Systems). Bernstein, D. J.; Lange, T: (Ed). eBACS: ECRYPT Benchmarking of Cryptographic Systems. https://bench.cr.yp.to, accessed May 21, 2020

Planificación de Trayectorias en Robots Cooperativos Mediante Visión Artificial.

Abel A. Rubín-Alvarado, José E. M. Gutiérrez-Arias, Fernando Reyes-Cortés, Gregoria Corona-Morales.
Facultad de Ciencias de la Electrónica, Benemérita Universidad Autónoma de Puebla.
CU, Boulevard 18 Sur, Av. San Claudio y Jardines de San Manuel, 72570 Puebla, Pue., México.
abel.rubin@alumno.buap.mx,-arigutmses5@gmail.com,
fernando.reyes@correo.buap.mx,gcoronamorales@gmail.com

Resumen. Al obtener un modelo matemático de cualquier sistema físico, es muy deseable poder comprobarlo antes de realizar su implementación, para ello se recurre a varios métodos, uno de ellos es la simulación. En este trabajo presentamos una metodología denominada SimMechanics, para el sistema de un robot móvil tipo diferencial, realizando una simulación más aproximada a la realidad, validando así el modelo. La planificación de la trayectoria se implementa únicamente en el robot líder con un controlador PID aplicado en la velocidad de cada uno de los motores, un segundo robot dotado de visión artificial denominado esclavo, que mantendrá identificado al robot líder en todo momento y manteniendo una distancia determinada robot líder, para lograr que la trayectoria sea recorrida por ambos robots, líder-esclavo de forma cooperativa.

Palabras Clave: CAD, Modelo Matemático, Simulación, control PID, Visión artificial, Trabajo colaborativo.

1. Introducción

La robótica cada día resuelve nuevas y fascinantes tareas para diversos sistemas, con el único afán de evitar que las personas continúen realizando actividades; de alto riesgo, de difícil acceso como en zonas de desastres en donde, el reconocimiento, búsqueda y rescate son cruciales.

A finales del siglo pasado, la industria emplea robots que pueden ser operados y/o supervisados por una persona logrando así mejoras en producción y reducción de costos, cumpliendo estas tareas los investigadores se enfocan en nuevas y más complicadas actividades a resolver, al punto de ya no ser necesaria la intervención humana para su operación, al contrario hemos ido conceptualizando la idea de tener robots de servicio, logrando así, que muchas de las tareas que son complejas para un solo robot se logren cumplir con la operación de varios de ellos.

Los robots móviles tipo diferencial han sido objeto de estudio desde hace varios años, debido a su configuración de dos ruedas independientes activas y una tercera pasiva, siendo muy recurrentes en las instalaciones industriales, para el transporte de mercancía, exploración, en el hogar y lugares como hospitales [1]. Su investigación considerando el modelo cinemático en conjunto con otro robot para lograr una tarea colaborativa, muchos de los trabajos antes de su implementación efectúa una simulación en computadora con la ayuda de software especializado, una de los objetivos de este trabajo es presentar un método denominado SimMechanics el cual proporciona un ambiente de simulación en 3D donde son incluidas todas las masas, inercias, articulaciones, restricciones y geometría 3D. Una animación 3D generada automáticamente permite visualizar la dinámica del sistema.

2. Antecedentes

El estudio y uso de los modelos [2] en donde su investigación realizan una aproximación de manera numérica al modelo dinámico llevando su implementación en un robot tipo no-holónomico, el móvil es comercial llamado PIONNER 3DX, y mejorar los algoritmos de control que se emplean para hacerlos más precisos, e introducir las ventajas de software especializado [3], quienes utilizan algoritmos RRT por sus siglas en inglés (Rapidly Exploring Random Trees) para trazar rutas, logrando que un robot no-holónomico sea capaz de evitar obstáculos en un ambiente estructurado con solo considerar punto de partida y el punto final, la implementación se llevó a cabo en un robot comercial modificado PIONNER P3-AT, el empleo de algoritmos y la posibilidad de inteligencia artificial o lograr que el robot aprenda son ideas que inician, debido a que estos algoritmos tienen otros enfoques [4], basados en IA por sus siglas (Inteligencia Artificial) emplean multi-agentes logrando así, la evasión de obstáculos para robots en configuración líder-seguidor, empleando una cámara instalada en el robot seguidor y aplicando un algoritmo llamado (BUG2), que en general logra llegar al punto final y en caso de encontrarse un obstáculo, el móvil rodea hasta volver a interceptarse con la línea que inicialmente lo llevaría al punto final, la implementación la realizan en un robot LEGO.

3. El modelado matemático.

Se obtiene un modelo que describe el movimiento del robot móvil en una configuración diferencial correspondiente para la implementación del robot líder, se exporta de SolidWorks a MATLAB obteniendo un modelo a bloques dentro de un ambiente 3D para una simulación que permita visualizar su comportamiento, la implementación del robot líder únicamente cuenta con los sensores tipo encoders en cuadratura de los motores de DC, que son controlados por una estrategia de control PID para las velocidades logrando que una trayectoria definida se programe únicamente en el robot denominado líder, un segundo robot móvil denominado esclavo, contará con un sistema de visión artificial implementado en una tarjeta de desarrollo raspberry pi 3, su propósito será identificar al robot líder y mantenerse a una distancia predeterminada mientras el robot líder recorre la trayectoria que le fue programada.

3.1 Modelo Matemático de un Robot Móvil.

Se considera al robot líder con la siguiente configuración [5] como se muestra en la figura 1, el cual está constituido por dos ruedas activas alineadas por un eje, cada rueda cuenta con un motor de corriente directa y giro independiente, una tercera rueda omnidireccional que asegura que la plataforma del robot se encuentre paralela al plano XY, para efectuar el análisis correspondiente se definen los siguientes parámetros:

- c - punto medio del eje que une las dos ruedas traseras del robot líder.
- s - centro de masa de la plataforma del robot líder.
- a - longitud de c al centro de cualquiera de las dos ruedas activas; izquierda y derecha.
- b - longitud de c al centro de masa de la plataforma del robot líder.
- V - magnitud del vector velocidad del punto c, respecto al sistema
- α - orientación del robot líder (ángulo formado por el eje X y el vector velocidad del punto c.

- v_l y v_r - velocidades del centro de masa de las ruedas izquierda y derecha respectivamente.
- R - Radio de las ruedas activas.
- φ_l y φ_r - parámetros que miden la rotación de las rueda derecha e izquierda respectivamente.

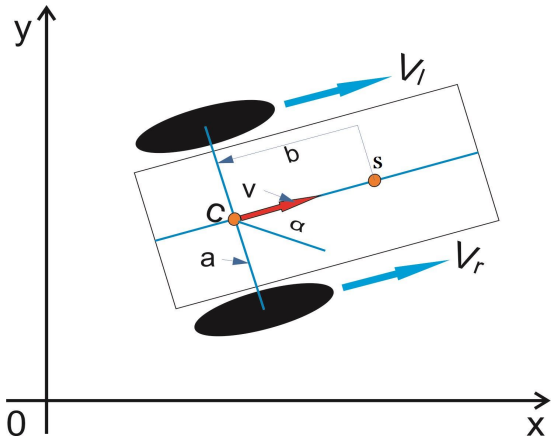


Fig. 1. Configuración del Robot líder desplazándose en el área de trabajo.

Una vez definidos estos elementos serán considerados los siguientes conceptos como los **Supuestos fundamentales del modelo**:

1. El robot líder se compone de partes rígidas (cada punto del robot tiene una posición y orientación fija con respecto al sistema relativo al robot líder).
2. No hay deslizamiento en la dirección perpendicular a la rodadura.
3. No hay deslizamiento traslacional entre la rueda y el suelo.
4. El centro de masas del robot líder puede no coincidir con el centro geométrico de éste.
5. El espacio donde se mueve el robot líder es perfectamente plano y sin inclinaciones.

Por lo que la posición del robot líder se puede ver como la proyección del punto **s** en el plano X,Y, obteniendo así la velocidad dando origen al modelo que nos describe su movimiento [5,6,7], el modelo cinemático se encuentra dado por la ecuación (1):

$$\begin{aligned} \dot{x}_s &= V\cos(\alpha) - b\Omega sen(\alpha) \\ \dot{y}_s &= Vsen(\alpha) + b\Omega\cos(\alpha) \\ \dot{\alpha} &= \Omega \end{aligned} \tag{1}$$

Considerando el supuesto fundamental planteado anteriormente, donde no existe deslizamiento traslacional ni perpendicular, con esto se presentan condiciones que relacionan la velocidad que presenta el centro de masas **cm** de cada una de las ruedas y, la velocidad angular de rotación de ellas, siendo representadas por la ecuación (2):

$$v_{cm} = R\dot{\varphi} \tag{2}$$

Por lo que, considerando la ecuación (2), las velocidades de las ruedas derecha e izquierda quedan expresadas por:

$$\begin{aligned} v_r &= R\dot{\varphi}_r \\ v_l &= R\dot{\varphi}_l \end{aligned} \tag{3}$$

Así que, al obtener las velocidades de cada una de las ruedas, es posible calcular la velocidad lineal del robot líder.

3.2 SimMechanics.

El diseño de CAD es de gran ayuda debido a todas las características y propiedades que son incluidas en él mismo, tales como; las masas de cada pieza, las inercias, las restricciones de cada articulación, etc. Además de tener una fiel reproducción del sistema a la hora de realizar una simulación que nos muestre su comportamiento, es decir su posición, orientación parámetros de velocidad de las ruedas.

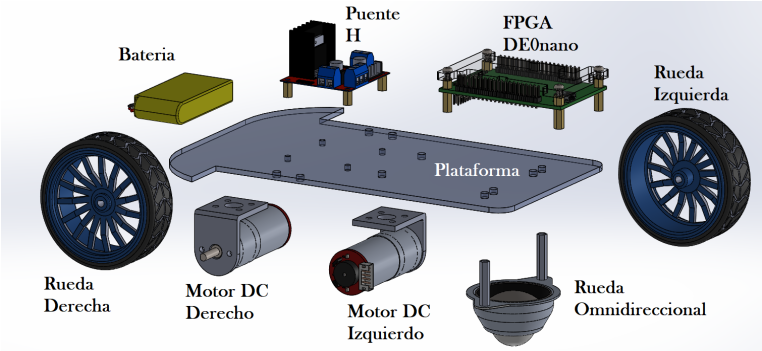


Fig. 2. Piezas diseñadas en CAD para el ensamble del robot líder.

Se diseñó el robot móvil en la herramienta de SolidWorks 2018, donde se ensamblan todas las partes que involucran un robot móvil tipo diferencial en su implementación física que será para el líder, como se muestra en la figura 2.

Llevando a cabo el ensamble correspondiente, obteniendo una réplica más cercana a la realidad, con la herramienta de SimMechanics se exporta el diseño generando los archivos necesarios para que el software de Matlab pueda interpretarlos y se visualicen desde la herramienta de Simulink como un modelo a bloques como lo muestra la figura 3 y una animación de cada una de las piezas ensambladas interactuando con las características que lo engloban: pesos, inercias, restricciones, dimensiones, etcétera.

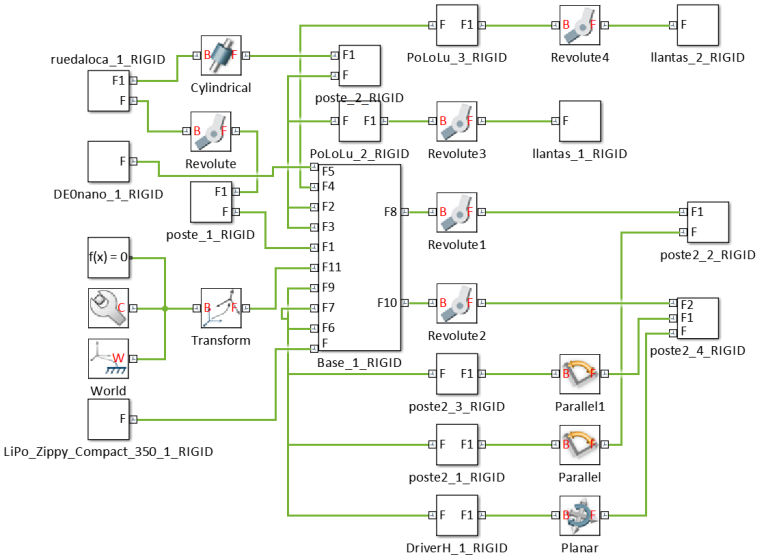


Fig. 3. Modelo generado por la importación del sistema desde SolidWorks con SimMechanics.

Se ingresan los valores de gravedad para simular el ambiente real, en la simulación que nos proporciona la herramienta de SimMechanics, permite visualizar en un ambiente 3D una simulación del comportamiento del sistema como se muestra en la figura 4. El modelo matemático obtenido y representado por la ecuación 1, se encuentra dentro del bloque Kinematics Subsystem que se muestra en la figura 5 y el Planar Joint, necesarios para indicar la variable a censar que en nuestro caso son las velocidades angulares de los motores DC, agrupando los bloques que corresponden a las ruedas derecha, izquierda y la rueda omnidireccional para tenerlos bien identificados.

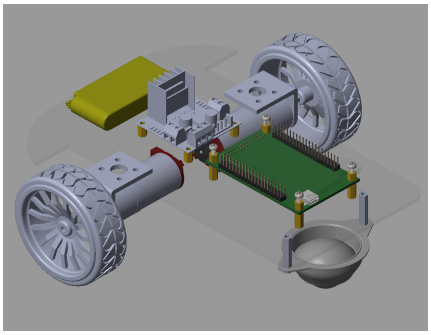


Fig. 4. Simulación del robot líder con SimMechanics.

Ahora podemos variar las entradas de los bloques rIzq y rDer que son las velocidades angulares correspondientes a los motores de DC conectados a las llantas, y validar su comportamiento.

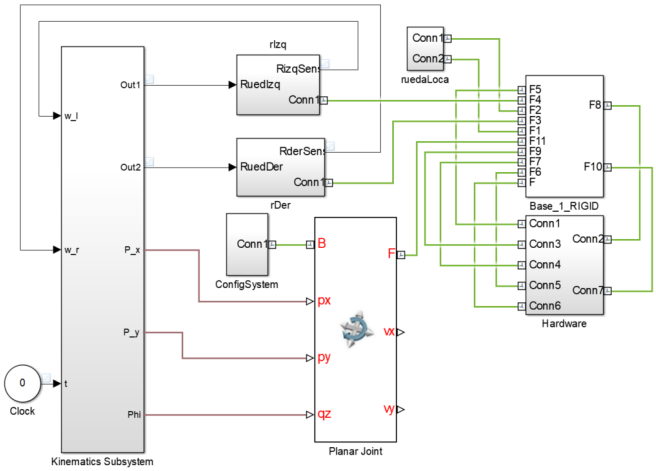


Fig. 5. Modelo a bloques del Robot Líder.

Este sistema final nos permite realizar una simulación fiel del comportamiento al llevar a cabo una implementación física y programar el modelo cinemático del robot móvil, para continuar con la implementación de los sistemas de control del robot líder.

3.4 Diseño del Control para el Robot Líder.

El controlador PID tiene como entrada, la velocidad angular de los motores de DC, en cada uno de los bloques que corresponden a las ruedas existe un sensor que otorga la información correspondiente para la retroalimentación del sistema, y así recorrer la trayectoria deseada.

La sintonización del controlador PID se realiza con el toolbox del mismo, para ambas ruedas con la información de la trayectoria planificada, mientras sea una línea recta las velocidades angulares φ_l, φ_r son de la misma magnitud en voltaje, y cuando existe una línea semicircular estas deberán de ser $\varphi_l > \varphi_r$, o $\varphi_l < \varphi_r$, de acuerdo al sentido horario o antihorario de la misma.

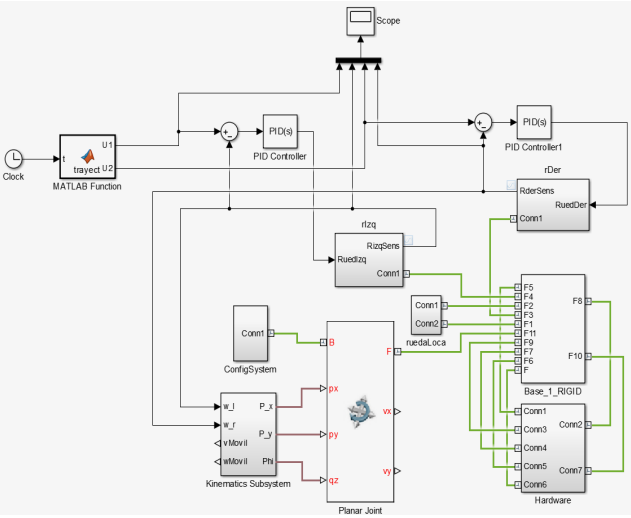


Fig. 6. Controladores PID para la velocidad angular de los motores del robot líder.

Con la trayectoria deseada programada en el robot líder, y los controles PID actuando, se observa cómo recorre la trayectoria, mientras que el segundo robot solo cuenta con la programación para la visión artificial con el fin localizar un objeto de color especifico y mantenerse a una distancia determinada, logrando así que cualquier trayectoria que sea definida en el robot lider, tambien la seguira el segundo robot, logrando una trayectoria en conjunto de manera colaborativa.

4. Robot Esclavo

Para el segundo robot emplearemos la misma configuración diferencial agregando una camara para dotarlo de visión, la tarjeta de desarrollo donde se implementa es una Raspberry Pi 3, se programa en Python bajo el ambiente de Raspbian, el objetivo del algoritmo que rige al robot esclavo esencialmente es para seguir un objeto de color definido y forma predeterminada, que en nuestro caso será una esfera de color rojo, que tendrá en su parte posterior el robot líder.

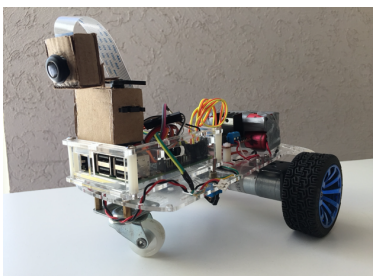


Fig. 7. Robot seguidor implementación en Raspberry pi 3.

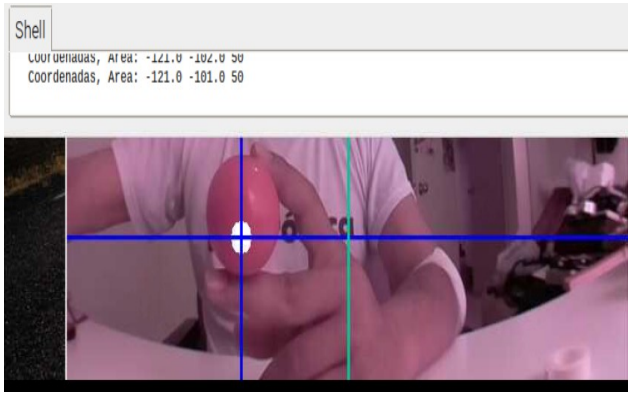


Fig. 8. Sistema de visión artificial, robot seguidor.

Para localizar el objeto se implementa el siguiente algoritmo.
Algoritmo 1. Detectando el objeto por color y se calcula su posición en el centro de la figura.

```
  while True:
      _, frame = cap.read()
      hsv_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
     # red color
      low_red = np.array([161, 155, 84])
      high_red = np.array([179, 255, 255])
      red_mask = cv2.inRange(hsv_frame, low_red, high_red)
      _, contours, _ = cv2.findContours(red_mask, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
      contours = sorted(contours, key=lambda x:cv2.contourArea(x),
reverse=True)
  for cnt in contours:
          (x, y, w, h) = cv2.boundingRect(cnt)
          x_medium = int((x + x + w) / 2)
          break
      cv2.line(frame, (x_medium, 0), (x_medium, 480), (0, 255, 0),
2)
```

De esta manera sabremos su ubicación con respecto al robot líder, y obtener las coordenadas en un plano X,Y, dependiendo de la distancia a la que se encuentre el robot líder, el área del objeto cambiará y el control que se implementa en el robot esclavo es con la finalidad de permanecer a una distancia y orientación específica detrás del robot líder, controlando las velocidades de los motores para avanzar o girar, con el único propósito de mantener el objeto de color rojo en el centro de su visión, asegurando asi se encuentre justo detrás del robot líder quien seguirá la trayectoria programada.

5. Resultados.

Si consideramos las velocidades del los motores del robot líder iguales, es decir $\varphi_l = \varphi_r = n$, donde n=1rad/s, se describe que el robot trace una línea recta, como se observa en la gráfica correspondiente a la simulación en la figura 9, con unas condiciones iniciales $x(0), y(0), \alpha(0) = (0,0,0)$ con t=30. Para que el robot líder vaya en una trayectoria semicircular o un círculo, ya sea en sentido horario o antihorario las velocidades angulares de los motores, deben de ser diferentes, en el mismo sentido, por ejemplo: para una trayectoria semicircular en sentido horario $\varphi_l > \varphi_r$, como lo demuestra su simulación en la figura 10.

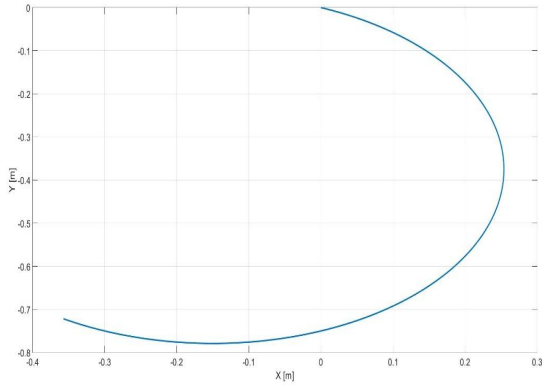


Fig. 9. Trayectoria con velocidades angulares iguales.

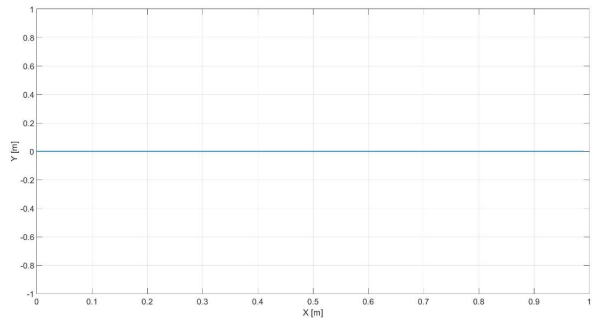


Fig. 10. Trayectoria con velocidades $\varphi_l > \varphi_r$.

Y de igual manera para que el robot líder siga una trayectoria semicircular en sentido antihorario, las velocidades angulares de los motores deben de ser $\varphi_l < \varphi_r$ como se muestra en la figura 11 correspondiente a la simulación.

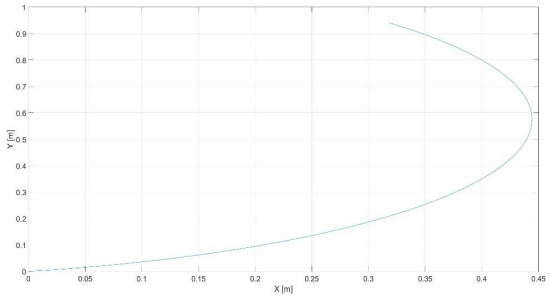


Fig. 11. Trayectoria con velocidades $\varphi_l < \varphi_r$.

Para llevar a cabo un trabajo colaborativo entre robots móviles, en este caso que sigan una trayectoria deseada, que involucre giros y líneas rectas, con las pruebas que se llevaron a cabo con SimMechanics, se implementó en el robot líder un controlador PID para las velocidades de los motores de DC, con el objetivo de regular el voltaje en cada uno de ellos, y lograr que el robot líder gire, o vaya en línea recta. En la figura 12 se muestra la trayectoria recorrida por el robot líder,

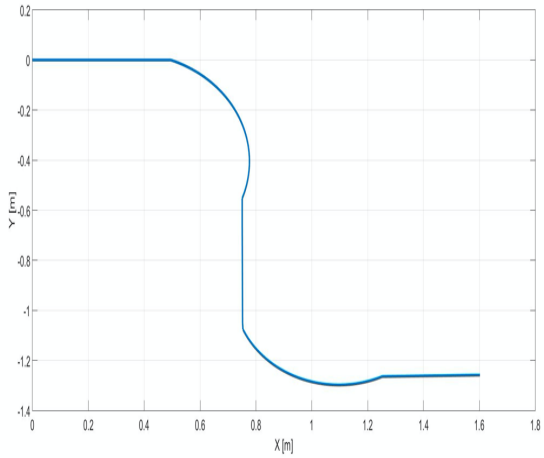


Fig. 12. Trayectoria recorrida por el robot líder.

Mientras el robot líder recorre la trayectoria definida, el robot esclavo mantiene identificado al robot líder por medio de la visión artificial proporcionada por la cámara como se muestra en la figura 13, se ingresa al sistema de raspbian que rigue al robot esclavo por medio de un acceso remoto VNC viewer, al inicializar el algoritmo para localizar al objeto de color rojo, y por medio del calculo del area del objeto este pueda mantenerse a una distancia definida avanzando, o para realizar un giro debido a que el objetivo se encuentra lejos del eje central de visión, ya sea derecha o izquierda respectivamente, como se muestra a continuacion.

Al identificar la esfera de color roja en la parte posterior del robot líder, las coordenadas asi como el área estimada, con esta información el robot seguidor avanza, gira hacia un sentido o hacia el opuesto, sin conocer la trayectoria.

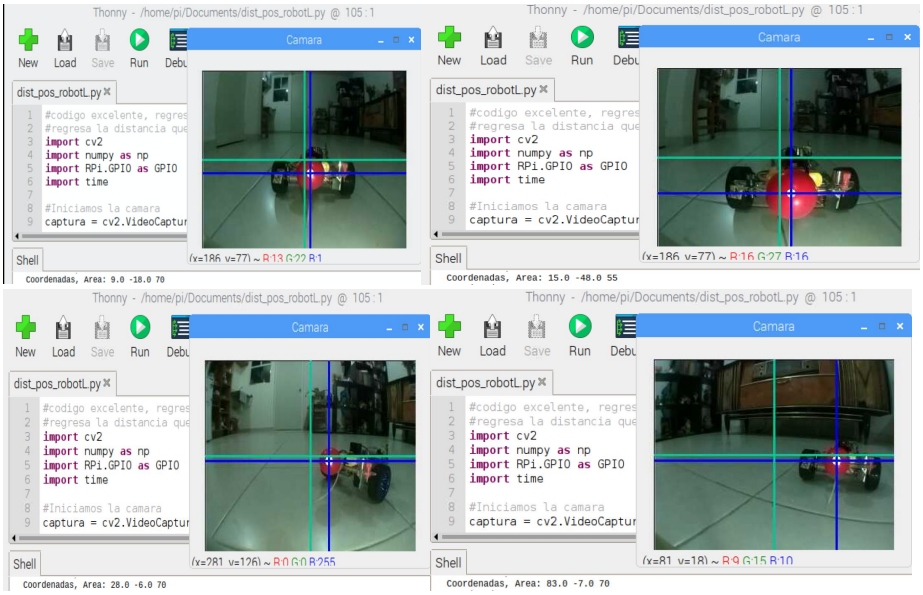


Fig. 13. Sistema de visión robot seguidor.

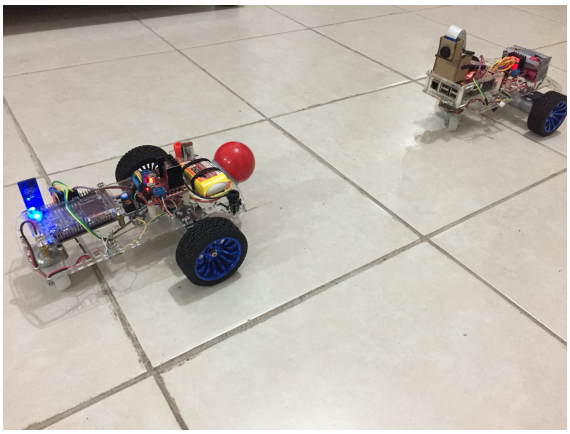


Fig. 14. Robots cooperativos para realizar una trayectoria por medio de visión artificial.

6. Conclusiones.

La herramienta SimMechanics ofrece además un ambiente en 3D donde se puede observar el comportamiento que tendría el sistema real, haciendo posible generar diversas pruebas y corregir algún inconveniente que se pueda presentar antes de las pruebas físicas.

Este trabajo nos permitió validar la opción de un sistema de visión artificial implementado en un robot móvil tipo diferencial, que solo identificando un objeto sin la necesidad de programar la trayectoria o conocerla, es capaz de recorrer la misma ruta que el robot líder.

La visión artificial nos permite controlar el movimiento de los motores de DC de acuerdo a la posición en la que se encuentra el objeto a identificar, esta puede identificar diversas formas, colores, incluso más de una al mismo tiempo, siendo muy versátil y de gran ayuda para que más robots esclavos se unan a una formación siguiendo la trayectoria del líder.

La estrategia de control si bien es de las más empleadas con los resultados obtenidos es posible implementar un control óptimo para el robot líder, para esto será necesario obtener el modelo dinámico.

7. Agradecimientos

Los autores agradecen a la Facultad de Ciencias de la Electrónica, Maestría en Ciencias de la Electrónica opción en Automatización. Al Consejo Nacional de Ciencia y Tecnología (Conacyt) por el apoyo brindado para esta investigación. De igual forma, agradecemos al Congreso Nacional de la Ciencias de la Computación (CONACIC) 2020, por la oportunidad de presentar este proyecto.

Referencias

1. Soria, C., Carelli, R., Kelly, R., & Zannatha, J. M. I. (2004). Control de Robots Cooperativos por Medio de Vision Artificial. In XVI Congreso de La Asociación Chilena de Control Automático (Vol. 223).
2. Rosales, A., Scaglia, G., Mut, V., & Di Sciascio, F. (2008). Control dinámico mediante métodos numéricos para robots móviles tipo uniciclo. Universidad Nacional de San Juan, Argentina..
3. Asqui Paguay, L. M. (2017). Planificación y seguimiento de caminos de manera autónoma para robots móviles tipo uniciclo. Master's thesis, Escuela Superior Politécnica de Chimborazo, 2017.
4. Cervera Andes, A. (2011). Coordinación y control de robots móviles basado en agentes. Universidad Politécnica de Valencia. PhD Thesis.
5. Corona Morales, G. (2016). Modelación matemática de un robot móvil y la síntesis de control óptimo. Benemérita Universidad Autónoma de Puebla. PhD Thesis.
6. Moisés, G. A. J. E., María, R. E. R., Castillo, M., & Montserrat, M. (2013). Modelado y Planificación de las Trayectorias de un Robot Limpiador. Personal de la Revista, 21.
7. Andaluz, G., Andaluz, V., & Rosales, A. (2011). Modelación, Identificación y Control de robots móviles. Escuela Politécnica Nacional, Quito, Tesis de Ingeniería en Electrónica y Control.
8. Zambrano, J. A. L., & Velandia, M. A. R. Artificial vision and communication in omnidirectional cooperative robots.
9. Sanchez, R. E. A., Pinzón, L. D. J., & Luna, J. A. G. (2012). Finding path through the algorithms: genetic and dijkstra using maps of visibility. *Scientia et technica*, *2*(51), 107-112.
10. Bermúdez-Bohórquez, G. R., Mancipe-Bernal, C. A., & Ortiz-Velásquez, J. E. (2017). Control of a robotic convoy through path planning and orientation strategies. *Revista científica*, *3*(30), 237-251.
11. Lin, C. H., Song, K. T., & Anderson, G. T. (2005, October). Agent-based robot control design for multi-robot cooperation. In *2005 IEEE International Conference on Systems, Man and Cybernetics* (Vol. 1, pp. 542-547). IEEE.
12. Cao, K. C., Jiang, B., & Yue, D. (2017). Cooperative path following control of multiple nonholonomic mobile robots. *ISA transactions*, *71*, 161-169.

Propuesta de Algoritmo Basado en Machine Learning para Enseñanza y Perfeccionamiento de Escritura

Rodrigo Bazán Zurita, María del Carmen Santiago Díaz, Gustavo Trinidad Rubín Linares, Ana Claudia Zenteno Vázquez, Yeiny Romero Hernández
Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, 14 sur Avenida San Claudio, Jardines de San Manuel C.P 72570 Puebla, México
rodrigo.bazanz@alumno.buap.mx, {marycarmen.santiago, gustavo.rubin, ana.zenteno, yeiny.romero}@correo.buap.mx

Resumen. Se propone el desarrollo de un algoritmo, usando IA que permita identificar la escritura correcta en niños entre 6 y 12 años, disminuyendo la generación de papel empleado para la práctica y enseñanza de escritura mediante el uso de tecnología en una tableta para recibir repeticiones de números. Se analizarán los dígitos y el algoritmo los clasificará en dos grupos: los escritos de manera errónea y los escritos de manera correcta.
Palabras clave: Inteligencia artificial, Machine Learning, Práctica de Escritura, Impacto Ambiental.

1 Introducción

La inteligencia artificial es un campo científico con el propósito de generar que una computadora sea capaz de replicar comportamientos humanos con las mismas habilidades que las personas emplean día a día en actividades sencillas como reconocer personas por la voz o por alguna característica peculiar. Al nacimiento de la inteligencia artificial en la década de los 50's se tenían altas expectativas de lo que se podía lograr; a través de la experimentación se dio origen a un algoritmo de clasificación mediante regresión lineal, después se crearon múltiples algoritmos con regresión polinomial capaces de predecir datos como la valuación de casas.

Uno de los avances que más impactaron a la sociedad fue el enfrentamiento entre el campeón de ajedrez Gary Kaspárov y la computadora "Deep blue" de IBM, la cual por medio de programación de inteligencia artificial, venció a Gary. Con el avance del tiempo y el estudio en esta rama, se han desarrollado herramientas con este tipo de algoritmos para facilitar la realización de acciones de la vida cotidiana, como predicciones de tiempo de traslado a algún destino o el asistente personal de Google, Google Assistant, que es capaz de entender órdenes mediante voz y ejecutarlas con una combinación de múltiples algoritmos entrenados mediante Machine Learning y Deep Learning, éstas son dos formas distintas de inteligencia artificial las cuales trabajan a partir de datos recabados acerca del usuario, por ejemplo, el desbloqueo de algún smartphone por reconocimiento facial, o la clasificación de fotos entre animales, personas, paisajes y otros objetos que se reconocen en las fotografías. Uno de los primeros experimentos de la inteligencia artificial fue la clasificación de dígitos escritos por seres humanos en década de los sesenta y sigue vigente en pleno siglo XXI, sin importar el tiempo transcurrido desde su generación; en el artículo "Sistemas de Reconocimiento de Caracteres para la lectura automática de Cheques" se describe la manera en la que se emplea la tecnología de reconocimiento de dígitos escritos para poder analizar de manera correcta un cheque [12].

El entorno de los ecosistemas se encuentra en riesgo constante por la deforestación, ya que la producción de pulpa de papel es la causa principal de la tala de árboles [1] en consecuencia, los incendios forestales registrados desde enero hasta agosto aumentaron un 13 % en comparación con los incendios totales de 2019 [15]. En busca de disminuir el uso del plástico en la vida cotidiana se propuso la sustitución de las bolsas de plástico por bolsas de papel, sin embargo, el plástico produce un menor daño para el medioambiente

que el papel [10]. Con el objetivo de satisfacer la demanda de papel se generan incendios forestales para facilitar la obtención de la pulpa de papel [6], lo cual, genera que se disminuyan los miembros de ecosistemas que evitan la transmisión de enfermedades entre humanos y fauna [5]. A partir de esto, se han iniciado investigaciones para ayudar al cuidado del medio ambiente haciendo uso de la tecnología, como la sustitución de tickets de papel por tickets enviados vía correo electrónico [7], también se ha propuesto una alternativa a la obtención de la pulpa del papel adquirida de árboles por la que es producida por el cáñamo, además el crecimiento de esta planta se da en un tiempo aproximado de 20 semanas, mientras que el de los árboles es de 20 años [12].

En un estudio escrito en 2014 se indicaba que la era de la tecnología en las aulas de las instituciones educativas estaba más cerca de lo que se pensaba [9]. La enfermedad COVID-19 orilló a la sociedad a iniciar un periodo de cuarentena para reducir contagios, los alumnos dejaron de asistir a la escuela provocando que el proceso de aprendizaje se realice a distancia entre docentes y alumnos a través del uso de la tecnología.

Con el objetivo de reducir el impacto ambiental originado por la necesidad de creación de libretas de práctica para aprendizaje y avanzando en la manera de enseñanza, se propone un algoritmo basado en la repetición de la metodología de aprendizaje con mayor aceptación empleada en los individuos de edades entre 4 y 12 años [14]. Otra investigación afirma la importancia de actividades pedagógicas y una de ellas plantea la relevancia que tiene alentar y animar al individuo para mejorar sus resultados, en los cuales se ve reflejado el aprendizaje [4]. Por esto, al finalizar el proceso del algoritmo, se obtiene un formato impreso en el cual se establece el porcentaje de éxito en la práctica con el objetivo de incitar la mejora de los resultados obtenidos para lograr una escritura excelente.

2 Metodología

El algoritmo tiene como base una red neuronal artificial densamente conectada, la característica principal de ésta es la identificación de dígitos escritos por humanos. La red neuronal se programó empleando Machine Learning, para entrenarla, es necesario tener una base de datos, en este caso se emplea la "MNIST", ya que contiene un total de 70,000 imágenes de dígitos escritos por humanos con una etiqueta que indica el número que representa cada una de estas imágenes.

La base de datos se descompone en 60,000 imágenes de entrenamiento y 10,000 de validación. Estas imágenes entrenaron a la red neuronal y validaron la eficacia de clasificación; después el algoritmo se encarga de analizar los dígitos obtenidos mediante una tableta para poder ingresar la información necesaria a la red con el objetivo de calificar el ejercicio e imprimir los números que fueron reconocidos de manera correcta.

2.1 Infraestructura de una red neuronal.

La base del algoritmo es una red neuronal densamente conectada, se emplean varias capas de neuronas artificiales para poder analizar los dígitos que serán ingresados por los usuarios.

Fue en 1943 cuando se publicó un artículo acerca del primer modelo de una neurona artificial en el que se establece una relación del modelo de la neurona artificial con una compuerta lógica puesto que el tipo de respuesta era binario [13]. Su funcionamiento se da al ingresar información en la capa de entrada donde se multiplica por pesos para ponderar la información, acto seguido ingresa la información ponderada a la neurona donde se realiza una sumatoria, el resultado de dicha operación, la entrada neta, pasa a la capa de salida donde según un umbral arroja un resultado. (Vea la Fig. 1)

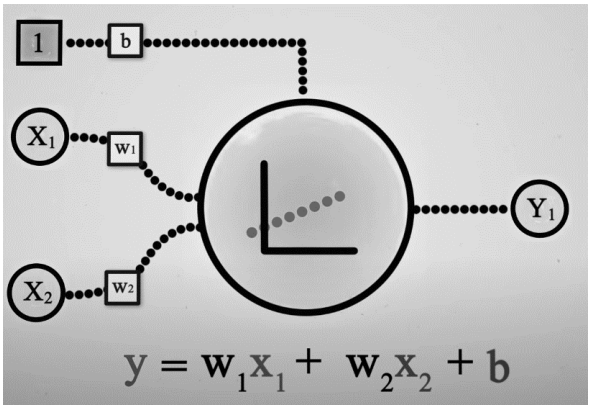


Fig. 1. Diagrama de estructura de una neurona artificial [3].

El umbral tiene un gran protagonismo ya que determina si la entrada neta cumple o no la condición de éste y arroja un valor. Una aplicación de una neurona artificial fue el perceptrón, la cual, fue capaz de realizar una clasificación de tipo binaria que obtuvo la solución con el uso del procedimiento de regresión lineal en donde están involucrados los valores de cada peso y un valor extra llamado sesgo [13]. El siguiente paso fue cuando un problema no se lograba solucionar con regresión lineal, por ejemplo, cuando se intentaron clasificar los resultados de una compuerta lógica XOR, la solución fue implementar dos neuronas (Vea Fig 2).

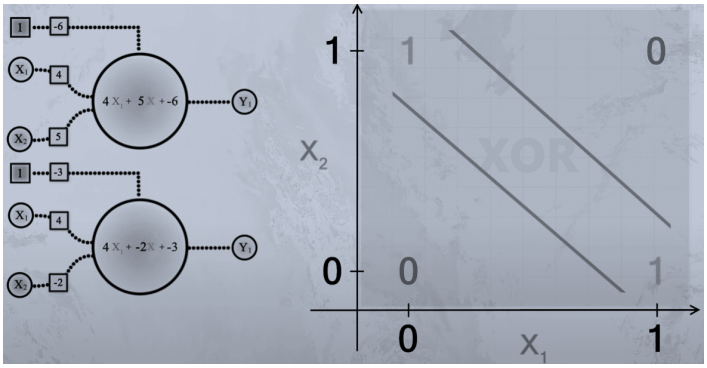


Fig. 2. Diagrama de la solución al problema de la compuerta XOR [3].

La infraestructura de una red neuronal se da al conectar los valores de la salida dados por el umbral con neuronas nuevas formando capas, entre las que encontramos tres tipos (vea tabla 1).

Tabla 1. Clasificación de capas

Tipos de capas.	Función de capa
Capa de entrada	En la primera capa se recibe la información con el previo procesamiento necesario para que pueda funcionar en la red neuronal.
Capas ocultas	Esta capa procesa la información para finalmente enviarla a la siguiente capa, una red neuronal tiene n número de capas ocultas.
Capa de salida	En esta capa existe una o múltiples neuronas que son las responsables de dar un valor de salida a la información ingresada y ya analizada en este punto.

2.2 Entrenamiento

Los algoritmos de Machine Learning tienen como principal característica la función de generar aprendizaje a partir del análisis de información, la cual es ingresada junto con una etiqueta que indica qué es; en el caso del presente experimento, se emplea la base de datos MNIST. El siguiente paso es procesar la información que se ingresa, la capa de entrada tiene un tamaño de 784 neuronas ya que cada píxel de cada imagen con dimensión de 28*28 px, ingresará a cada neurona. La red recibe los datos mediante una función llamada "entrenamiento" en la cual, se ingresan las imágenes con su respectiva etiqueta, de manera que, a medida que la información ingresa a la red neuronal, los pesos y los sesgos de cada neurona recibirán valores aleatorios. La red neuronal trabaja de manera automática y va modificando los pesos para mejorar la clasificación en cada procedimiento, este proceso se llama "época". Los valores de la red neuronal se ajustan mediante el algoritmo de backpropagation y el descenso de gradiente [8].

El backpropagation es un algoritmo publicado en 1986, el cual, ajusta el valor de cada peso de manera correcta para poder tener la respuesta esperada para la clasificación de esta información de una manera más eficaz y ahorrando recursos computacionales. Esto está ampliamente relacionado con el trabajo del descenso del gradiente que es una serie de pasos en la cual se busca el punto más bajo de una gráfica en la que se representa el error, es decir, se busca la manera de reducir significativamente el error de procesamiento de información [2].

Posteriormente se procesa la información y en la capa de salida se comprueba si se clasificó de una manera correcta. Este proceso lo realiza un número de veces, determinado por el valor que se le asigna a la variable epochs, que es la encargada de indicar el número de épocas que entrenará la red. Para tener mejores resultados se establece un número de épocas en un punto medio, con el fin de que la red no tenga un entrenamiento en el que en la etapa de validación, la tasa de eficacia sea baja.

Para el presente experimento se busca una tasa de aprendizaje no menor a 85% en la que permite la generalización en el aprendizaje, pero que no sea muy alto el número de épocas ya que se necesita evitar que la red neuronal sólo sea capaz de clasificar los números que analizó durante el entrenamiento (sobreajuste), es decir, que tenga un porcentaje de eficacia alto que se vea reflejado en la validación de datos que se realiza al término del entrenamiento donde el número de errores es necesario para ajustar los pesos a través del algoritmo de backpropagation [8] y descenso de gradiente [2].

Para saber si nuestra red neuronal tiene un entrenamiento correcto es necesario ingresar información que no haya analizado con anterioridad, la cual, reflejará un valor real de precisión en la clasificación. Esta información la obtenemos de la misma base de datos MNIST.

2.3 Procesamiento de la información

La información de entrada se recibe desde el almacenamiento de un dispositivo móvil con un tamaño de 10 pulgadas en la cual se muestra una imagen estructurada con un ejemplo del dígito a practicar y 6 números a repetir en 9 filas, diseñada de esta forma para que el algoritmo pueda recortar las áreas donde se inscriben los números que el usuario (en este caso un individuo de 8 años) escriba empleando un stylus para representar el uso de un lápiz de grafito.

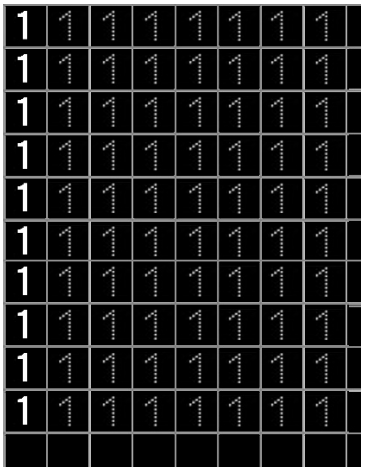


Fig. 3. Imagen empleada para la recepción de los números escritos por el usuario (Elaboración propia).

La figura 3 muestra una imagen con medidas idénticas a las dimensiones de una hoja tamaño carta, en ella se encuentra un total de 70 casillas con líneas punteadas para ingresar un dígito escrito a mano.

Las líneas punteadas desempeñan un papel importante, dado que, si no son cubiertas por los trazos generados causará que el algoritmo lo clasifique de una manera incorrecta y tendrá una mayor posibilidad de no ser reconocido de manera correcta. Cada uno de estos dígitos será recortado y procesado para convertirlo en una imagen de un único canal de escala de grises, redimensionado a un tamaño de 28*28 px para poder tener un total de 784 píxeles y finalmente convertido en un arreglo de 1*784 px, preparado para entrar a nuestra red neuronal.

Para concluir el proceso de la valoración de la información de datos se imprime una matriz de 7*10 elementos, representando las 70 repeticiones de cada número escrito por el usuario, donde se señala con una 'X' si se encontró un número que no se haya escrito adecuadamente y se representa con una '✔' si el dígito fue reconocido como correcto. Finalmente, se imprime un mensaje con la tasa de eficacia al escribir.

3 Desarrollo e implementación

Se utilizaron las herramientas de Tensorflow, la cual es una librería utilizada para crear modelos de Machine Learning y Deep Learning, y Keras, que es un complemento que está diseñado para la experimentación de redes neuronales, con las que se modeló una red neuronal densamente conectada.

Se importa la base de datos de las imágenes de entrenamiento y de validación junto con su etiqueta, posteriormente se inicia la creación del modelo de una red neuronal definiendo las funciones de activación a emplear. Procedemos a preparar las imágenes de entrenamiento y prueba, para ello las redimensionamos a un tamaño de 28*28 píxeles y las guardamos en un arreglo donde sufren una normalización para que la red neuronal sea capaz de procesar la información de cada píxel.

Por último, en la preparación del algoritmo y los datos de la red neuronal, aplicamos la técnica de "One hot encoding", la cual, crea una matriz en la que cada columna simboliza uno de los 784 píxeles, en el canal de escala de grises los valores en color negro se representan con un 0 y ese valor se asigna para los píxeles con valor 0 en esta matriz. Si el píxel tiene un valor distinto se asigna el valor de 1, de esta manera, la red analiza y ocupa los datos para entrenar.

4 Resultados.

La imagen se resolvió mediante la aplicación “Picsart”, la cual, tiene la función de dibujar sobre cualquier imagen, en este caso, será la que auxiliará en el proceso para que el usuario haga las 70 repeticiones en el dispositivo móvil con dimensiones de 10 pulgadas, utilizando el stylus incluido por el equipo (Vea Fig. 4).

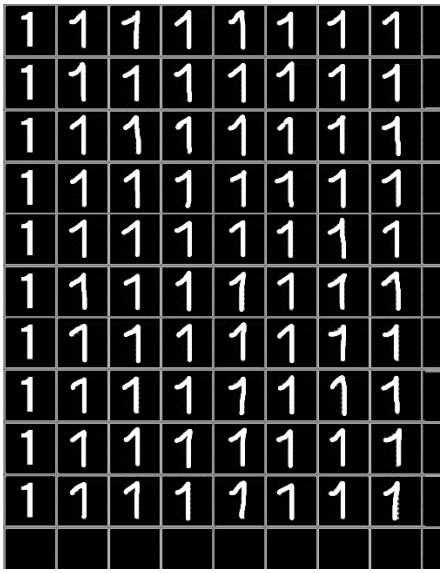


Fig. 4. Imagen resuelta por el usuario en cuestión (Elaboración propia).

Al finalizar la ejecución del algoritmo teniendo como entrada de información la plana resuelta por el usuario presentada en la figura 4 el algoritmo nos da el siguiente resultado. Se genera una representación de la imagen con los dígitos para indicar los dígitos reconocidos con éxito (paloma) y los que no fueron reconocidos de manera correcta (x) (vea Fig 5).

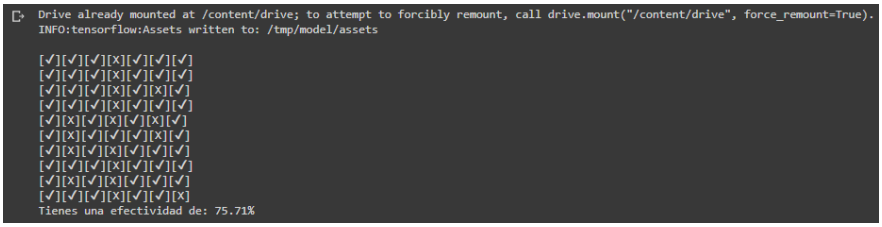


Fig. 5. Impresión de resultados del algoritmo propuesto (Elaboración propia).

El mapa impreso refleja la ubicación de los aciertos y errores detectados en la imagen procesada. Así como un porcentaje de efectividad, que en este caso fue de 75.71%.

Se realizó el experimento en otro usuario de 7 años, se presentan los resultados en la figura 6 y figura 7.

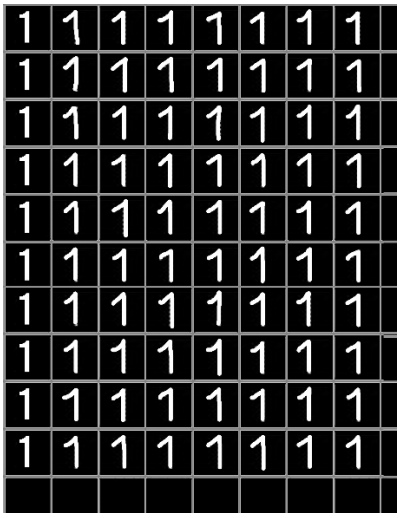


Fig. 6. Ingreso de datos realizado por el usuario 2 (Elaboración propia).

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
INFO:tensorflow:Assets written to: /tmp/model/assets

[✓][✓][✓][X][✓][✓][✓]
[✓][X][✓][X][✓][✓][✓]
[✓][✓][✓][✓][✓][✓][✓]
[✓][✓][✓][X][✓][✓][✓]
[✓][X][✓][X][✓][X][✓]
[✓][✓][✓][X][✓][✓][✓]
[✓][✓][X][X][✓][X][✓]
[✓][✓][✓][X][✓][X][✓]
[✓][✓][✓][X][✓][X][✓]
[✓][X][✓][X][✓][✓][✓]
Tienes una efectividad de: 69.24%
```

Fig. 7. Impresión del análisis realizado a imagen de la figura 6 (Elaboración propia).

5 Conclusión

Al comparar la información obtenida por el algoritmo encontramos que los puntos que no fueron cubiertos de manera correcta tuvieron influencia en los resultados ya que los dígitos que no fueron trazados de manera perfecta se detectaron como errores, mientras que los dígitos ingresados de manera correcta, cubriendo los puntos indicados, se verificaron con éxito.

Se diseñó un algoritmo que mida las deficiencias en la escritura mediante el uso de un dispositivo móvil y hemos realizado las pruebas de manera correcta, aún hay mejoras que realizar, como trasladarlo a una interfaz directamente desarrollada para móviles, también se mejorará a manera de invitar al usuario a practicar hasta lograr una alta tasa de efectividad de escritura.

6 Proyectos futuros

Este proyecto se encuentra aún en desarrollo, el éxito en el reconocimiento de números fue posible por la base de datos empleada y su gran cantidad de información para entrenar la red neuronal, sin embargo, a falta de una base de datos robusta de caracteres alfabéticos, se diseñará una propia para emplearse en el proyecto. Después de reconocer cualquier carácter ingresado, se hará la actualización para no trabajar con la edición de imágenes, sino directamente trabajar con los trazos ingresados en un dispositivo móvil para implementar la inteligencia artificial a la cotidianidad de los humanos y también disminuir el uso de papel destinado a realizar libros de caligrafía que generan tala de árboles que es uno de los principales factores que agravan el calentamiento global.

Agradecimientos

Agradecer a Ana Paola Zurita Águila por su aportación en el asesoramiento de la redacción de este trabajo; a M. en C. María del Carmen Santiago Díaz y al Dr. GustavoTrinidad Rubín Linares por el apoyo recibido para mejorar la presentación en el proyecto.

Referencias

1. ABC Color. (4 de Julio de 2003). La industria papelera es la principal causa de la tala de árboles. *ABC color*. Obtenido de https://www.abc.com.py/articulos/la-industria-papelera-es-la-principal-causa-de-la-tala-de-arboles-706336.html#:%7E:text=La%20tecnolog%C3%ADa%20de%20fabricaci%C3%B3n%20de,ha%20degradado%20el%20medio%20ambiente.
2. Curry, H. B. (1944). The method of steepest descent for non-linear minimization problems. *Quarterly of Applied Mathematics*, 258-261. doi:https://doi.org/10.1090/qam/10667
3. Dot CSV. (19 de Marzo de 2018). ¿Qué es una Red Neuronal? Parte 1 : La Neurona | DotCSV [Video]. Youtube. Obtenido de https://youtu.be/MRIv2IwFTPg?list=PL-Ogd76BhmcB9OjPucsnc2-piEE96jJDQ
4. El Heraldo de México. (4 de Septiembre de 2020). *4 actividades pedagógicas en niños para aprender a leer y escribir.* Obtenido de https://heraldodemexico.com.mx/ninos-actividades/4-actividades-pedagogicas-en-ninos-para-aprender-a-leer-y-escribir/
5. Greenpace. (20 de Marzo de 2020). *La pérdida de bosques y el deterioro ambiental están aumentando el riesgo de transmisión de enfermedades*. Obtenido de https://es.greenpeace.org/es/sala-de-prensa/comunicados/la-perdida-de-bosques-y-el-deterioro-ambiental-estan-aumentando-el-riesgo-de-transmision-de-enfermedades/
6. Lecrec, L. (Noviembre de 10 de 2019). La huella de la industria papelera detrás de los incendios forestales. Obtenido de https://www.animalpolitico.com/blog-invitado/la-huella-de-la-industria-papelera-detras-de-los-incendios-forestales/

7. Manero, P. (8 de Septiembre de 2020). Aplicación apuesta por decir adiós a tickets de papel para reducir impacto ambiental. NotiPress. Obtenido de https://notipress.mx/negocios/app-apuesta-decir-adios-tickets-papel-reducir-impacto-ambiental-4989
8. McCulloch, W. S. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 115-133. doi:https://doi.org/10.1007/bf02478259
9. Mireia Pi, T. P. (2014). *Perspectivas 2014: Tecnologías y pedagogía en las aulas.* junio: Planeta de Libros.
10. Moreno, A. G. (15 de Enero de 2020). Las bolsas de papel no son más ecológicas que las de plástico. *El País*, pág. https://elpais.com/elpais/2020/01/14/buenavida/1579007063_227992.html.
11. Nieto, G. (2 de Febrero de 2020). Economía verde. *El Universal*, págs. https://www.eluniversal.com.mx/opinion/guillermo-nieto/economia-verde.
12. Palacios, R. G. (2003). Sistemas de Reconocimiento de Caracteres para la lectura automática de Cheques. *Publicaciones Marsathrusets Institute ty . Tecnology*, 18-20.
13. Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 386–408.
14. Scheuer, N. E. (2015). Niños de educación inicial y primaria hablan sobre la enseñanza de la escritura. *Revista de Educación*, 354.
15. Vivir, R. (8 de Septiembre de 2020). Los incendios forestales de 2020 podrían ser peores que en 2019 a nivel mundial, advierte WWF. *El espectador*. Obtenido de https://www.elespectador.com/noticias/medio-ambiente/alarmante-aumento-de-incendios-forestales-en-2020/

Selección de Líderes en una Red Social, Utilizando Modelos de Localización de Servicios.

Araceli López[1], Isaí Tlalmis[1], Eduardo López[1], Rogelio González[2]
[1]Facultad de Ciencias Básicas, Ingeniería y Tecnología, Universidad Autónoma de Tlaxcala, Ángel Solana S/N, San Luis Apizaquito, Apizaco, Tlax. México CP 90319.
[2]Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, 14 Sur y Ave. San Claudio, CP 72590, Puebla, Pue. México.
[1]{araceli.lopez, eduardo.lopez01}@uatx.mx, [1]isatla_94@hotmail.com, [2]gzzvzz@gmail.com

Resumen. En este trabajo se aborda la problemática de seleccionar un líder en una red social dada, es modelado como uno de optimización lineal, utilizando los Problemas de Localización de Servicios con Cobertura y de Localización con Cobertura Máxima donde los nodos son los individuos y los arcos la relación entre ellos. Se proponen cuatro parámetros para cuantificar la relación social entre los individuos, El modelo se probó en 4 grupos diferentes de estudiantes de la Facultad de Ciencias Básicas, Ingeniería y Tecnología, de la Universidad Autónoma de Tlaxcala, se resolvió utilizando LINGO 15, se muestran los resultados y se concluye que los mejores parámetros para medir una relación social entre los estudiantes son dos; $1 - |P_i - P_j|$ y $1 - (P_i - P_j)$.

Palabras Clave: Localización de cobertura máxima, Líder, Red social.

1 Introducción

Desde la antigüedad, el hombre ha tenido la necesidad de comunicarse con sus semejantes, lo cual ha provocado una interrelación entre un conjunto de individuos que comparten relaciones de parentesco o simplemente un interés en común. Durante siglos, el tema del individuo en sociedad ha sido discutido por psicólogos, filósofos, sociólogos y economistas. Al mismo tiempo que se han intentado desarrollar modelos sociológicos, biológicos, políticos y psicológicos del comportamiento del individuo dentro de una sociedad [1].

En la década de los cincuenta, algunos antropólogos británicos realizaron estudios de campo en donde se utilizó ampliamente el concepto de red social, dando a éste un valor heurístico. Pero no fue hasta la década pasada cuando el estudio y análisis de redes sociales ha tomado mayor interés dentro de muchas disciplinas como la sociología o la antropología. Iniciándose así, muchas líneas de investigación relacionadas con las diversas aplicaciones del análisis de redes, al igual que el estudio de innumerables situaciones sociales [2].

La idea de red, fue tomada en gran parte de la teoría matemática de los grafos. En esta teoría se llama red a una serie de puntos vinculados por una serie de relaciones que cumplen determinadas propiedades. Es decir, un nodo de la red está vinculado con otro mediante una línea que presenta la dirección y el sentido del vínculo. Entre dos puntos puede haber múltiples tipos de relaciones representadas por grafismos diferentes: estos multígrafos se utilizan cuando dos puntos están relacionados con más de un vínculo de naturaleza diferente [3].

En el presente trabajo se analiza cómo cuantificar una relación de amistad, de confianza, social y de carácter académico, entre dos personas para definir quién de todas las personas de un grupo de estudiantes podría ser un líder, esto se modela como un problema de red logística y para resolverlo se utiliza un modelo de optimización lineal.

2 Planteamiento del problema

En una sociedad sus participantes se relacionan de muchas maneras para lograr diferentes fines, por ejemplo; satisfacción laboral [4], managemet [5], familia-empresa [6], donde siempre existe un líder que los represente y/o toma decisiones importantes para ella. ¿Cómo se relacionan los individuos para elegir un líder, un gobernador, un director o hasta un jefe de grupo? ¿Cómo medir estas relaciones para lograr elegir al "mejor" individuo de una sociedad?

Se abordará la problemática de cómo elegir a un representante o líder de un grupo de estudiantes de la Facultad de Ciencias Básicas, Ingeniería y Tecnología (FCBIyT) de la Universidad Autónoma de Tlaxcala (UATx). Esta problemática tiene que ver con el estudio de una red social y que a su vez se estudiará desde el punto de vista de la optimización como una red logística, se plantea su modelo matemático, se busca resolver el problema planteado usando parámetros referentes a la red y los resultados se traducen a la red social.

El objetivo de esta investigación es modelar matemáticamente el problema de selección de líder en un grupo de estudiantes considerando las conexiones entre ellos.

3 Esquema de un líder

Los líderes no solo se encuentran en puestos relevantes de instituciones públicas y en grandes multinacionales, también hay líderes en contextos cercanos y cotidianos, como los grupos de amigos, compañeros de clase, comunidades de vecinos, equipos deportivos o en las familias. Cabe mencionar que todas las personas son distintas y no todos comparten el mismo modelo de líder, además hay rasgos de la persona líder que se califican como mejor a peor dependiendo de la sociedad en la que sean evaluados.

Como en el presente trabajo se abordó la selección de líder como un problema de red logística, donde los arcos conectan a las instalaciones entre sí y regularmente esta conexión es conocida, entonces surge la pregunta; ¿Cómo cuantificar la relación entre las personas (amistad, parentesco, laborales, etc.)? Para tal fin, se propuso medir estas conexiones con las siguientes características de un líder; intelecto, habilidad comunicativa o social, confianza y seguridad. Se elaboró un cuestionario para evaluar liderazgo y se validó mediante juicio de expertos, dicho procedimiento consistió en tres etapas: elaboración de instrumento, evaluación por parte de expertos quienes de manera individual valoraron las categorías de claridad, coherencia, relevancia y suficiencia de las preguntas y finalmente se hicieron los ajustes al cuestionario de acuerdo a las sugerencias de los expertos para proceder a su aplicación.

El juicio de expertos se define como: "una opinión informada de personas con trayectoria en el tema, son reconocidas por otros como expertos cualificados en éste, y que pueden dar información, evidencia, juicios y valoraciones" [7]. Es un método utilizado con frecuencia en las ciencias sociales y de la conducta [8, 9].

El cuestionario consiste de 16 preguntas, de las cuales 7 se enfocaron a cuestiones sociales, 4 de carácter académico o intelectual, 4 de carácter de confianza y una pregunta donde se describen las características que esperan los encuestados de un líder. En la Tabla 1, se muestran algunas preguntas del cuestionario.

Tabla 1. Preguntas del cuestionario para evaluar liderazgo.

Pregunta 1:	Carácter social
¿Qué compañero del grupo es el que generalmente se encarga de organizar algún convivio o reunión?	Diana, Michel, Diana, Michel, Mauricio, Mauricio, Michel, nadie.
Pregunta 2:	Carácter Académico

¿A qué compañero le pides ayuda cuando tienes problemas de alguna materia o dudas en algún tema de clase?	Verónica, Carlos, Mauricio, Mauricio, Juan Nicolás, Michel, Mauricio, Michel, Hugo, Gerardo.
Pregunta 3:	Carácter Confianza
¿Cuándo realizan alguna actividad deportiva, qué compañero motiva o alienta al resto del equipo?	Yo, Jonatán, Jonatán, Juan Nicolás, Jonatán, Jonatán, nadie, Hugo.
Pregunta 15:	Carácter Académico
Si tuvieras que elegir a un nuevo jefe de grupo, ¿Por quién votarías?	Diana, Gerardo, no sé, Diana, Hugo, Gerardo, Hugo, Antonio.
Pregunta 16:	
¿Por qué?	

La última columna de la Tabla 1, presenta las respuestas a las preguntas del cuestionario, se contabilizaron las menciones de los nombres de los estudiantes, ver la Tabla 2.

Tabla 2. Número de menciones del grupo 1 (ISE) por categoría.

	Carácter académico	Carácter social	Carácter confianza
Michel	5	9	1
Mauricio	10	8	0
Jonatán	0	9	7
Hugo	5	5	3
Jaime	0	3	1
Antonio	3	3	3
Gerardo	5	1	2
Fernanda	0	5	0
Otras respuestas	2	7	9

El cuestionario se aplicó a 4 grupos de estudiantes de la FCBIyT; 2 grupos de Ingeniería en Sistemas Electrónicos (ISE), 1 grupo de Ingeniería Química (IQ) y 1 grupo de la Licenciatura en Matemáticas Aplicadas (LMA), con número de estudiantes 8, 12, 14 y 22 respectivamente, como los grupos son pequeños se consideró el total de estudiantes de cada grupo para responder el cuestionario. En la Tabla 2, el segundo renglón se lee como sigue; a Michel la mencionaron 5 veces en el carácter académico, 9 veces en el carácter social y una vez en el carácter confianza, datos del grupo 1 de ISE. La fila que habla de "otras respuestas", se refiere a las que estuvieron fuera de contexto.

4 Modelo matemático para seleccionar un líder

La metodología que se utilizó fue la siguiente: i) comparar una red logística con una red social, los nodos de la red son los estudiantes, los arcos entre cada instalación son las conexiones sociales que existen entre cada estudiante, ii) cuantificar las conexiones sociales, iii) utilizar el Problema de Localización de Servicios con Cobertura (PLSC) para encontrar el número mínimo de líderes, iii) utilizar el Problema de Localización con Cobertura Máxima (PLCM) para elegir al estudiante líder y con cuantas personas esta mejor relacionado [10].

a. Cuantificar las conexiones sociales

En el PLSC el peso o ponderación y las distancias entre los nodos es cuantificable, en el problema de seleccionar un líder; ¿Cómo medir el peso o ponderación?, ¿Cómo medir la relación entre los individuos? Para dar respuesta a estas preguntas, los autores del presente trabajo proponen los siguientes parámetros:

Tabla 3. Parámetros propuestos para medir la relación entre dos personas.

Parámetro 1	Discreto (0,1)
Parámetro 2	$\|P_i - P_j\|$
Parámetro 3	$1 - \|P_i - P_j\|$
Parámetro 4	$1 - (P_i - P_j)$

Donde P_i es la ponderación que se obtiene utilizando la siguiente relación

$$a_i = \frac{Número\ de\ menciones\ por\ categoría}{(Número\ de\ encuestados)*(Número\ de\ preguntas\ por\ categoría)} \quad \textbf{(1)}$$

De la Tabla 2 y utilizando la ecuación (1), para Michel se tiene lo siguiente; peso en el carácter académico = $\frac{5}{8*4} = 0.15625$, peso en el carácter social= $\frac{9}{8*7} = 0.16071429$ y peso en el carácter confianza = 0.03125. De manera análoga se calculan los datos de la Tabla 4.

Tabla 4. Ponderaciones del grupo 1 de ISE de cada categoría.

	Ponderación carácter académico	Ponderación carácter social	Ponderación carácter confianza	Suma de las 3 ponderaciones
Michel	0.15625	0.16071429	0.03125	0.34821429
Mauricio	0.3125	0.14285714	0	0.45535714
Jonatán	0	0.16071429	0.21875	0.37946429
Hugo	0.15625	0.08928571	0.09375	0.33928571
Jaime	0	0.05357143	0.03125	0.08482143
Antonio	0.09375	0.05357143	0.09375	0.24107143
Gerardo	0.15625	0.01785714	0.0625	0.23660714
Fernanda	0	0.089285714	0	0.089285714

b. Problema de Localización de Servicios con Cobertura

Considerando el Parámetro 3: $1 - |P_i - P_j|$ como una forma de medir la relación entre los estudiantes y utilizando los Algoritmos de Driska y de Floyd se calculó la arborescencia de rutas más cortas en una red, los cuales se programaron en el software Wolfram Mathematica 7.0, permitiendo así construir la siguiente matriz *C.*

$$C = \begin{bmatrix} 1 & 0.84375 & 0.84375 & 1 & 0.84375 & 0.9375 & 1 & 0.84375 \\ 0.84375 & 1 & 0.6875 & 0.84375 & 0.6875 & 0.78125 & 0.84375 & 0.6875 \\ 0.84375 & 0.6875 & 1 & 0.84375 & 1 & 0.90625 & 0.84375 & 1 \\ 1 & 0.84375 & 0.84375 & 1 & 0.84375 & 0.9375 & 1 & 0.84375 \\ 0.84375 & 0.6875 & 1 & 0.84375 & 1 & 0.90625 & 0.84375 & 1 \\ 0.9375 & 0.78125 & 0.90625 & 0.9375 & 0.90625 & 1 & 0.9375 & 0.90625 \\ 1 & 0.84375 & 0.84375 & 1 & 0.84375 & 0.9375 & 1 & 0.84375 \\ 0.84375 & 0.6875 & 1 & 0.84375 & 1 & 0.90625 & 0.84375 & 1 \end{bmatrix}$$

Fig 1. Matriz de distancias con el parámetro 3 para el grupo 1 de ISE.

Se consideró también la distancia estándar o umbral como la media de las distancias de la matríz C, denotada por D, entonces D= 0.875. El correspondiente PLSC queda de la siguiente manera:

$$Minimizar\ Z = \sum_{j=1}^{8} x_j$$

$$Sujeto\ a:$$

$$\sum_{j \in N_i} x_j \geq 1\ i = 1, \ldots, 8$$

$$x_3 = 0$$
$$x_5 = 0$$
$$x_8 = 0$$
$$x_j = (0,1)\ j = 1, \dots, 8$$

Donde $x_j = \begin{cases} 1, si\ el\ nodo\ j\ es\ elegido\ como\ líder \\ 0,\ en\ caso\ contrario \end{cases}$

Utilizando el software Lingo 15, se obtuvo que sólo es necesario un líder, en este caso es el nodo 2 que corresponde a Mauricio.

c. Problema de Localización con Cobertura Máxima

El modelo de cobertura del PLCM que se utilizo es el siguiente:

$$Maximizar\ Z = \sum_{i=1}^{8} a_i Y_i$$

$$Sujeto\ a:$$
$$\sum_{j \in N_i} x_j \geq Y_i \qquad i = 1, \dots, 8$$
$$\sum_{j=1}^{8} x_j = p$$
$$x_3 = 0$$
$$x_5 = 0$$
$$x_8 = 0$$

$$x_j = (0,1), \qquad j = 1, \dots, 8$$
$$Y_i = (0,1), \qquad i = 1, \dots, 8$$

Donde $x_j = \begin{cases} 1, si\ el\ nodo\ j\ es\ elegido\ como\ líder \\ 0,\ en\ caso\ contrario \end{cases}$

a_i es un parámetro que denota el volúmen de demanda o población asociado al nodo *i*. Y_i variable binaria que indica si se cubre o no el nodo *i*, N_i es el conjunto de ubicaciones potenciales que cubrirá el nodo *i*, dentro de la distancia estándar D.

Se resuelve utilizando el software LINGO 15, dando como resultado el nodo 2, Mauricio, como líder y además cubre a los integrantes 1, 4, 6 y 7.

4 Resultados

La metodología que se describe en 4.1, 4.2 y 4.3, se aplicó a cada uno de los 4 grupos de estudiantes, de tal manera que cada grupo tiene sus modelos PLSC y PLCM, con los diferentes parámetros de la Tabla 3. En seguida se muestran algunos resultados obtenidos:

Tabla 5. Resultados para el grupo 1 de ISE.

Parámetro	Nodo elegido	Peso del individuo	Nodos cubiertos
$1 - \lvert P_i - P_j \rvert$	2	0.45535714	1, 4, 6 y 7
$1 - (P_i - P_j)$	6	0.24107143	1, 2, 4 y 7

En la Tabla 5, se observa que usando el parámetro $1 - |P_i - P_j|$ se tiene como líder al nodo 2, Mauricio, éste es uno de los integrantes con mayor peso y cubre a 4 integrantes de la red.

Con el parámetro $1 - (P_i - P_j)$ se tiene como líder potencial al nodo 6, Antonio; quien tiene una ponderación menor a Mauricio y cubre tambien 4 nodos, sin embargo, se encuentra cerca de personas como Mauricio, Michel y Hugo quienes tienen las mayores ponderaciones de su grupo en la categoría académica (Tabla 4), por lo que el modelo puede ayudar a observar el comportamiento de los nodos en su entorno.

Cuando se probó el parámetro discreto [11], las restricciones para los modelos PLSC y PLCM resultaron redundantes no aportando relevancia a los modelos, en este caso como las distancias pueden ser 0 o 1 y D =1, basta con que al asignar a un nodo como posible candidato cubra al resto de ellos.

Para el parámetro $|P_i - P_j|$ se obtuvo el nodo 7, Gerardo como líder, cubriendo los nodos 1, 4 y 6 (Michel, Hugo y Antonio), observe que no cubre a Mauricio que tiene el mayor peso de todos. Este parámetro hace que los nodos con mayor peso estén más lejos del resto de los nodos y los nodos con menor peso se encuentran cercanos a los demás nodos. Situación contraria a lo que se busca, ya que se espera que los nodos con mayor ponderación esten cercanos a los otros nodos en sus respectivas categorías; por lo tanto la expresión $|P_i - P_j|$ no puede usarse como una forma de medir la relación entre individuos [11].

Otros resultados se pueden observar en las Tablas 6,7 y 8

Tabla 6. Resultados para el grupo 2 de ISE.

Parámetro	Nodo elegido	Peso del individuo	Nodos cubiertos
$1 - \|P_i - P_j\|$	3	0.62202381	2, 4, 5, 6, 7, 9, 10, 11 y 12
$1 - (P_i - P_j)$	10	0.130952381	1, 2, 3, 4, 5, 6, 7, 8, 9, 11 y 12

Tabla 7. Resultados para el grupo de LMA

Parámetro	Nodo elegido	Peso del individuo	Nodos cubiertos
$1 - \|P_i - P_j\|$	1	0.46031746	3, 5, 7, 8, 9, 10, 11, 12, 13 y 14
$1 - (P_i - P_j)$	10	0.26984127	1, 2, 3, 4, 6 y 11

Tabla 8. Resultados para el grupo de IQ.

Parámetro	Nodo elegido	Peso del individuo	Nodos cubiertos
$1 - \|P_i - P_j\|$	13	0.553571429	Todos los nodos (23)
$1 - (P_i - P_j)$	20	0.015873016	Todos los nodos (23)

5 Conclusiones y trabajos futuros

Se logró encontrar una forma interesante de medir la relación entre los estudiantes de un grupo y así poder determinar quién de ellos es su líder. Se obtuvo la modelación matemática de seleccionar un líder, resolviéndolo como uno de optimización lineal; utilizando primero una semejanza entre una red social y una red logística y después los

modelos PLSC y PLCM, para obtener la matriz de distancias (matriz de relación social entre los estudiantes) se utilizaron los algoritmos de Driska y de Floyd. La contribución más importante de los autores fue proponer y verificar los parámetros de la Tabla 3, los cuales ayudaron a cuantificar la relación social entre un grupo de estudiantes bajo un cuestionario que define liderazgo, se concluye que los mejores parámetros que miden la relación social son $1 - |P_i - P_j|$ y $1 - (P_i - P_j)$.

Se utilizaron como instancias a cuatro grupos de estudiantes de las diferentes licenciaturas de la Facultad de Ciencias Básicas, Ingeniería y Tecnología de la UATx.

Los trabajos futuros propuestos son; considerar más aspectos cualitativos que describan a un líder, experimentar con una red social más grande, demostrar matemáticamente que los parámetros propuestos pueden ser una métrica.

Referencias

1. Arranz Ainhoa. *Rasgos de la persona líder, tipos de liderazgo y consejos para liderar un equipo*. Ed. Cognifit, México (2017).
2. Lozares Carlos. *La teoría de las redes sociales*. Universidad Autónoma de Barcelona, España, pp. 103-126 (1996).
3. Requena F. *El concepto de red social*. Universidad de Málaga, España. 2002.
4. Yañez R., Arenas M., Ripoll M., *The impact of interpersonal relationships on the general job satisfaction*. SciELO, Perú, ISSN 1729-4827. Lima Perú. Diciembre 2010.
5. Zangaro M. *Subjetividad y trabajo: el management como dispositivo de gobierno*. Núcleo Básico de Revistas Científicas Argentinas del CONICET. N°16, vol. XV, ISSN 1514-6871(Caicyt-Conicet). Santiago del Estero, Argentina. Verano 2011.
6. Carrasco AJ., Sánchez G. *El capital humano en la empresa familiar: un análisis exploratorio en empresas españolas*. Revista FIR, FAEDPYME International Review//Vol. 3N°5// pp.19-29// enero-junio 2014.
7. Escobar J. Cuervo A. *Validez de contenido y juicio de expertos: una aproximación a su utilización*. Avances en Medición, 6,27-36, (2008).
8. Garrote P.R., Rojas C. *La validación por juicios de expertos: dos investigaciones cualitativas en lingüística aplicada*. Revista Nebrija de lingüística aplicada a la enseñanza de lenguas, (18), 124-139. (2015).
9. Galicia L.A., Balderrama J. A., Navarro R. *Validez de contenido por juicio de expertos: propuesta de una herramienta virtual*. Apertura. Guadalajara, Jal. (2017).
10. Serra de la Figuera Daniel. *Métodos Cuantitativos para la Toma de Decisiones*. Universitat Pompeu Fabra, España, pp. 67-72 (2002).
11. Tlalmis I.; López A.; López E.; Reyes V. *Análisis de la Selección de Líderes en una población por medio de Modelos de Localización de Servicios*. Tesis de Licenciatura en Matemáticas Aplicadas. Universidad Autónoma de Tlaxcala. Apizaco Tlax., México, pp 72-78. Junio de 2018.

Comunicación entre CPU y FPGA para Envío de Imágenes

Oscar Kaleb Hernández-Badillo[1], Nicolás Quiroz-Hernández[1], Selene Maya-Rueda[1], Juan Díaz-Téllez[1], and Leonardo Delgado-Toral[2]

[1]Facultad de Ciencias de la Electrónica, Benemérita Universidad Autónoma de Puebla, Av. San Claudio y 18 Sur Ciudad Universitaria, 7250, Puebla, Pue.

[2]Coordinación de Ciencias Computacionales, Instituto Nacional de Astrofísica, Óptica y Electrónica, Luis Enrique Erro #1, 72840, Sta. María Tonanzintla, San Andrés Cholula, Pue.

[1]{oscar.hernandezba,juan.diaztel}@alumno.buap.com.mx,[1] {selene.maya,nicolas.quirozh}@correo.buap.mx, [2]leonardodelgado@inaoep.mx

Resumen. Actualmente la inteligencia artificial (IA) se ha vuelto indispensable para diferentes aplicaciones como son la detección de rostros, sistemas de vigilancia, reconocimiento de objetos, etc. En un procesador, las operaciones realizadas para llevar a cabo una tarea de IA son complejas, por lo que es conveniente tener un acelerador (FPGA, GPU) para el procesamiento digital de imágenes y redes neuronales, de esta forma se emplea el concepto de cómputo en la frontera (Edge Computing). Para enviar los datos al acelerador es necesario una etapa intermedia que corresponde a este trabajo. Esta etapa debe asignar un formato adecuado de representación, con ayuda del módulo receptor del circuito UART (Transmisor/Receptor Asíncrono Universal) se reciben los datos, se almacenan en memoria RAM y se envían al circuito de preprocesamiento para representar los datos adecuadamente. En el presente trabajo se implementaron los bloques UART, RAM y controlador; el bloque UART se encarga de la recepción de la imagen sin procesar y de la transmisión de la imagen procesada, el bloque RAM se encarga del almacenamiento de los datos antes y después de ser procesados y el bloque controlador debe sincronizar la llegada de los datos para su almacenamiento en la memoria RAM y su recuperación para enviarlos hacia la etapa de preprocesamiento que fue emulada haciendo una operación aritmética básica, la cual permite validar la funcionalidad de las etapas requeridas para el proyecto completo. Para probar el funcionamiento de la implementación presentada se utilizó una PC con un procesador AMD Ryzen 7 y una tarjeta Amiba 2 de la empresa Intesc con un FPGA Spartan 6 (XC6SLX9). Para probar los módulos implementados se utilizó un banco de imágenes en escala de grises con un tamaño de 64*64 píxeles las cuales se enviaron por el puerto serial virtual a una velocidad de 19,200 baudios, los datos fueron regresados hacia la PC, donde se realizó la operación contraria al preprocesamiento para verificar que los datos recibidos corresponden con los datos enviados. Esta etapa es necesaria para continuar con el desarrollo del bloque preprocesamiento y las capas de una red neuronal convolucional. Los resultados experimentales muestran que el módulo controlador proporciona una sincronización adecuada entre los módulos UART y RAM, gracias a la recepción y transmisión adecuada de las imágenes.

Keywords: FPGA, VHDL, UART, CNN.

1 Introducción

En años recientes, la inteligencia artificial (IA) se ha visto involucrada en una amplia variedad de aplicaciones. La IA hace posible el surgimiento de vehículos autónomos, ha mejorado el diagnóstico y tratamiento de enfermedades, facilita el proceso educativo [3], es utilizada en el reconocimiento de patrones y en el procesamiento de señales [4], entre otros. A medida que el desarrollo de mejoras en tecnologías computacionales de hardware incrementa, también lo hacen las ramas relacionadas a las ciencias computacionales,

como lo son las áreas de la inteligencia artificial, control de sistemas, análisis remoto de datos, ingeniería biomédica y reconocimiento de objetivos militares [3]. En los trabajos [3] [6] [7] [8] se habla de un acelerador de redes neuronales implementado en FPGA que lleve a cabo el procesamiento, propio de este tipo de sistemas inteligentes. Una de las principales características de la implementación en hardware, es la explotación del paralelismo en redes neuronales, el cual puede mejorar la velocidad de procesamiento y cumplir con los requerimientros de la aplicación. El tratamiento de imágenes y video en tiempo real tiene una amplia gama de aplicaciones, por ejemplo, en las ciudades inteligentes se requiere desarrollar sistemas multicámara en tiempo real para análisis de comportamiento y actividad humana, comprensión de escenarios concurridos, monitoreo ambiental, vigilancia, control de tráfico, etc. [10][11]. En automóviles autónomos, se requiere analizar el movimiento de personas, vehículos, señalética, semáforos, etc. [9][12] De esta manera, surge la importancia de sistemas enfocados a la aceleración del procesamiento implementados en FPGA o GPU, de tal manera que ayuden al microprocesador de una computadora a llevar a cabo las tareas requeridas [5]. Un FPGA a diferencia de un GPU, proporciona grandes ventajas gracias a su bajo consumo de potencia y su propiedad de reconfiguración [8]. Gracias a las características de un FPGA, la implementación de una red neuronal convolucional en hardware puede alcanzar un rendimiento eficiente, para hacerlo posible, es necesario transferir las imágenes al FPGA y almacenarlas en un espacio de memoria RAM, hasta que los datos son tomados por una etapa posterior para su procesamiento.

Como se muestra en el diagrama de bloques del proyecto, representado por la figura 1, se usa una tarjeta Amiba 2 de la empresa Intesc con un FPGA Spartan-6, donde se implementan los módulos UART, RAM y controlador. El circuito convertidor de protocolos (FTDI) se emplea para comunicarse con la PC mediante USB y pasar los datos al FPGA mediante el protocolo serial RS-232.

Los datos recibidos consisten en los valores de los píxeles que forman parte de la imagen, en el FPGA cada dato consta de 8 bits, cada vez que se reciben 8 bits se almacenan en la memoria RAM, donde el circuito controlador se encargará de hacer que se espere hasta que cada píxel de la imagen haya sido recibido. Después, se extraen todos los datos y se

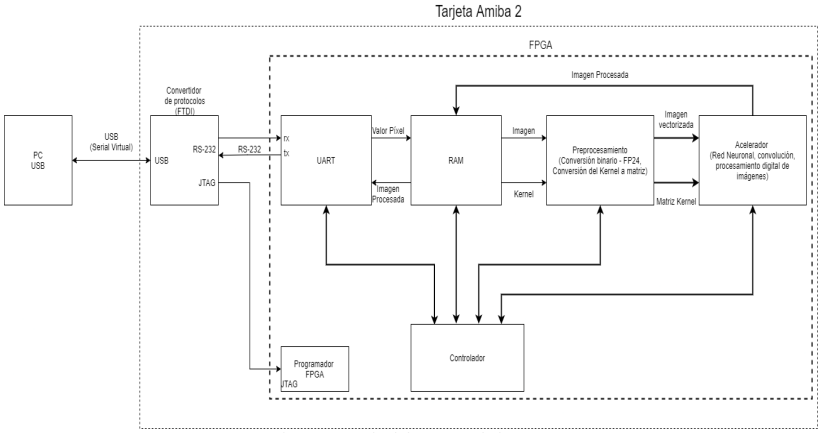


Fig. 1 Diagrama de bloques del proyecto.

transfieren a la etapa de preprocesamiento, garantizando que la imagen se ha recibido en orden y que está lista para ser cambiada al formato que se necesita para después pasarse al módulo acelerador, donde puede contenerse una red neuronal, una operación de convolución o alguna operación del procesamiento digital de imágenes. La imagen procesada es regresada a la memoria RAM para ser transmitida por el circuito transmisor del módulo UART hacia la PC, donde se verificarán las operaciones que fueron aplicadas a la imagen.

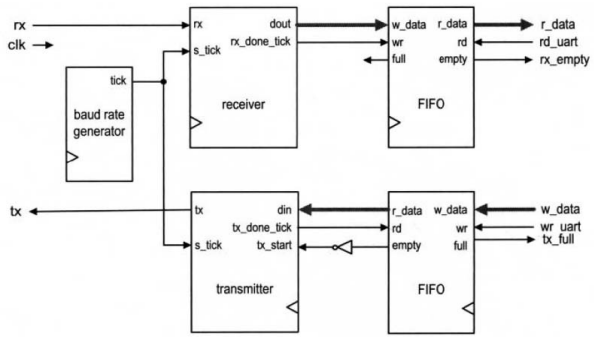
2 Implementación

A continuación, se explican los diferentes elementos creados en VHDL, comenzando por el módulo transmisor/receptor llamado UART. También, se explica el módulo que se encarga del almacenamiento de los datos en la memoria RAM y, por último, el módulo despachador encargado de extraer de manera sincronizada los datos almacenados en la memoria RAM.

2.1 UART

La comunicación serial se implementó para transferir imágenes entre el FPGA y una computadora, mismas que serán procesadas por una red neuronal convolucional, además, se emplea para transmitir los resultados de la operación hacia la computadora. El circuito completo consta de un módulo receptor y un módulo transmisor los cuales son representados en la figura 2.

Fig 2. Diagrama a bloques de una UART completo [2].



En la implementación, se incluye un circuito llamado *Ticker*, el cual proporciona un pulso para sincronizar la recepción y transmisión de los datos. La velocidad de transferencia asignada al diseño es de 19200 baudios. La ecuación 1 determina la frecuencia con la que se emite un pulso hacia los circuitos receptor y transmisor.

$$f = \frac{V}{B \times F_m} \tag{1}$$

Donde, V es la frecuencia de trabajo de la tarjeta que se va a utilizar, B es la velocidad de transferencia en baudios y F_m es el factor de muestreo, que en este caso será igual a

$$f = \frac{50MHz}{19200 \times 16} \tag{2}$$

16.

De la ecuación 1 se deriva la ecuación 2, dando como resultado una f = 163, lo que significa que el circuito *Ticker* emitirá un pulso cada 163 ciclos del reloj, es decir, cada 3.26 μs.

2.2 Circuito despachador y memoria RAM

Para garantizar un almacenamiento y transmisión adecuados se diseñó una máquina de estados (figura 3), cuyo propósito consiste en esperar a que se reciban todos los datos y sean almacenados en la memoria RAM. El módulo *Despachador* contiene una máquina de estados con los estados *IDLE*, *SOLICITAR* e *INICIAR*.

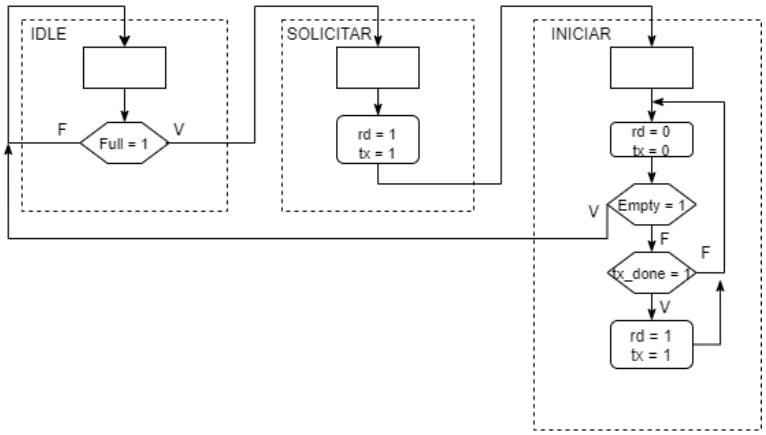


Fig. 3 Carta ASM de la máquina de estados encargada de leer los datos de la memoria RAM

El estado *IDLE* se encarga de mantener la máquina de estados en espera de que la memoria RAM contenga todos los datos que serán usados en el bloque de preprocesamiento y el bloque acelerador. Una vez que los datos han sido recibidos, el controlador de la memoria RAM emite una señal que ocasiona un cambio del estado *IDLE* al estado *SOLICITAR* en la máquina de estados, donde solicita la lectura y transmisión de un dato, después, pasa del estado *SOLICITAR* al estado *INICIAR* donde se extraen todos los datos para que sean enviados a la etapa de preprocesamiento.
Una vez que se han leído todos los datos contenidos en la memoria RAM, la máquina de estados pasa del estado *INICIAR* al estado *IDLE* donde espera hasta que la memoria RAM contenga datos.
Por último, la figura 4 muestra la implementación completa del circuito, donde se tiene el circuito *Ticker* encargado de sincronizar *módulo Receptor*, *la memoria RAM*, *el módulo despachador* y el *módulo Transmisor*.

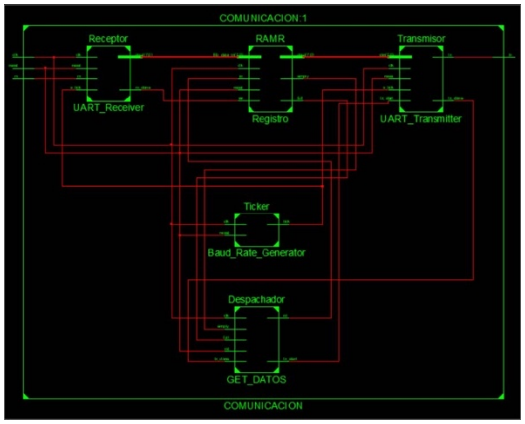


Fig. 4 Diagrama esquemático de la implementación del circuito en VHDL.

3 Resultados

En la figura 5 muestra la simulación del circuito receptor. La simulación consistió en la transmisión de 10 dígitos bajo las especificaciones del diseño, de tal manera que la señal *x_dout* muestra la salida de los 10 dígitos, lo que indica que el circuito recibe los bits adecuadamente.

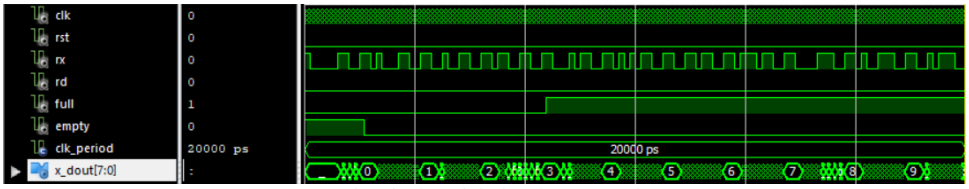


Fig. 5 Simulación del circuito transmisor.

En la figura 6 se muestra la simulación del funcionamiento adecuado del circuito transmisor. La señal din recibe un caracter y realiza la transmisión de los bits que conforman ese caracter a través del puerto *tx*, también, se observa que la transmisión del caracter se realiza durante el estado data.

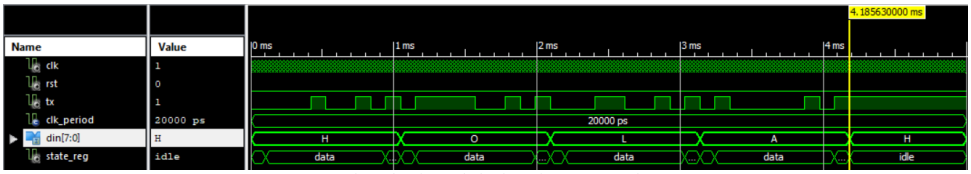


Fig. 6 Simulación del circuito transmisor.

La fig. 7 muestra la simulación del *circuito UART* en conjunto con la *memoria RAM* y el módulo despachador. Como se observa en la imagen, el circuito mantiene una señal *rx* y una señal *tx*. En el fondo de la imagen se encuentra una señal denominada din que indica el dato que está siendo almacenado en el registro, es decir, el dato que se ha recibido y está en espera de ser transmitido. Una vez que el controlador de la *memoria RAM* le indica al *módulo despachador* que comience a extraer los datos, comienza la transmisión y se detiene hasta que ya no hay más datos. De esta manera, se valida el correcto funcionamiento del *módulo despachador* en conjunto con el controlador de la *memoria RAM* y su sincronización con el *módulo Transmisor* del *circuito UART*, lo que garantiza que una vez que los datos hayan sido recibidos, éstos estarán listos para ser pasados a la etapa de preprocesamiento que será desarrollada en una versión posterior al proyecto.

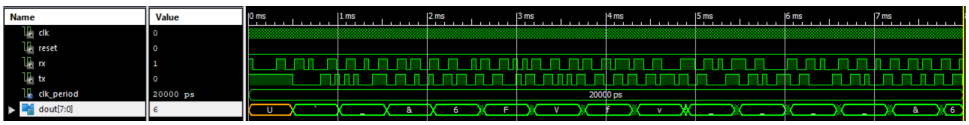


Fig. 7 Simulación del circuito sincronizado con el módulo Despachador.

Las figuras 8, 9 y 10 muestran una interfaz gráfica desarrollada bajo el lenguaje de programación Python en el entorno de desarrollo Spyder para configurar el puerto serial de una manera más rápida. La interfaz permite establecer la velocidad de transferencia, seleccionar si se utilizarán bits de paridad, los bits de stop, el tamaño de los datos que serán usados en la comunicación, en este caso son de 8 bits y, por último, permite seleccionar el puerto por el que se realizará la transmisión. También, se muestra el envío de 3 imágenes de 64*64 pixeles, después de su almacenamiento en el FPGA y su paso por la etapa de preprocesamiento donde se realizó una suma de 1 a todos los datos, ésta se devuelve hacia la PC con el módulo transmisor. En la PC, se restó el valor que fue sumado en la etapa que emuló el preprocesamiento, como se puede apreciar, las imágenes no presentan distorsión alguna, por lo que se puede constatar que el sistema funciona de manera adecuada.

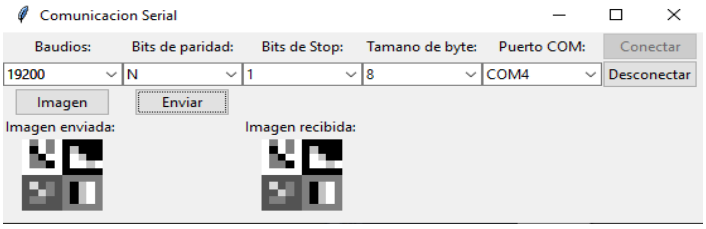


Fig 9 Envío y recepción de la primera imagen.

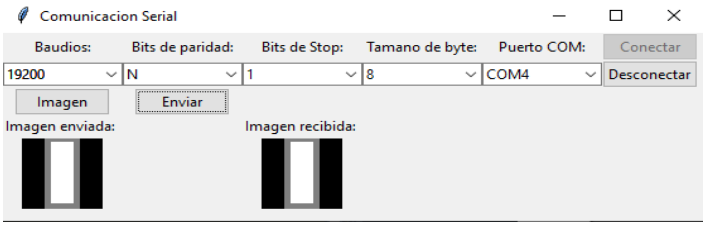


Fig. 10 Envío y recepción de la segunda imagen.

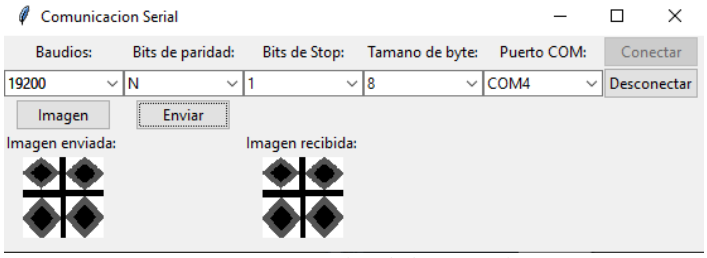


Fig. 11 Envío y recepción de la tercera imagen.

4 Conclusiones

El diseño del controlador (*Módulo Despachador* y controlador de la *memoria RAM*) fue realizado con el fin de poder gestionar la recepción y transmisión ordenada de imágenes a escala de grises en un FPGA, como una herramienta que será usada en un proyecto que consiste en la implementación de las principales capas de una red neuronal, donde se harán

uso de operaciones de punto flotante para realizar la operación de convolución, la función de activación ReLU y SOFTMAX así como la implementación del perceptrón; no obstante, puede ser empleado en sistemas empotrados que necesiten el procesamiento e intercambio de imágenes.
Más adelante, se pretende implementar diferentes arquitecturas de redes neuronales empleando las capas que serán implementadas, utilizando la concurrencia de un FPGA es posible la aceleración del procesamiento de una red neuronal en el FPGA gracias a la concurrencia que ofrecen este tipo de dispositivos.

Referencias

1. P. P. Chu, FPGA prototyping by VHDL Examples: Xilinx Spartan-3 Version, Wiley, 2011.
2. P. P. Chu, RTL Hardware Design usign VHDL, Wiley, 2006.
3. C. Zhang, Y. Wang, J. Guo y H. Zhang, «Digital Recognition Based on Neural Network and FPGA Implementation,» *9th International Conference on Intelligent Human-Machine Systems and Cybernetics,* 2017.
4. B. P. Orozco, «Inteligencia Artificial,» *FCCyT,* 2018.
5. H. Chapman, «From Internet of Things to Smart Cities,» *Enabling Technologies,* 2018.
6. L. Shuai y K. Choi, «Artificial Neural Network Implementation in FPGA: A Case Study,» *ISOCC,* 2016.
7. L. Yongsoon y K. Seok-Bum, «FPGA Implementation of a Face Detector Using Neural Networks,» *IEEE CCeCE/CCGEI,* 2006.
8. L. Bing, Z. Danyin, F. Lei, F. Shou y F. Ping, «An FPGA-Based CNN Accelerator Integrating Depthwise Separable Convolution,» *Electronics,* 2019.
9. M. Maheswaran, Handbook of Smart Cities Software Services and Cyber Infrastructure, Springer, 2018.
10. E. M. G., Tecnología para una movilidad sostenible.
11. A. Urueña., Tecnologías orientadas a la movilidad: valoración y tendencias, 2014: Fundación Vodafone, 2014.

A Feedback Methodology for High-Definition Reconstructions Using CNNs

Armando Levid Rodríguez-Santiago[1,3], José Aníbal Arias-Aguilar[1,4], Hiroshi Takemura[2,5] and Alberto Elías Petrilli-Barceló[2,5]

[1] Universidad Tecnológica de la Mixteca, Km. 2.5 Carretera a Acatlima, 69000, Huajuapan de Léon, Oaxaca, México. Graduate Studies Division

[2] Tokyo University of Science, 2641 Yamazaki, Noda, Chiba 278-8510, Japan. Department of Mechanical Engineering Faculty of Science and Technology

[3] levid.rodriguez@gmail.com

[4] anibal@mixteco.utm.mx

[5] {takemura, petrilli}@rs.tus.ac.jp

Abstract. We present a methodology for the reconstruction of orthomosaics in high-definition resolution. Our proposal consists of two CNN networks connected and working with each other. Each one has been modified in their internal structure from known architectures and re-trained to have enough knowledge to calculate corresponding parameters from two images and perform image stitching to obtain an orthomosaic with high-resolution details. Results obtained show that our methodology provides similar results to those of a reconstruction performed by an expert in orthophotography but with high definition details.

Keywords: Deep Learning · CNN · 2D Reconstruction · Aerial images.

1 Introduction

In the process to generate an orthomosaic (orthophotography), techniques of aerial photogrammetry are used. Photogrammetry is a technique that determines geometric properties of the terrain and spatial relations from aerial photographic images [3].

The union of a set of images that are superimposed and joined in a single image produce an orthophotography. An orthophotography allows us to have current visual knowledge of an area of interest, as well as the visualization of a surface through a photograph where it is possible to appreciate all present elements in a land with validity similar to that of a cartographic plane.

However, it is necessary that the resolution of the generated orthophoto was as high as possible.

In the context of this work, we propose a novel methodology to generate high-resolution orthomosaics. Unlike previous works, we define a proposal that combines main elements of two deep neuronal network models joined with a closed-loop feedback that optimizes the process to generate results.

2 Related Works

Recent research in Deep Learning has included studies to obtain terrain models such as those presented in [2, 4, 8, 12] where they perform image pairing or 3D reconstructions using deep neuronal network techniques. Resulting maps or models need to have details in high-definition (HD), so it is necessary to work with aerial images with high-resolution. In

traditional methodologies, artificial vision techniques and algorithms are usually applied. However, many of these algorithms are not optimized to work with high-definition (HD) images [10, 13].

To deal with this problem, some techniques and architectures have been proposed, such as the one presented in [15]. However, when we need to work with a large number of images, the problem becomes more complex. Nevertheless, to solve problems like these, we can find in the literature proposals ranging from image enhancement to super resolution scaling to recover content from low-resolution (LR) images [7, 20, 18, 11].

3 Methodology

Our approach consist of two main stages. In the first one, feature extraction and key point correspondences are performed from high-resolution input images. With these correspondences at the output of this neural network, input images are stitching. At the end of this process, a low-resolution orthophotography is obtained.

Then, second stage is carried out estimating a high-resolution image from orthophotography obtained in previous stage. In this stage, SRGAN architecture is used to estimate a high-resolution image of orthomosaic obtained in previous stage. This is done through transfer learning and fine tuning of the network. Figure 1 shows the proposed methodology.

Finally, last result of second stage is used as feedback at the input of first stage to build a complete orthomosaic with high-definition details. In this way we are able to handle a large amount of high-definition images and reconstruct large lands areas.

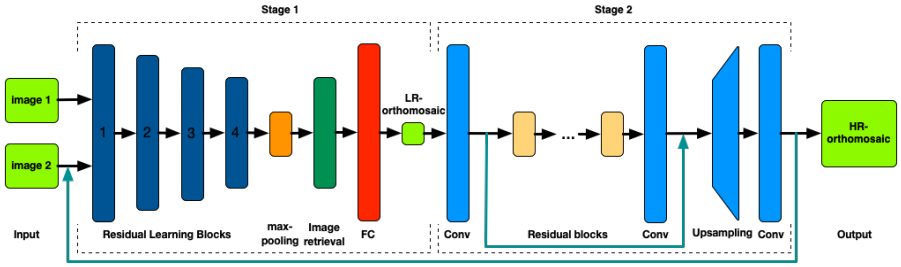


Fig. 1. General structure of proposed methodology. It consists of two stages for the generation of an HD orthomosaic. First stage is in charge of processing and perform the matching of input images. Second stage is responsible to generate a super-resolution image from the low resolution image from previous stage. This result is also used as feedback at the input. Our methodology optimizes the processing in the orthomosaic generation when working with large amounts of images. Each stage is described in subsequent sections.

3.1 Generation of a low-resolution orthomosaic

First stage aims to generate a stitching from two input images. Based on the model developed by Noh et al. [14]. We take the output of the fourth convolutional block of ResNet50 [6] network pre-trained with ImageNet dataset [16].

With this output is possible to find point correspondences between pair of inputs supplied to the network and stitching is carried out. To be able to carry out this procedure we use transfer learning and fine tuning with our previously generated database. Network layers are shown in figure 2.

At input layer, two high-resolution images are supplied. One is taken directly from the database. In the first iteration, second image is also taken from the database, however, for next iterations, an orthomosaic in high-resolution will be supplied as second image. That is, closed-loop feedback is established while first image will be continually taken from the database.

Next stage consists of a fully convolutional network (FCN), which is responsible for extracting dense features from input image. We use the first fourth convolutional blocks of ResNet50 model trained on ImageNet dataset. As in Noh's work, we use an image pyramid and apply each level independently for each input image (see figure 2b). We use a max-pooling layer at the output.

Before features of two branches are combined, we require that their features have some correspondence. Hence, we perform feature descriptor matching for all pairs of images using image retrieval techniques. We use the upper max-pooling layer to establish correspondences, then, descriptors are matched by searching their nearest neighbors. Finally, standard CNN layer (see figure 2c) [19, 14, 1] perform a low-resolution orthomosaic, which is the objective of first stage.

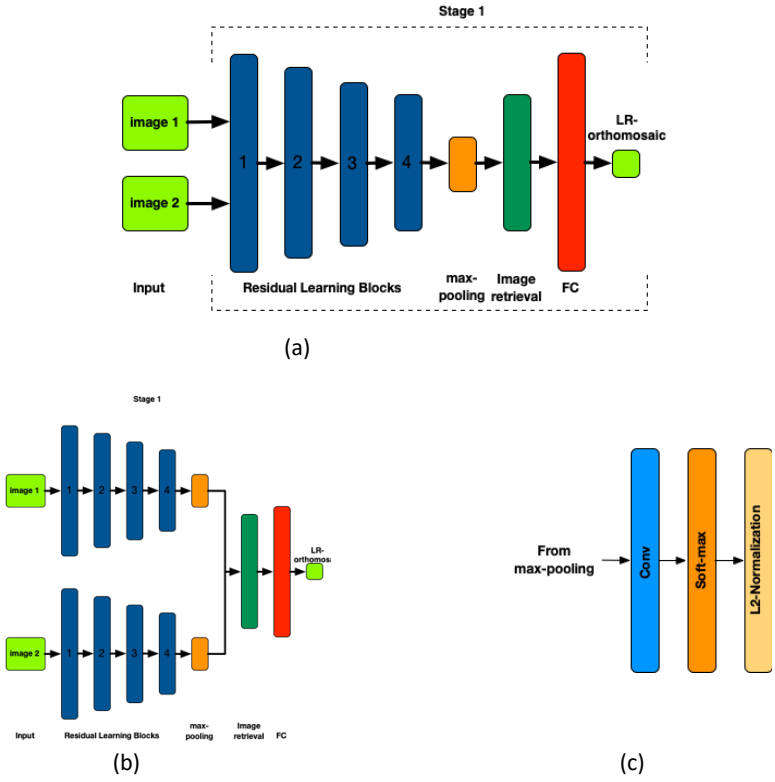


Fig. 2. Inspired by Noh's work and based on the ResNet50 model, first stage is made up of four Residual Learning Blocks (fig. 2a). However, we use simultaneously two branches for extracting dense features from input images (fig. 2b). Then we perform feature descriptor matching for all pairs of descriptors with an image retrieval block (fig. 2c). In this way, pairing of input images is obtained.

3.2 Generation of a high-resolution orthomosaic

Second stage of our proposed methodology aims to generate an orthomosaic with high-definition resolution. While database processing is not complete, their output is used as feedback. At the end, we have a super-resolution orthomosaic.

Inspired by Wang et al,. [18], we follow architectural guidelines for using the SRGAN developed by Ledig et al,. [9]. SRGAN is a generative adversarial network (GAN) for super-resolution imaging (SR). It is efficient for scaling images with a factor of x4 between LR and HR images. General model is illustrated in figure 3a.

Main parts of the second network are Residual Blocks (see figure 3b). Designed with two convolutional layers, batch-normalization and Parametric ReLU [5] as activation

function. The model is composed of 16 of these identical blocks. However, after fine-tuning we only need 10 of these blocks. Upsampling layer increase resolution of the input image with two subpixel convolution blocks as proposed by Shi et al,. [17]. (see figure 3c).

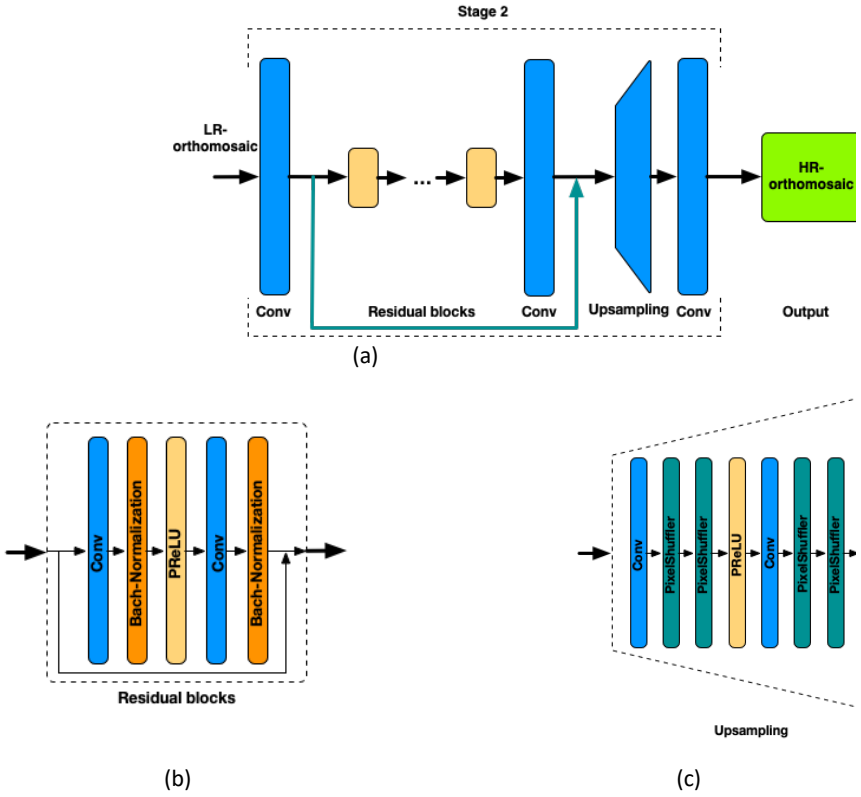


Fig. 3. Second stage of methodology is responsible for generating high definition images. Based on the work developed by Wang, we used a SRGAN model (fig. 3a). One of the main parts are Block Residues, shown in fig. 3b. Another important part is the Upsampling block (fig. 3c.), which is in charge of upscaling input images.

4 Experimental Results

To obtain good results from the models is necessary to increase their knowledge of a specific problem. With this purpose we used a dataset including 2,100 aerial images of entire university campus (Table 1). With this new information, fine-tuning and transfer learning of network architectures is carried out.

Table 1. UTM campus image dataset. This is the way images have been organized so that they can be used to perform fine-tuning and transfer learning of CNN models.

Height\Overlaping	30%x30%	50%x50%
50 mts.	400	800
100 mts.	200	400
150 mts.	100	200
Total	700	1,400

In order to evaluate our proposal, we analyze qualitative results in two steps. In the first

one, we determine the efficiency of our process to generate a low-resolution orthomosaic. In the second step, we verify results to generate a high-definition orthomosaic.

Finally, we check the similarity of final result compared with two orthomosaics, being the first one a manual reconstruction done by an expert in orthophotography, and the second an orthomosaic generated by commercial software.

For testing we selected images related to university greenhouse and nursery area. For orthomosaic generation, we used original images from our database (4K resolution) of selected areas. Because of the terrain conditions, images taken at 100 meters height were selected. With this configuration, it is possible to appreciate a great amount of details in the areas of interest. Total area covered was approximately 22,500m^2. Results are shown in figure 4a, where a low-resolution orthomosaic can be seen. Reconstruction is quite acceptable.

Next stage is the generation of a high-resolution orthomosaic. Then, SRGAN model receives as input an image generated in previous stage. This model is pre-trained to generate images up to 4x the scale of original input image. It is shown in figure 4b where there is an image obtained in previous stage re-generated by the SRGAN model. As we can see, generated textures are very unfavorable compared to originals (see figure 4e).

The result shows that it is necessary that the model has knowledge to work with our dataset, otherwise results are far from having good textures. Therefore, fine-tuning and transfer learning are performed using our database. Then models have knowledge to be able to generate super-resolution images with the characteristics of our database.

Results are shown in figure 4c. Image obtained in the first stage is shown but now with super-resolution. Details can be seen in figure 4f, which is an area of the image. Now it is possible to observe details even when image has been zoomed in.

Although each stage has been configured correctly and both can work together to generate orthomosaics, it was observed that first stage presents limitations when working with more than 100 images. Therefore, it is necessary to simplify the image pairing process. For this, closed-loop feedback is used between output and input of the networks. So now stitching between an image from the database and the output is performed, both in HD. This process improves processing times and increases the ability to work with more than 100 images. The cost is a little lose in image's resolution, which are now HD images.

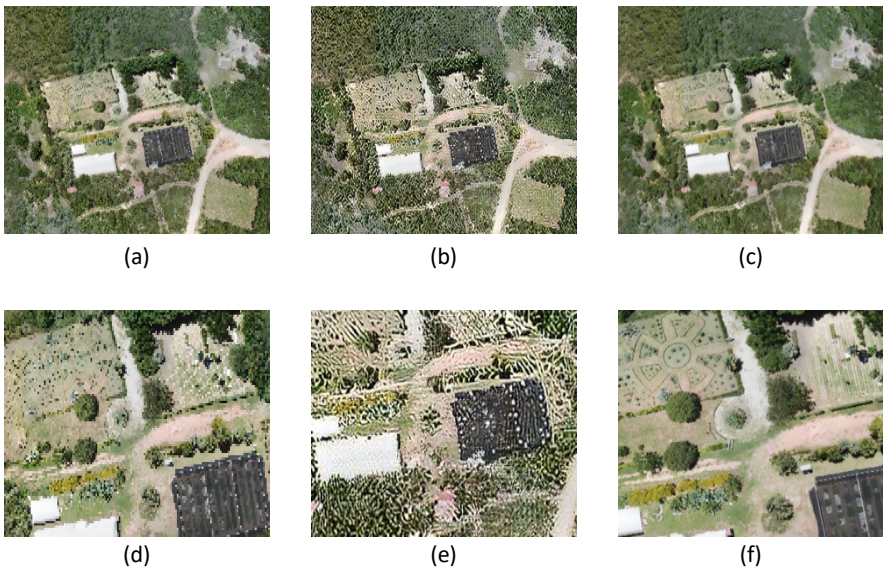


Fig. 4. Results of stages of the proposed methodology. In upper row, results of each stage of methodology are shown. Bottom row shows a zoom applied to each output. Low-resolution orthomosaic reconstruction (fig. 4a). High-resolution reconstruction of the original SRGAN model (fig. 4b). High resolution orthomosaic obtained by the re-trained and modified SRGAN model (fig.

4c).

Obtained results are also validated comparing similarity of the output against a manual reconstruction carried out by an expert and by the commercial Pix4DMapper software.

Manual reconstruction is shown in figure 5a. It was done with images that had high-definition resolution. However, details of image may be scarce, as can be seen in figure 5d. Orthomosaic obtained by commercial software (figure 5c) has been obtained in high-definition. Pix4DMapper™ was only able to get 80% of the selected area. In addition, their database requires special characteristics to guarantee correct operation.

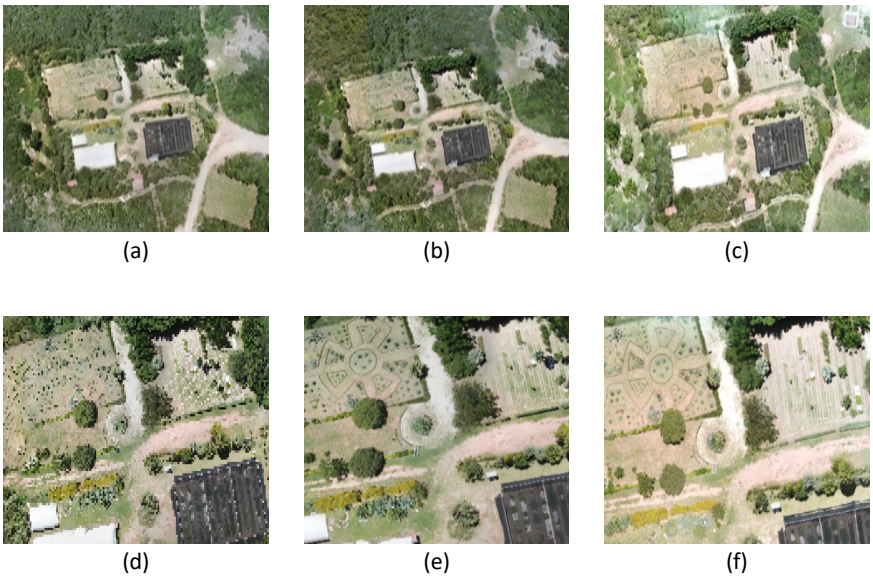


Fig. 4. Results of the proposed methodology. In the upper row, results of each stage are shown. Bottom row shows zoom applied to each output. Low-resolution orthomosaic reconstruction (fig. 4a). High-resolution reconstruction of the original SRGAN model (fig. 4b). High resolution orthomosaic obtained by re-trained and modified SRGAN model (fig. 4c).

We analyze the similarity of resulting orthomosaics. We use Euclidean distance to determine similarity between each one (smaller distance, greater similarity). Results are shown in Table 2. Processing times and resolution of each orthomosaic are shown. These results show that proposed methodology is better in several aspects that commercial software. Furthermore, results are validated in its similarity to the reconstruction of an expert.

Table 2. Orthomosaic comparison. Table shows Euclidean distance as a measure of similarity between our orthomosaic, a manual reconstruction carried out by an expert, and an orthomosaic obtained by Pix4DMapper™ software.

Orthomosaic	Euclidean distance	Processing time	Resolution
Our orthomosaic.	0.0	60 mins.	HD
Manual reconstruction.	7.294145	1,500 mins.	HD
Orthomosaic from Pix4DMapper.	20.139497	120 mins.	HD

5 Conclusions

In this work, a simple methodology for reconstruction of high-definition ortho- mosaics is presented.

This study focuses on verifying the methodology that combines the main structure of two deep neural network models with closed-loop feedback working together to generate an orthomosaic in high-definition. In addition, we also verify the ability of the networks to process large amounts of high-definition images.

Resulting orthomosaics were evaluated using Euclidean distance as a measure of similarity. Our orthomosaic was compared with a manual reconstruction performed by an expert in photogrammetry, and a reconstruction obtained with commercial software. It is shown that our methodology provides similar results to those of a manual reconstruction but with high-definition details.

Generation of orthomosaics in full high-definition (FHD) or in higher-resolutions is being considered for future work.

References

1. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN architecture for weakly supervised place recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5297–5307 (2016)
2. Chen, Y., Liu, L., Gong, Z., Zhong, P.: Learning CNN to pair UAV video image patches. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 10(12), 5752–5768 (2017)
3. Cheng, Y., Xue, D., Li, Y.: A fast mosaic approach for remote sensing images. In: 2007 International Conference on Mechatronics and Automation. pp. 2009–2013. IEEE (2007)
4. Ghamisi, P., Yokoya, N.: Img2dsm: Height simulation from single imagery using conditional generative adversarial net. IEEE Geoscience and Remote Sensing Let- ters 15(5), 794–798 (2018)
5. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human- level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034 (2015)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
7. Huang, H., He, R., Sun, Z., Tan, T.: Wavelet-srnet: A wavelet-based cnn for multi- scale face super resolution. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1689–1697 (2017)
8. Le, C., Li, X.: Jigsawnet: Shredded image reassembly using convolutional neural network and loop-based composition. IEEE Transactions on Image Processing 28(8), 4000–4015 (2019)
9. Ledig,C.,Theis,L.,Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super- resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4681–4690 (2017)
10. Li, J., Ai, M., Hu, Q., Fu, D.: A novel approach to generating DSM from high- resolution UAV images. In: Geoinformatics (GeoInformatics), 2014 22nd International Conference on. pp. 1–5. IEEE (2014)
11. Li, K., Ye, L., Yang, S., Jia, J., Huang, J., Wang, X.: Single image super resolution based on generative adversarial networks. In: Eleventh International Conference on Digital Image Processing (ICDIP 2019). vol. 11179, p. 111790T. International Society for Optics and Photonics (2019)
12. Li, S., Zhu, Z., Wang, H., Xu, F.: 3D Virtual Urban Scene Reconstruction From a Single Optical Remote Sensing Image. IEEE Access 7, 68305–68315 (2019)
13. Li, T., Hailes, S., Julier, S., Liu, M.: UAV-based SLAM and 3D reconstruction system. In: Robotics and Biomimetics (ROBIO), 2017 IEEE International Conference on. pp. 2496–2501. IEEE (2017)

14. Noh, H., Araujo, A., Sim, J., Weyand, T., Han, B.: Large-scale image retrieval with attentive deep local features. In: Proceedings of the IEEE international conference on computer vision. pp. 3456–3465 (2017)
15. Rodríguez-Santiago, A.L., Arias-Aguilar, J.A., Petrilli-Barceló, A.E., Miranda- Luna, R.: A Simple Methodology for 2D Reconstruction Using a CNN Mode. In: Mexican Conference on Pattern Recognition. pp. 98–107. Springer (2020)
16. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: ImageNet large scale visual recognition challenge. International journal of computer vision 115(3), 211–252 (2015)
17. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1874–1883 (2016)
18. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Change Loy, C.: Esrgan: Enhanced super-resolution generative adversarial networks. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 0–0 (2018)
19. Widya, A.R., Torii, A., Okutomi, M.: Structure from motion using dense CNN features with keypoint relocalization. IPSJ Transactions on Computer Vision and Applications 10(1), 6 (2018)
20. Yamanaka, J., Kuwashima, S., Kurita, T.: Fast and accurate image super resolution by deep CNN with skip connection and network in network. In: International Conference on Neural Information Processing. pp. 217–225. Springer (2017)

Análisis de Poemas en Español para Identificar Ideación Suicida

Nancy A. Carmona-Contreras, Maya Carrillo-Ruiz, Luis E. Colmenares-Guillen, José L. Hernández-Ameca

Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla Ciudad Universitaria. Blvd. Valsequillo y Av. San Claudio, Col. Jardines de San Manuel, Puebla, México. CP 72570.

nancyacc13@hotmail.com, maya.carrillo@correo.buap.mx, enrique.colmenares@correo.buap.mx, joseluis.hdzameca@correo.buap.mx

Resumen. Son diversos los factores que orillan a una persona a cometer un suicidio, entre ellos destacan las enfermedades mentales, el abuso de sustancias, la violencia y el entorno sociocultural. En el 2015 en México ocurrieron 2 599 fallecimientos por lesiones auto infligidas en jóvenes de entre 15 y 29 años, lo que representó una tasa de 8.2 suicidios por cada 100,000 habitantes, cifra que incrementó en el año 2017 llegando a 2662 fallecimientos. En el presente trabajo, se dan los primeros pasos para construir una herramienta que, en un futuro, apoyen en redes sociales a detectar tendencias suicidas, ahora se inicia detectando tendencias suicidas en poemas en español, utilizando técnicas de procesamiento de lenguaje natural y aprendizaje automático. Los resultados obtenidos son alentadores ya que combinando representaciones textuales que capturan la parte léxica y sintáctica de los documentos se obtiene una exactitud del 74.89%.

Palabras Clave: detección de ideación suicida, POST, aprendizaje automático, procesamiento de lenguaje natural.

1 Introducción

En el 2015 en México ocurrieron 2 599 fallecimientos por lesiones auto infligidas en dicho grupo de edad, lo que representa una tasa de 8.2 suicidios por cada 100 000 habitantes de éste mismo grupo [1]. Datos del año 2017 indican que las tasas de suicidio de jóvenes de entre 15 y 29 años van desde 7.1 a 9.3 suicidios por cada 100 000 habitantes, donde la población de 20 a 24 años muestra la tasa más alta de suicidio con 9.3 suicidios por cada 100 000 jóvenes de esa edad [2].

Un estudio realizado en estudiantes de licenciatura de entre 17 y 26 años reveló que el 30.9 por ciento se había lesionado a sí mismo. Aunque el investigador Everardo Castro Silva indica que se tratan de lesiones no suicidas, si este comportamiento no se trata a tiempo podría escalar a trastornos más graves incluso llegar a suicidio [3].

Según la OMS el suicidio es la segunda causa más frecuente de muerte en jóvenes con más de 800 000 suicidios – una muerte cada 40 segundos - y planea poner en marcha programas integrales para disminuir la tasa de suicidios de los países miembro en un 10% para el año 2020 aplicando estrategias de prevención del suicidio, por medio de la sensibilización pública, política y mediática sobre la magnitud del problema, promoción de información responsable por parte de los medios en relación con los casos de suicidio, promoción de iniciativas de prevención del suicidio en el trabajo, mejora de las respuestas del sistema de salud a las autolesiones y suicidios y la optimización del apoyo psicosocial

con los recursos comunitarios disponibles, tanto para quien haya intentado suicidarse como para las familias de quienes hayan cometido suicidio [4].

La académica e investigadora Isabel Stange Espíndola aseguró que los suicidas generalmente dan indicios o señales antes de que deseen quitarse la vida, que deben ser tomadas en cuenta como llamadas de auxilio por las personas que están cerca del individuo. Como medida preventiva en la BUAP, se ha diseñado un programa permanente de prevención del suicidio en la universidad. Con el apoyo de estudiantes de psicología, dirigidos por la maestra Stange, se inició una serie de actividades que involucran información que pueda orientar al público sobre qué hacer en caso de observar señales de alarma en algún conocido, así como identificar las ideas suicidas [5].

En función de lo mencionado anteriormente, es importante emplear mecanismos que ayuden a detectar tendencias suicidas a tiempo. En el presente trabajo, se dieron los pasos iniciales para llegar a construir herramientas que, en un futuro, apoyen en redes sociales a detectar tendencias suicidas, se inició detectando tendencias suicidas en poemas en español, utilizando técnicas de procesamiento de lenguaje natural y aprendizaje automático.

2 Trabajo Relacionado

Hajime Sueki en su investigación [6] realizó una encuesta en línea donde se le preguntaba a los jóvenes si eran usuarios de Twitter, si eran usuarios activos de esta red y si habían publicado alguna vez que querían morir y que querían cometer suicidio, además de proporcionar algunos datos demográficos como edad, sexo, educación, estatus marital, ingresos, estatus laboral, uso de alcohol y visitas al hospital. Posteriormente se les preguntó si se habían lesionado deliberadamente, si tenían pensamientos suicidas, si habían planeado alguna vez un suicidio y si lo habían intentado. Para comenzar con el análisis primero se clasificaron a los 14 529 participantes en 4 grupos: 6 382 no tienen cuenta de Twitter, 4 243 son usuarios inactivos de twitter, 2 936 son usuarios activos sin tuits suicidas y 968 son usuarios activos con tuits suicidas. Después de conformar un data set final de n=1 000, con 250 de cada grupo señalado anteriormente, se calculó que el 61.3% son mujeres y la edad promedio es de 24.9 años. Posteriormente se realizó un análisis de regresión logística para examinar la relación entre el uso de Twitter y el comportamiento suicida a partir del conjunto de datos final. Se encontró que el hecho de publicar algo como "quiero morir" mostró una significativa relación con un historial de ideas suicidas (3.2 veces más susceptible a tener ideas suicidas,3.2 veces más a probable a planear un suicidio y 2.1 veces más a intentar un suicidio que aquellos que no lo hacen). Publicar "quiero cometer suicidio" tiene una significativa relación con lesiones auto provocadas, tener un plan de suicidio o de haberlo intentado (son 2 veces más susceptibles a tener comportamientos de auto lesión, 2 veces más propensos a tener un plan suicida y 3.5 veces más propensos a intentar un suicidio que aquellos que no publicaron "quiero cometer suicidio") así que una búsqueda con "quiero cometer suicidio" podría ayudar a encontrar más personas suicidas con historial de intentos de suicidio. Tener una cuenta en Twitter y el uso regular de este, por sí mismo no presentó ninguna asociación con comportamiento suicida.

En otro trabajo sobre Twitter, Bridianne O'Dea, et al. [7] recolectaron publicaciones – incluyendo el nombre de usuario y foto - de Twitter por medio de una API, que contuvieran determinadas frases consistentes con ideas suicidas. En este estudio se contemplaron 3 posibles clases para las publicaciones: fuertemente preocupante, posiblemente preocupante y seguro de ignorar. Se utilizó el esquema de ponderación TF-IDF, además se añadió una ligera variante en los documentos: se fijó un filtro para que aquellas palabras que tuvieran un valor de frecuencia mayor a 0.7 se removieran, ya que se cree que son palabras que al repetirse mucho contribuyen con poca información. La primera prueba se hizo con un clasificador implementado por el equipo de investigadores que categorizó 2 000

publicaciones: 14% fue encontrado "altamente preocupante", 57% como "posiblemente preocupante" y 29% como "seguro de ignorar". Se hicieron otras pruebas con clasificadores y el que mejores resultados fue el obtenido con Máquina de soporte vectorial (SVM) con TF-IDF que obtuvo una exactitud de hasta 76%.

En [8] Jhon Pestian, et al. buscan probar si los algoritmos de aprendizaje automático pueden clasificar notas de suicidio de igual o mejor manera que los expertos en salud mental y practicantes. Primero recolectaron 66 notas de personas que cometieron suicidio y notas "fabricadas", escritas por personas a las que se les pidió escribir una nota como si estuvieran dispuestos a cometer suicidio, y se pre procesaron. Después se separaron en 33 de suicidas y 33 de no suicidas. Las notas fueron mostradas a expertos en salud mental y practicantes quienes las clasificaron en genuinas y fabricadas, a cada nota clasificada se le anotó algún concepto emocional. Se construyó una ontología con ayuda de expertos en literatura y de consultas en PubMed.Se seleccionaron 5 métodos de aprendizaje automático: árboles de decisiones, reglas de clasificación, modelos funcionales, lazy learners y meta learners. En promedio el mejor algoritmo de aprendizaje automático (Logistic Model Trees -LMT) se desempeñó mejor que los practicantes y profesionales de salud mental. Todos los algoritmos lo hicieron significantemente mejor que los practicantes. Nueve de diez algoritmos dieron un desempeño mejor que los profesionales de salud mental.

También las tendencias de búsquedas de Google han sido utilizadas para detectar suicidio Jhon F. Gun y David Lester en [9] recolectaron 3 conjuntos de datos de todos los estados de EE.UU.: a) Tendencias de búsqueda para los términos "cometer suicidio", b) Tendencias de búsqueda para los términos "prevención de suicidio" y c) Tendencias de búsqueda para los términos "como suicidarse". El equipo investigador utilizó los datos sobre suicidio en los EE.UU. de un estudio previo. Luego, se empleó la correlación de Pearson – el índice de correlación de Pearson se usa para medir el grado de relación entre dos variables - para investigar la relación entre las tendencias de búsqueda y las tasas de suicidio. La relación entre las tasas de suicidio y el volumen de búsquedas para "cometer suicidio" es significativa y positiva [r=0.31 p=0.01]. La relación entre las tasas de suicidio y el volumen de búsquedas de "como suicidarse" fue menos significativa y positiva [r=0.21 p=0.07]. La relación entre las tasas de suicidio y el volumen de búsquedas sobre "prevención de suicidio" fue significativa y positiva [r=0.61 p=0.001]. Estos resultados indican que, en estados – de EE.UU. - con altas tasas de suicidio, es más alto el volumen de búsquedas de términos asociados al suicidio.

Por otro lado, en [10] encontramos un trabajo parecido al desarrollado en este trabajo, donde se trata de predecir la probabilidad de que un músico/escritor de canciones pueda o haya cometido suicidio. Para esto construyeron un corpus de letras de canciones que incluye un grupo de desarrollo, un grupo de entrenamiento y otro de prueba. Empleando algunos algoritmos de Weka lograron alcanzar hasta un 70.6 % de precisión con el algoritmo SimpleCart. El corpus fue conformado por letras de escritores masculinos que cometieron suicidio y que no lo hicieron. El corpus de entrenamiento se formó con 533 canciones, el de test con 109 canciones y el de desarrollo con 109. Cada canción fue limpiada a mano, cada sentencia fue separada por un punto para ser procesada por un POS-tagger y un lematizador. Las letras también fueron tokenizadas usando OpenNLP tokenizer. Luego los atributos fueron calculados por un script en Python. La selección de los atributos puede identificarse en 3 grupos: 1) Atributos de vocabulario: conteo de tipos (obtenidos del POS-tagger), tokens, 2) tiempo de la canción en segundos, 3) TTR (type-token ratio). En cuanto a atributos sintácticos se exploró la diferencia en el uso de la construcción pasiva y en la proporción de pronombres en primera persona con respecto al resto de los pronombres. Para los atributos semánticos se propuso una hipótesis de encontrar una diferencia en el uso de palabras sexuales en las letras suicidas (específicamente una cantidad elevada de palabras relacionadas con sexo y muerte). Las clases semánticas consideradas fueron: sensualidad, acción, muerte, amor, depresión y drogas.

En [11] se usan algoritmos de Procesamiento de Lenguaje Natural para identificar a mujeres embarazadas con comportamiento suicida donde se identificó a 2 945 mujeres por

medio de PLN y 196 mujeres por datos codificados de un total de 275 843 mujeres embarazadas o que recientemente habían pasado por parto. También se identificó a 1423 (125 por código de diagnóstico y 1 298 por PLN) mujeres embarazadas con comportamiento suicida de 9 331 mujeres con resultados positivos, esto quiere decir que con el PNL los resultados incrementaron 11 veces, de 125 a 1423. Esto arroja un estimado de la prevalencia de comportamiento suicida de 515.87 por cada 100 000 mujeres embarazadas o con un parto reciente.

3 Método propuesto

A continuación, se describen los pasos seguidos para detectar ideación suicida en poemas: a) Primero se amplió el corpus presentado en [12] se agregaron 6 autores y 300 poemas nuevos, obteniendo así un corpus final conformado por un total de 900 poemas de 18 autores diferentes. La razón de elegir poemas de estos autores y no de otras fuentes, por ejemplo, internet/redes sociales, es que en estos casos se tiene la certeza de que los textos corresponden a personas que cometieron suicidio.

En la Tabla 1 se encuentra la lista completa de los autores que conforman el corpus. b) Reunidos los poemas estos fueron preprocesados para convertirlos a minúsculas, quitar signos de puntuación y números, eliminar palabras vacías (palabras como preposiciones, artículos, conjunciones), y truncar mediante Snowball Stemmer, es decir eliminar prefijos y sufijos. c) Después se calculó la ponderación tf-idf de los poemas con y sin palabas vacías, d) Posteriormente se calculó la ponderación tf-idf de la representación de los poemas con etiquetas POST (part-of-speech tagging**)** con y sin palabras vacías. Las etiquetas POS sirven para asignar a cada palabra su categoría gramatical, en este trabajo se usó el Stanford POS tagger. e) También se experimentó con la suma y el promedio de los valores tf-idf de los términos y las etiquetas POS. La ponderación tf-idf se calculó utilizando la ecuación (1).

$$Wij = tfi,j \; x \log(\frac{N}{dfi}) \tag{1}$$

Donde tf_{ij} es el número de ocurrencias del término i en j, df_i es el número de documentos con el término i y N es el número de documentos en la colección.

f) También se obtuvo el promedio de palabras por oración y se añadió como un nuevo atributo. Para cada poema se contó el número de palabras y se dividió entre el número total de oraciones en el mismo, como se representa en la ecuación 2.

$$promedioPalabras_i = \frac{totalPalabrasOracion_j}{totalOracionesPoema_i} \tag{2}$$

g) De la misma manera se hizo con el número de trigramas de cada poema, agregando el valor como atributo. Los trigramas son subcadenas de 3 caracteres.

h) Finalmente los documentos así representados se clasificaron empleando los algoritmos SVM, Naive Bayes, y C4.5 y utilizando Weka.

Tabla 1. Detalle de poetas que conforman el corpus

Poeta	Número de documentos	País de origen	Tipo
Alejandra Pizarnik	50	Argentina	Suicida
Alfonsina Storni	50	Suiza-Argentina	Suicida
Jaime Torres Bodet	50	México	Suicida
José Antonio Ramos Sucre	50	Venezuela	Suicida
José Asunción Silva	50	Colombia	Suicida

Leopoldo Lugones	50	Argentina	Suicida
José Agustín Goytisolo	50	España	Suicida
Manuel Acuña	50	México	Suicida
Violeta Parra	50	Chile	Suicida
Amado Nervo	50	México	No suicida
Humberto Garza	50	México	No suicida
Delmira Agustini	50	Uruguay	No suicida
José Emilio Pacheco	50	México	No suicida
Juan de Dios Peza	50	México	No suicida
Julián del Casal	50	Cuba	No suicida
Octavio Paz	50	México	No suicida
Ramón López Velarde	50	México	No suicida
Salvador Díaz Mirón	50	México	No suicida

4 Resultados

Se experimentó con los 900 poemas empleando validación cruzada a 10 pliegues. Los experimentos realizados fueron: a) Representación tf-idf con y sin palaras vacías, b) Representación tf-idf con etiquetas POST con y sin palabras vacías, c) Suma de tf-idf de términos y etiquetas POST con y sin palabras vacías y d) Promedio de tf-idf de términos y etiquetas POST con y sin palabras vacías. Todo esto se realizó 3 veces, una con tf-idf de términos solamente, otra para tf-idf + promedio de palabras y la última para tf-idf + número de ngramas. Los resultados de estos experimentos se encuentran en las Tablas 2, 3 y 4.

Tabla 2. Exactitud para la representación tf-idf con validación cruzada a 10 pliegues

Representación	Naive Bayes	C4.5	SVM
Términos con palabras vacías	66	61.22	74.89
Términos sin Palabras vacías	66.33	61.89	71.56
POS con palabras vacías	60.78	59	63.22
POS sin palabras vacías	60.67	54.22	60.89
Suma tf-idf de términos y POST con palabras vacías	60.33	58.78	72.78
Suma tf-idf de términos y POST sin palabras vacías	54.78	60.67	70.44
Promedio tf-idf de términos y POST con palabras vacías	60.33	59.89	72.78
Promedio tf-idf de términos y POST sin palabras vacías	54.67	59.78	70.44

Tabla 3. Exactitud para la representación tf-idf + promedio de palabras con validación cruzada a 10 pliegues

Representación	Naive Bayes	C4.5	SVM
Términos con palabras vacías	65.89	61.33	74.89
Términos sin Palabras vacías	66.22	61.78	71.78
POS con palabras vacías	71.78	59.56	62.22
POS sin palabras vacías	60.44	57.78	61
Suma tf-idf de términos y POST con palabras vacías	60.33	59.2222	72.89
Suma tf-idf de términos y POST sin palabras vacías	54.78	60.89	70.22
Promedio tf-idf de términos y POST con palabras vacías	60.33	60.22	72.89
Promedio tf-idf de términos y POST sin palabras vacías	54.67	60.22	70.22

Tabla 4. Exactitud para la representación tf-idf + número de ngramas con validación cruzada a 10 pliegues

Representación	Naive Bayes	C4.5	SVM
Términos con palabras vacías	66	61.22	74.33
Términos sin Palabras vacías	66.22	61.67	71.56
POS con palabras vacías	60.78	60.67	62.22
POS sin palabras vacías	60.33	59	60.44
Suma tf-idf de términos y POST con palabras vacías	60.33	58.67	73.44
Suma tf-idf de términos y POST sin palabras vacías	54.78	60.67	71
Promedio tf-idf de términos y POST con palabras vacías	60.33	60	73.44
Promedio tf-idf de términos y POST sin palabras vacías	54.67	59.78	71

Como podemos observar en la Tabla 2 la mayor exactitud de 74.89% se obtiene con SVM utilizando en algoritmo SMO de Weka. Aunque el trabajo reportado [10] tiene condiciones diferentes y utiliza un corpus distinto reporta una exactitud del 70.6% lo que hace equiparable nuestros resultados a dicho trabajo. De igual manera los experimentos que combinan la representación léxica y sintáctica calculando la suma o promedio de tf-idf de los términos más el tf-idf de las etiquetas POST con las palabras vacías obtienen una exactitud aceptable del 72.78%. Puede observarse también en la tabla 2 que los resultados obtenidos utilizando solo etiquetas POST son los más bajos.

En la tabla 3 se muestra también un porcentaje del 74.89% de exactitud obtenido al analizar la representación tf-idf más el promedio de palabras como atributo adicional para cada poema, sin embargo, al no representar una mejora con respecto al mejor resultado de la tabla 2 podemos concluir que el atributo no aporta información para la clasificación. Del mismo modo sucede en la tabla 4 donde el mayor porcentaje obtenido es de 74.33% de exactitud con el número de ngramas como nuevo atributo.

5 Conclusiones y Trabajo Futuro

En este trabajo puedo observarse que la información que proporcionan las etiquetas POST, de manera independiente no apoyan a la tarea de identificar textos con ideación suicida. De igual manera, las palabras vacías demuestran tener un lugar importante para la detección de tendencias suicidas, pues la exactitud en la mayoría de los experimentos aumentó cuando se mantienen. El suicidio es un problema de salud que se ha incrementado en los últimos años y debe abordarse desde una perspectiva multidisciplinaria incluyendo tecnologías de internet, comunicación, psicologías, tecnologías del lenguaje. Estas últimas pueden ayudar a través del análisis de textos a la detección de pensamientos y comportamientos suicidas expresados a través de redes sociales. Las palabras empleadas y la manera en la que las persona se comunican en sus blogs o redes sociales proporcionan información sobre su estado psicológico y sobre su personalidad. El procesamiento y análisis de los textos que una persona comparte en internet pueden ayudar a detectar cambios emocionales. Si bien en este trabajo se obtuvo una exactitud mayor al 74% para diferenciar textos con tendencias suicidas de los que no las presentan se necesitan realizar experimentos adicionales con textos extraídos de internet, mismos que presentarán retos adicionales pues dejarán de contar con la buena redacción y estructura de los poemas.

Referencias

1. Instituto Nacional de Estadística y Geografía: Estadísticas a propósito del Día Mundial para la prevención del suicidio. *INEGI*. https://www.inegi.org.mx/contenidos/saladeprensa/aproposito/2017/suicidios2017_Nal.pdf. (2017). Accedido el 1 de julio de 2020
2. Instituto Nacional de Estadística y Geografía: Estadísticas a propósito del Día Mundial para la prevención del suicidio. *INEGI*. https://www.inegi.org.mx/contenidos/saladeprensa/aproposito/2019/suicidios2019_Nal.pdf. (2019). Accedido el 1 de agosto de 2020
3. Castro-Silva E.; Benjet C., Juárez-García F., Jurado-Cárdenas S., Lucio-Gómez-Maqueo M.E., Valencia-Cruz A.: Non-suicidal self-injuries in a simple of Mexican university students. Revista Salud Mental. Vol. 40, No. 5, pp. 191-199 (2017)
4. Organización Mundial de la Salud: Plan de acción sobre salud mental 2013-2020. *Organización Mundial de la Salud* http://apps.who.int/iris/bitstream/handle/10665/97488/9789243506029_spa.pdf. Accedido el 8 de agosto de 2020
5. Patiño-González D.: Analizan fenómeno del suicidio en México. *CienciaMx*. http://www.cienciamx.com/index.php/ciencia/salud/18118-analizan-fenomeno-suicidio-mexico (2017). Accedido el 21 de mayo de 2020
6. Hajime S.: The association of suicide-related Twitter use with suicidal behaviour: A cross-sectional study of young internet users in Japan. *Journal of Affective Disorders*. Vol. 170, pp 155-160 (2015)
7. O'Dea B., Wan S.; J.-Batterham P.; L.-Calear A.; Paris C.; Christensen H.: Detecting suicidality on Twitter. *Internet Interventions*. Vol. 2, No. 2, pp 183-188 (2015)
8. Pestian J.; Nasrallah H.; Matykiewicz P.; Bennett A.; Leenaars A.: Suicide Note Classification Using Natural Language Processing: A Content Analysis. *Biomedical Informatics Insights*. Vol. 3, pp 19-28 (2010)
9. F.-Gunn J.; Lester D.: Using google searches on the internet to monitor suicidal behavior. *Journal of Affective Disorders*. Vol. 148, No. 2-3, pp 411-412 (2013)
10. Mulholland M.; Quinn J.: Suicidal Tendencies: The Automatic Classification of Suicidal and Non-Suicidal Lyricists Using NLP. *International Joint Conference on Natural Language Processing*. pp 680-684 (2013)
11. Zhong Q.; Mittal L.P; Nathan M.; Brown K.; Knudson-González D.; Cai T.; Finan S.; Gelaye B.; Avillach P.; Smoller J.W.; Karlson E.; Cai T.; Williams M.: Use of natural language processing in electronic medical records to identify pregnant women with suicidal behavior:

towards a solution to the complex classification problem. *European Journal of Epidemiology*. Vol. 34, No. 2, pp 153-162 (2019)

12. Sánchez-Arenas J.L.: Detección de Tendencias Suicidas Utilizando Procesamiento de Lenguaje Natural, Tesis de Licenciatura, Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla. (2017)

Efectos del Ajuste Paramétrico en la Detección del Movimiento para un MOG

Antonio Trejo-Morales[1], Diana-Margarita Córdova-Esparza[1],
Hugo Jiménez-Hernández[2]
[1] Universidad Autónoma de Querétaro, Facultad de Informática,
Av. de las Ciencias S/N, Campus Juriquilla, Querétaro, C.P. 76230, México.
[2] Laboratorio de Modelado, Simulado y Visión por Computadora, CIDESI.
Av. Playa Pie de la Cuesta No.702, Desarrollo San Pablo, Querétaro, C.P. 76125, México
antonio.trj.mrls@gmail.com

Resumen. El proceso de sustracción de fondo consiste en diferenciar los objetos fijos y en movimiento. El rendimiento de un método de sustracción de fondo depende de las variables del escenario y los parámetros del modelo. El presente trabajo tiene como objetivo la descripción de los efectos paramétricos del ajuste, para el modelo de sustracción de fondo de mezcla de gaussianas, aplicado a un video de monitoreo vial. Los parámetros utilizados consisten en los filtros espaciales para el proceso de adquisición de la imagen, así como para la detección del movimiento y la velocidad de ajuste en la estimación del fondo. En los resultados se aprecia como la combinación óptima de parámetros ayuda a detectar de forma más eficiente el movimiento.

Palabras Clave: Movimiento, Mezcla de Gaussianas, Criterio Paramétrico.

1 Introducción

La sustracción de fondo para la identificación y seguimiento de objetos en secuencias de vídeo juega un papel importante, es una técnica fundamental y muy crítica, a menudo, lograr un buen resultado es una tarea complicada, debido a diversos factores que influyen en la calidad de las secuencias de vídeo, por ejemplo: cambios drásticos de luminosidad [1], movimientos periódicos en las cámaras que provocan ruido [2], fondo no estático [3], como el reflejo de la luz del agua, el movimiento de las ramas de los árboles, sombras [4], condiciones climáticas, los cuales no permiten el modelado ideal del fondo.

A consecuencia de esta problemática, se han generado varios trabajos disponibles en la literatura actual, dando como resultado el método de mezcla de gaussianas [5], convirtiéndose en el más popular para la sustracción de fondo [10], dado que controla los cambios de luminosidad, movimientos periódicos y ruido de las cámaras. Sin embargo la detección del movimiento se vuelve inestable en la precisión, ya que es un método muy sensible a la estimación paramétrica con la que el usuario decida trabajar.

Es aquí donde nace la oportunidad de esta investigación, ya que una inadecuada parametrización en el modelo de fondo, afecta considerablemente la percepción de los objetos detectados, debido a que se altera drásticamente en tamaño y forma la detección de los objetos. Por lo tanto, la presente investigación se centra en proponer un criterio de parametrización ideal comúnmente utilizado en secuencias de vídeo preferentemente en escala de grises, o espacio de color transformado a escala de grises [10].

Este trabajo de investigación está organizado de la siguiente manera: la sección 2, describe los trabajos relacionados y relevantes al modelo de fondo a partir de la mezcla de gaussianas; en la sección 3, se describe la propuesta como método fundamental para la sustracción de fondo y los parámetros que lo integran; en la sección 4, se presenta el criterio de parametrización propuesto en conjunto con los resultados; finalmente en la sección 5, se informan las conclusiones.

2 Trabajos relacionados

Es importante mencionar algunas variantes del trabajo de [5], referente a la mezcla de gaussianas existentes en la literatura, y que se ha convertido en un elemento frecuente para los sistemas de videovigilancia. Iniciando cronológicamente con la investigación de [2], quién propuso un sistema de tres componentes para el mantenimiento del fondo capaz de descartar movimientos periódicos. Además, este enfoque se limita a escenarios con condiciones de iluminación fijas.

Posteriormente, el trabajo de [6] se basa en el algoritmo de [5], para realizar el modelo de fondo y proponer un novedoso algoritmo basado en bloques para la sustracción de fondo, con el objetivo de identificar regiones de un fotograma de vídeo que contienen objetos en movimiento, pero en lugar de utilizar el vector de color o valor de niveles de gris, se utiliza una medida de textura del patrón binario local (LBP, por sus siglas en inglés), debido a que es invariante a cambios monotónicos en escala de grises; opera en tiempo real bajo el supuesto de una cámara fija con distancia focal fija.

En [7] se utiliza un modelo de mezcla de K gaussianas, donde K es una constante predefinida para cada píxel. Propone un esquema eficaz para mejorar la tasa de convergencia sin comprometer la estabilidad del modelo, reemplazando el factor de retención estático global con una tasa de procesamiento adaptativa calculada para cada gaussiana en cada imagen.

En el trabajo [8] se propone un procedimiento de esperanza-maximización (EM) modificado para construir el modelo de mezcla gaussianas con un número de componentes K adaptativo (AKGMM, por sus siglas en inglés). Desarrolla una regla de selección de componentes de fondo heurística para tomar una decisión de clasificación de píxeles basada en el modelo propuesto.

El estudio de [11], propone una representación y discriminación de características basadas en densidades de probabilidad del modelo estadístico de mezcla gaussiana, se hace una revisión y se discuten los métodos de estimación de máxima verosimilitud para los modelos de mezcla gaussiana, propone el uso de información de confianza en la clasificación y se presenta un método para estimar la confianza, ya que el problema central en los métodos estadísticos es la estimación de funciones de densidad de probabilidad condicional.

Otro enfoque es el trabajo de [9] quien propone una técnica que utiliza el algoritmo de mezcla de Gaussianas para detectar cambios estructurales, incluidos los inducidos por ruido, variación de la iluminación, mediante una cascada de pruebas.

Finalmente, el trabajo de [10] se centra en aumentar la robustez del modelo de mezcla de gaussianas para entornos complejos. Su propuesta consiste en redefinir los parámetros de distribución, como el peso, la media y la varianza, es un modelo basado en regiones que elimina el efecto del ruido causado por cambios drásticos en la iluminación.

3 Propuesta

La propuesta es encontrar precisamente los parámetros que hacen que el modelo de fondo se ajuste correctamente para la detección de los objetos (ver Fig. 1). Así como también mostrar el efecto de los parámetros en la detección. Variando los valores de éstos, comparar los resultados, y de acuerdo a situaciones concretas, ilustrar qué combinación es la que mejor resultado arroja.

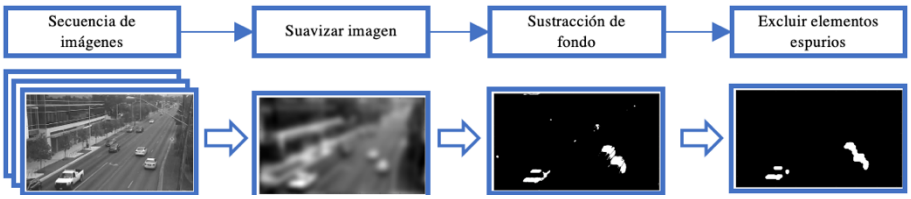


Fig. 1. Criterio a seguir en la parametrización, para encontrar un ajuste de convergencia.

3.1 Suavizar imagen

En general, es común encontrar en la literatura diferentes técnicas de filtrado en el dominio de la frecuencia, para suavizar la imagen y lograr disminuir las variaciones de intensidad entre los píxeles vecinos, diferenciándose según el resultado esperado [13,14]: comenzando por los filtros paso-bajo, su propósito de implementación es para la reducción de ruido, suavizar y aplanar las imágenes, atenuando o eliminando las componentes de alta frecuencia y a la vez dejando sin modificar las bajas frecuencias, con el inconveniente de que se reduce o se pierde la nitidez; los filtros paso-alto, son comúnmente utilizados para detectar cambios en la luminosidad, detectar bordes o para resaltar detalles finos en una imagen, eliminan las componentes de baja frecuencia y debilitan las componentes de alta frecuencia; y los filtros paso-banda, también son utilizados para detectar patrones de ruido, con el inconveniente de que este tipo de filtro elimina demasiado contenido de una imagen.

De acuerdo a esta información, se procede a implementar el filtro paso-bajo basado en la distribución normal o gaussiana. Este tipo de función de transferencia simula una distribución gaussiana, donde el valor máximo de uno aparece en el centro de la campana, y luego su valor comienza a disminuir según la función gaussiana, cuya dispersión o apertura está dada por el parámetro sigma σ, definido por la frecuencia de corte D_0 (ver ecuación (1)). El resultado final en el espectro de la frecuencia será un conjunto de valores entre cero y uno.

$$H(u,v) = e^{-D^2(u,v)/2D_0^2} \quad \textbf{(1)}$$

A continuación, se muestra el resultado de este primer planteamiento (ver Fig. 2), se puede observar la imagen resultante al aplicar el filtro paso-bajo basado en la distribución normal o gaussiana, es importante resaltar que se está buscando suprimir el ruido de la imagen para mejorar la detección de características pequeñas, por lo tanto, se elige un valor de sigma de manera experimental, para hacer que el gaussiano sea más pequeño que la característica. Por ejemplo, para un tamaño de filtro = [15 15], se obtiene un valor de sigma = 4.

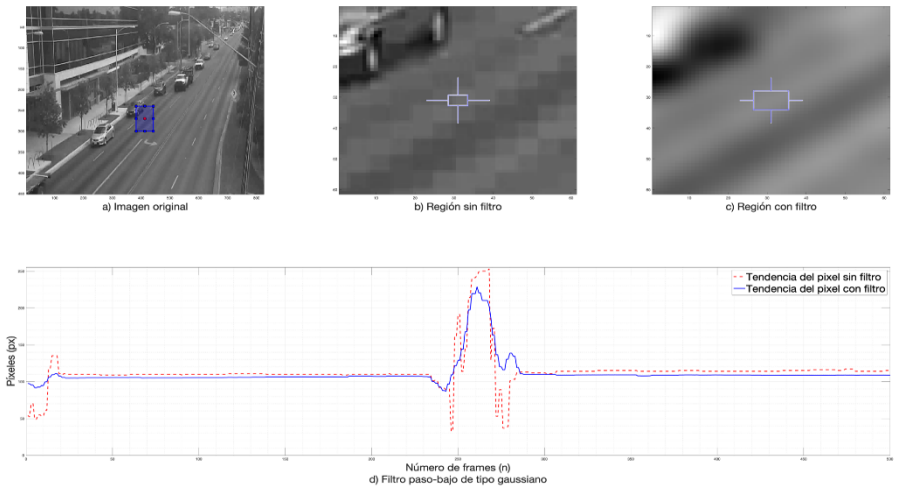


Fig. 2. Efecto del filtro de paso-bajo de tipo gaussiano para la reducción de ruido, suavizar y aplanar la imagen con parámetros estipulados por el usuario para la prueba: imagen original en la opción a, la región de la imagen sin filtro en la opción b, la región atenuada con el filtro aplicado en la opción c, con un tamaño de filtro = [15 15] y un valor de sigma = 4, finalmente en la opción d, la gráfica que representa la tendencia de los valores de los píxeles con filtro y sin filtro.

3.2 Modelo de fondo

Como ya se mencionó en las secciones 1 y 2, el método propuesto por [5], se utiliza eventualmente en algoritmos de análisis de videovigilancia para detectar y analizar ciertas situaciones de especial interés. Sin embargo, un problema común de este método es lograr una parametrización aceptable como lo expresa [10]. Sin más preámbulo, este método en lugar de utilizar una gaussiana para modelar la intensidad de un píxel $X = [x_0, y_0]$, en una secuencia de imágenes $I^t = \{I_1, I_2, \dots, I_n\}$, para una serie de tiempo, se basa en una función de probabilidad (ver ecuación (2)), que utiliza k gaussianas, como el número de distribuciones para cada píxel, donde típicamente k toma valores entre 3 y 5. ω_i^t es el peso estimado, es decir, qué cantidad de los datos serán considerados por este modelo, μ_i^t es la media i-ésima o valor medio i-ésimo en el tiempo t, Σ_i^t es la matriz de covarianzas de la mezcla de gaussianas en un momento t, θ^t representa el vector de parámetros en el modelo $\theta = (\omega_1^t, \dots, \omega_k^t, \mu_1^t, \dots, \mu_k^t, \Sigma_i^t)$ para un momento t, y $G(\,)$ es la función gaussiana de densidad de probabilidad.

$$P(X^t|\theta^t) = \sum_{i=1}^{k} \omega_i^t * G(X^t|\mu_i^t, \Sigma_i^t) \quad \textbf{(2)}$$

El resultado de aplicar el modelo de fondo de la mezcla de gaussianas se muestra en la Fig. 3 y de acuerdo a la ecuación (2), los parámetros controlados de manera experimental son los siguientes: el número de gaussianas $k = 5$, el cual permite modelar múltiples comportamientos de fondo; el peso estimado $\omega = 150$, que representa el número de imágenes de vídeo iniciales a procesar; la tasa de procesamiento $ts = 0.005$, que controla la rapidez con la que el modelo se adapta a las condiciones cambiantes, como la iluminación; la matriz de covarianza $\sigma^2 = 48^2$; y el umbral $th = 0.845$, para representar la posibilidad mínima de que los píxeles se consideren parte del fondo.

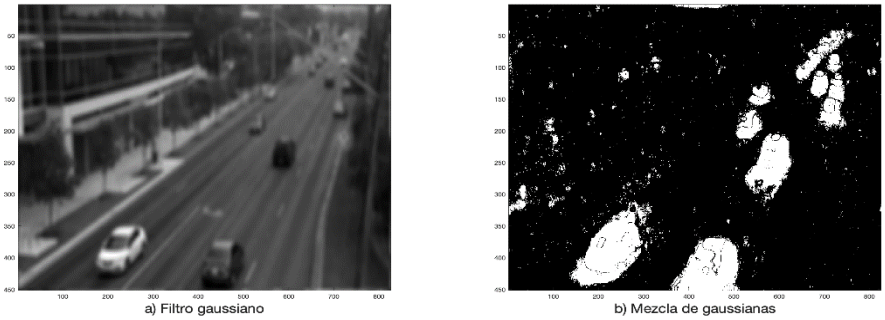


Fig. 3. Imagen de secuencia de video, izquierda, e imagen derecha, la sustracción de fondo utilizando el modelo de mezcla de gaussianas [5], aplicando los parámetros equivalentes a: $k = 5,\ \omega = 150,\ ts = 0.005,\ \sigma^2 = 48^2\ y\ th = 0.845.$

Es importante mencionar que al aplicar el modelo de fondo se aprecian perturbaciones en el resultado, estas perturbaciones es un tipo de ruido intrínseco derivado del proceso, típicamente representa a los objetos espurios. El paso siguiente es observar qué tipo de forma tiene ese ruido, analizar el vecindario para conocer el comportamiento a su alrededor y proceder a su eliminación.

3.3 Excluir elementos espurios

La morfología en términos de procesamiento de imágenes, es el estudio de la estructura de los objetos a partir de sus respectivas imágenes para describir su propia forma. Para aplicar el proceso morfológico a una imagen, se utilizan operaciones morfológicas que permiten obtener una descripción de las estructuras geométricas de los objetos [15], como contornos, hoyos, esquinas, grietas, entre otras. Las operaciones básicas son: la dilatación y la erosión, y su combinación entre ellas, generando como resultado los términos de operación para apertura y cierre, operando espacialmente sobre una imagen para realce de características o eliminación. La erosión del conjunto B por el elemento estructural E, denotada como $B \otimes E$, se aplica para ampliar las regiones de la imagen con un valor igual a 1 y de color blanco. La dilatación del conjunto B por un elemento estructural E, denotada por $B \oplus E$, reduce las regiones de la imagen con un valor igual a 0 y de color negro. La apertura de un conjunto B por un elemento estructural E, denotada por $(B \otimes E) \oplus E$, es una combinación de aplicar la operación de erosión y después la dilatación, el efecto es eliminar los puntos o elementos finos. Y cierre de B por el elemento estructural E, denotado por $(B \oplus E) \otimes E$, aplica la dilatación y posteriormente la erosión, su efecto es rellenar los huecos negros de cierto tamaño.

Tomando en cuenta a la morfología matemática, surgen los filtros morfológicos para representar a la transformación local de características geométricas, satisfaciendo la propiedad de invariabilidad a traslación [14, 15], típicamente son aplicados a imágenes binarias para la eliminación de elementos indeseados en la imagen. Los filtros utilizan un parámetro llamado elemento estructurante de cierto tamaño y forma, se elige a priori, de acuerdo con la morfología sobre la que se va aplicar y en función de la obtención de las formas que se desea extraer.

A partir de los operadores morfológicos se procede a realizar el procesamiento posterior al modelo de fondo (ver Fig. 4), mediante los filtros compuestos de apertura y cierre, con el objetivo de excluir los valores espurios contenidos en la imagen.

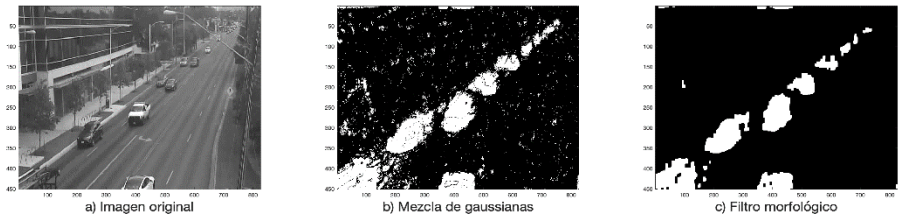


Fig. 4. Imagen del modelo de fondo, izquierda, e imagen derecha, resultado de aplicar los filtros morfológicos de apertura y cierre, utilizando un elemento estructurante de tamaño = 7 píxeles y la forma = Cuadrado.

3.4 Criterio de parametrización

El criterio de parametrización se basa en el análisis de los parámetros de las técnicas para detectar factores externos al movimiento (ver Tabla 2). Por ejemplo, para suavizar la imagen y atenuar el ruido, se recomienda variar el tamaño del filtro en función del valor de la desviación estándar de la distribución gaussiana, y así, lograr una reducción de cambios bruscos en la iluminación presentada en cada iteración.

Para la variación en los parámetros del modelo de mezcla de gaussianas, se sugiere hacer variaciones en: el número de gaussianas, disminuir el valor únicamente si el fondo es estático. Por otro lado, aumentar el valor ayuda cuando el fondo no está constante, por ejemplo, cuando hay movimiento de las hojas de los árboles; establecer un valor considerablemente aceptable para el número de fotogramas de procesamiento; variar el valor de la tasa de procesamiento (ts), para controlar la rapidez con la que el modelo de fondo se adapta a los cambios de iluminación. Si ts es muy alto, los objetos que se mueven lentamente pueden convertirse en parte del fondo. Si ts es demasiado bajo, el modelo de fondo no podrá adaptarse a los cambios de iluminación; establecer un umbral mínimo para determinar el modelo de fondo, y así, lograr que los píxeles se consideren valores de fondo; y definir un valor para la varianza del modelo de mezcla.

Finalmente, hacer un ajuste en el parámetro del filtro removedor de elementos espurios, haciendo variaciones al valor del elemento estructurante, y estableciendo la forma en 2D para el elemento, las más comunes son: cuadrada, rectangular, disco, etc.

Tabla 2. Resumen de los parámetros de ajuste para cada técnica utilizada.

Técnica	Parámetro	Valor predeterminado	Valor propuesto
Filtro gaussiano	Tamaño del filtro	[3 3]	[15 15]
	Desviación estándar de la distribución gaussiana (sigma)	0.5	4
Mezcla de Gaussianas	Número de gaussianas (k)	3	5
	Tasa de procesamiento (ts)	0.005	0.005
	Varianza inicial (σ^2)	30^2	48^2
	Relación de fondo mínima (th)	0.7	0.84
	Fotogramas de procesamiento (ω)	150	500
Filtro morfológico	Tamaño del elemento estructurante	4 píxeles	8 píxeles
	Forma del elemento estructurante	Disco	Cuadrado

A continuación, se muestran los resultados de aplicar los parámetros predeterminados con los que cuenta cada técnica (ver Fig. 5), se observa que en el resultado que arroja el filtro morfológico, aparecen elementos pequeños que en la realidad no deberían considerarse como objetos. Por otra parte, cuando se modifican los parámetros experimentalmente,

para remover los objetos espurios (ver Fig. 6), se observa que algunos objetos disminuyen su tamaño, e inclusive otros se enlazan entre sí, este fenómeno se debe a una configuración de parámetros erróneos.

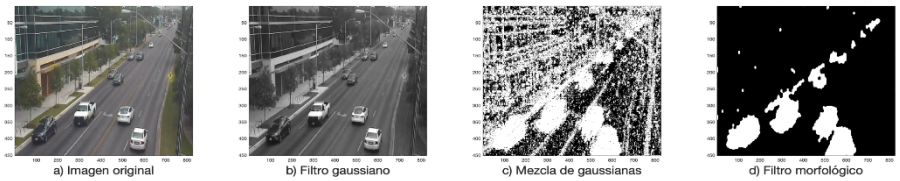


Fig. 5. Resultado de aplicar los parámetros predeterminados para cada técnica.

Para ambas figuras (Fig. 5 y Fig. 6), la primera imagen representa: a) la imagen original, la segunda imagen b) el resultado de aplicar el filtro paso-bajo, la tercera imagen c) muestra el resultado del modelo de fondo, y finalmente en la última imagen d) se ilustran los objetos en movimiento.

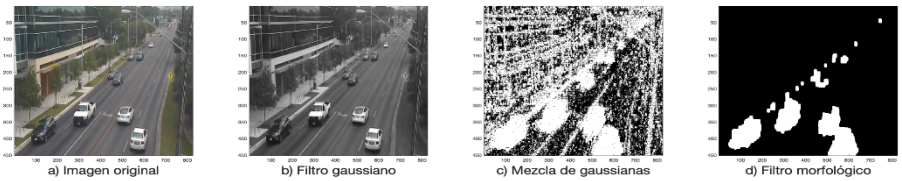


Fig. 6. Resultado de aplicar los parámetros propuestos para cada técnica, con el objetivo de excluir los elementos espurios.

4 Resultados

Se implementaron las técnicas descritas en la sección 3, haciendo distintas variaciones experimentales en los parámetros de entrada (ver Tabla 3). Se utilizó la secuencia de video que provee el Centro de Computación Avanzada de Texas (TACC) [16].

Tabla 3. Resumen de los parámetros finales de ajuste aplicados a cada técnica.

Técnica	Parámetro	Valor final
Filtro gaussiano	Tamaño del filtro	[17 17]
	Desviación estándar de la distribución gaussiana (sigma)	4.0
Mezcla de Gaussianas	Número de gaussianas (k)	6
	Tasa de procesamiento (ts)	0.005
	Varianza inicial (σ^2)	48^2
	Relación de fondo mínima (th)	0.845
	Fotogramas de procesamiento (ω)	750
Filtro morfológico	Tamaño del elemento estructurante	7 píxeles
	Forma del elemento estructurante	Cuadrado

Una vez establecidos los parámetros para cada técnica, se realiza la detección de los objetos, obteniendo los siguientes resultados (ver Fig. 7). Como puede observarse, el criterio propuesto se adapta rápidamente a cambios drásticos en la iluminación y fondos dinámicos para detectar objetos reales en movimiento. Es importante señalar que en esta

propuesta se difumina el ruido de fondo de baja frecuencia, debido a la presencia de movimiento ocasionado por las ráfagas de viento y ondas de calor.

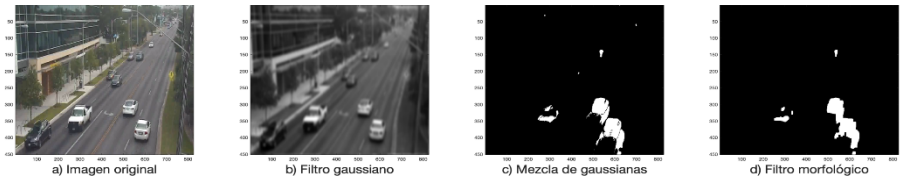


Fig. 7. Resultado de aplicar los parámetros finales en cada técnica. La primera columna representa a) la imagen original, la segunda columna b) el resultado de aplicar el filtro paso-bajo de tipo gaussiano, la tercera columna c) muestra el resultado del modelo de fondo, finalmente en la última columna d) se ilustran los objetos en movimiento.

En general, los resultados son buenos por las siguientes razones: 1) la inclusión de un filtro paso-bajo que elimina cambios drásticos de iluminación; 2) se confirmó que las ráfagas de viento, los movimientos de la cámara, el movimiento del sol y los árboles, afectan considerablemente a los métodos de sustracción de fondo; 3) el aplicar un criterio de intensidades mayores a un valor dado en píxeles y aplicar un proceso morfológico al resultado de la mezcla de gaussianas, también ayuda a obtener mejores resultados. Finalmente, la característica más importante de esta propuesta, es que este criterio se puede adaptar para cualquier aplicación de modelado de fondo.

5 Conclusiones y trabajos futuros

Se puede concluir que el método de sustracción de fondo basado en una mezcla de gaussianas realiza adecuadamente su tarea, aunque exhibe algunas deficiencias en su parametrización. En este trabajo se proponen algunas técnicas simples que mejoran notablemente los resultados actuales reportados en la literatura. En particular, este trabajo deja abierta la posibilidad de seguir experimentando con los parámetros para estos métodos, así como, determinar y aplicar una manera más objetiva para su evaluación y comparación. Finalmente, adicionar una técnica para la detección y eliminación de sombras, para lograr aumentar la robustez en el modelo de fondo.

Referencias

1. Huwer, S.; Niemann, H.: Adaptive change detection for real-time surveillance applications. In: Proceedings Third IEEE International Workshop on Visual Surveillance, Dublin, Ireland, pp. 37-46 (2000), https://doi.org/10.1109/VS.2000.856856.
2. Toyama, K.; Krumm, J.; Brumitt, B.; Meyers, B.: Wallflower: Principles and practice of background maintenance. In: Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, vol.1, pp. 255-261 (1999), https://doi.org/10.1109/ICCV.1999.791228.
3. Ridder, C.; Munkelt, O.; Kirchner, H.: Adaptive Background Estimation and Fore-ground Detection using Kalman-Filtering. In: Proceedings of International Conference on recent Advances in Mechatronics (ICRAM'95), pp. 193–199 (1995).
4. Barbuzza, R.; Dominguez, L.; Perez, A.; Esteberena, L.F.; Rubiales, A.; D'amato J.P.: A Shadow Removal Approach for a Background Subtraction Algorithm. In: De Giusti A. (eds) Computer Science – CACIC 2017. Communications in Computer and Information Science, vol 790. Springer, pp. 101-110 (2018), https://doi.org/10.1007/978-3-319-75214-3 10.
5. Stauffer, C.; Grimson, W.: Adaptive background mixture models for real-time tracking. In: 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149), Fort Collins, CO, USA, Vol. 2, pp. 246-252 (1999), https://doi.org/10.1109/CVPR.1999.784637. 6.

6. Heikkilä, M.; Pietiläinen, M.; Heikkilä, J.: A texture based method for detecting moving objects. In: 15th British Machine Vision Conference (BMVC 2004), pp. 187–196 (2004), https://doi.org/10.5244/C.18.21.
7. Lee, D.S.: Effective Gaussian mixture learning for video background subtraction. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 5, pp. 827-832 (2005), https://doi.org/10.1109/TPAMI.2005.102.
8. Tan, R.; Huo, H.; Qian, J.; Fang, T.: Traffic Video Segmentation Using Adaptive-K Gaussian Mixture Model. In: Advances in Machine Vision, Image Processing, and Pattern Analysis. IWICPAS 2006. Lecture Notes in Computer Science, vol. 4153. Springer, Berlin, pp. 125-134 (2006), https://doi.org/10.1007/11821045_13.
9. Teixeira, L.; Corte-Real, L.: Cascaded change detection for foreground segmentation. In: 2007 IEEE Workshop on Motion and Video Computing (WMVC'07), Austin, TX, USA, pp. 6-6 (2007), https://doi.org/10.1109/WMVC.2007.11.
10. Santoyo-Morales, J.E.; Hasimoto-Beltran, R...: Video Background Subtraction in Complex Environments. In: Journal of Applied Research and Technology, Volume 12, Issue 3, pp. 527-537 (2014), https://doi.org/10.1016/S1665-6423(14)71632-3.
11. Paalanen, P.; Kämäräinen, J.; Ilonen, J.; Kälviäinen, H.: Feature representation and discrimination based on Gaussian mixture model probability densities—Practices and algorithms. In: Pattern Recognition, Volume 39, Issue 7, pp. 1346-1358 (2006), https://doi.org/10.1016/j.patcog.2006.01.005.
12. May, P.; Ehrlich, H.C.; Steinke, T.: ZIB Structure Prediction Pipeline: Composing a Complex Biological Workflow through Web Services. In: Nagel, W.E., Walter, W.V., Lehner, W. (eds.) Euro-Par 2006. LNCS, vol. 4128, Springer, Heidelberg, pp. 1148-1158 (2006).
13. González, R.C.; Woods R.E.: Digital Image Processing. Pearson, Prentice Hall. ISBN-13: 9780131687288, Third Edition, pp. 174-179 (2007).
14. González, R.C., Wintz, P.: Procesamiento digital de imágenes. Addison Wesley, pp. 89-269 (1996).
15. Serra, J.: Introduction to mathematical morphology. In: Computer Vision, Graphics, and Image Processing, Volume 35, Issue 3, pp. 283-305 (1986), https://doi.org/10.1016/0734-189X(86)90002-2.
16. Dubrow, A.: Artificial intelligence and supercomputers to help alleviate urban traffic problems. In: Texas Advanced Computing Center, (2017). https://www.tacc.utexas.edu/-/artificial-intelligence-and-supercomputers-to-help-alleviate-urban-traffic-problems

Análisis de la Deserción Escolar Mediante Técnicas de Minería de Datos

Alma A. Pedro[1], Marisol Contreras[1], Jasiel H. Toscano[2], Eduardo Sanchez[3], Salvador E. Lobato[1]

[1] Instituto de Estudios Ambientales, Universidad de la Sierra Juárez, Avenida Universidad s/n. 68725 Ixtlán de Juárez, Oaxaca, México

[2] Departamento Ciencia de la Computación, Pontificia Universidad Católica de Chile, Avenida Vicuña Mackenna 4860 Santiago de Chile

[3] Instituto de Computación, Universidad Tecnológica de la Mixteca, Carretera a Acatlima Km.2.5. 69000 Huajuapan de León, Oaxaca, México

[1]{almaalheli, li01140015, salvadorl}@unsij.edu.mx, [2]jhtoscano@uc.cl, [3]esanchez@mixteco.utm.mx

Resumen. En este artículo se describe el análisis de variables relacionadas a la deserción escolar, mediante técnicas de Minería de Datos. La información se obtuvo de un cuestionario aplicado a alumnos activos, egresados y a alumnos que han desertado del programa de licenciatura en informática. El cuestionario agrupa información de cuatro categorías: personal, familiar, económica y académica. La información fue tratada para homogeneizar y discretizar las variables. Posteriormente, se realizó un análisis de componentes principales para explorar y reducir la dimensión de los datos. Finalmente, se aplicaron algoritmos de árboles de decisión (J48, RandomTree y REPTree) para obtener patrones que describan el comportamiento de las variables. Los resultados muestran que la precisión del algoritmo REPTree es mayor en comparación con J48 y RandomTree, por esta razón es más recomendable para describir el comportamiento de las variables de la deserción escolar. Estos resultados, forman las bases para desarrollar un sistema computacional para predecir la deserción escolar.

Palabras Clave: Deserción Escolar, Minería de Datos, J48, RandomTree, REPTree, Weka, PCA, R.

1 Introducción

La deserción escolar universitaria es uno de los principales problemas que enfrentan los sistemas educativos a nivel internacional [1]. En México de acuerdo al indicador por cohorte, de cada 100 alumnos que ingresan a algún programa universitario el 42 % deserta, esto significa que cerca de la mitad de los alumnos que ingresan desertan de la educación superior. Este problema tiene grandes afectaciones en el aspecto social, académico y económico de las familias, los estudiantes, las instituciones y también del Estado [2].

En la actualidad la definición de la deserción escolar sigue en discusión. Sin embargo, los investigadores en consenso la definen de la siguiente forma: “un abandono que puede ser explicado por diferentes categorías de variables: socioeconómicas, individuales, institucionales y académicas” [2]. La Secretaria de Educación Pública (SEP) realiza una definición al respecto y describe a la deserción escolar como “el abandono de las actividades escolares antes de terminar algún grado o nivel educativo” [3].

El problema de la deserción escolar ha sido objetivo central de diversas investigaciones y por esta razón se ha estudiado desde distintos enfoques [1, 4-7]. En el caso específico de México los estudios enfocados a esta problemática han identificado

causas universales como: las presiones económicas familiares, las dificultades de integración, la inadecuada orientación escolar, la reprobación escolar reincidente, problemas de salud, la edad de ingreso y el traslape de horarios estudios-trabajo [8]. En nuestro país esta realidad se agudiza con mayor profundidad en los sistemas educativos de carácter público [9]. La gran parte de los estudiantes que acuden a las universidades públicas provienen de grupos socioculturales y académicos más pobres [10].

A nivel nacional, Oaxaca es considerado uno de los estados con mayor rezago social y marginación. De acuerdo a las estadísticas presentadas por [9] en el año 2010, el 62 % de la población tiene educación básica terminada, el 14 % concluyó la educación media superior y únicamente un 10 % termina la educación superior, estas cifras muestran que es bajo el porcentaje de personas que culminan con su preparación profesional, para ser exactos uno de cada diez oaxaqueños concluyen estudios de educación superior.

Si bien es cierto que el nivel de rezago social y la marginación son factores que inciden directamente en el nivel de educación de las personas, también, es necesario comprender la naturaleza de otros factores debido a que el problema de la deserción escolar es multifactorial, es importante analizar los motivos que llevan a los estudiantes a abandonar sus estudios. Para esto es necesario analizar información de alumnos que llevan a buen término sus estudios y también información de los alumnos que se han dado de baja. Dado que son múltiples variables relacionadas a este problema, los datos presentan una alta dimensionalidad y se requiere la aplicación de técnicas para encontrar información útil que ayude a determinar reglas que describan el problema de la deserción escolar [11].

De acuerdo a la problemática planteada en los párrafos anteriores, en este artículo se describe el análisis de las variables involucradas en el problema de la deserción escolar mediante la aplicación de técnicas de Minería de datos. Es importante mencionar que en términos de la deserción escolar en el nivel superior, en el estado de Oaxaca ha sido mínima la exploración de este problema. En este trabajo el análisis de la deserción se especifica en una universidad de carácter público con una matrícula en su mayoría proveniente de comunidades con altos índices de marginación.

Los resultados de este trabajo forman las bases para desarrollar un sistema computacional que ayude a predecir la deserción escolar en una etapa temprana, antes de que el alumno deserte del programa de estudios. La información del sistema ayudará a identificar cuáles son los elementos a reforzar para evitar la deserción y crear alternativas de solución al problema. Este trabajo está estructurado de la siguiente forma: en la sección 2 se mencionan algunos trabajos relacionados al tema, la sección 3 detalla la metodología para llevar a cabo el análisis de la deserción escolar, en la sección 4 se muestran los resultados obtenidos y finalmente, en la sección 5 se presentan las conclusiones derivadas de este trabajo.

2 Trabajos relacionados

La deserción escolar ha sido tema de diversas investigaciones que se han centrado en cinco enfoques de estudio, los cuales son: psicológico, sociológico, economicista, organizacional e interaccionista [1] [5]. Estos enfoques se han estudiado de forma aislada y por esta razón en los últimos años surgen estudios de carácter holístico que involucran diversas variables de acuerdo a las características del sujeto y su contexto [5] [12]. La gran parte de las investigaciones realizadas en torno a este problema, centran su atención en responder cuáles son las causas que provocan las grandes tasas de deserción escolar y las variables que se involucran en este problema de acuerdo a cada contexto [4][3]. Sin embargo, poco se ha explorado en el análisis de las variables involucradas en la deserción escolar aplicando técnicas de Minería de Datos.

Los resultados que ofrecen estas técnicas permiten analizar el comportamiento y la importancia de las variables [13] [14]. En [11] se desarrolla la aplicación de técnicas para la exploración de los datos como el análisis de componentes principales (ACP) y también

métodos como los árboles de decisión para la predicción del fracaso escolar, los resultados que se obtienen muestran que los árboles de decisión con recomendables para el análisis de datos de deserción escolar.

3 Metodología

La metodología propuesta involucra los siguientes pasos: análisis exploratorio de archivos, diseño de la herramienta para obtener información, obtención de la información, preparación de los datos y análisis de los datos. Los resultados derivados de la etapa de análisis forman las bases para el desarrollo de un sistema para predecir la deserción escolar. En la Fig. 1 se detalla el esquema general de la metodología implementada.

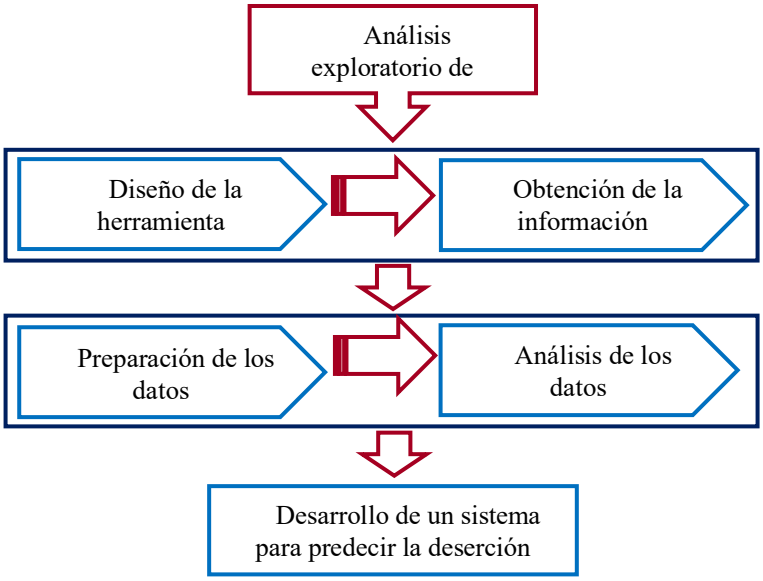


Fig. 1. Metodología general para el análisis de la deserción general.

3.1 Análisis exploratorio de archivos

Inicialmente se realizó un análisis exploratorio de los archivos de baja de alumnos para encontrar variables que explicaran el problema de la deserción escolar del programa de licenciatura en informática. Las variables que se extrajeron se muestran se muestran en la Tabla 1.

Tabla 1. Variables extraídas de los archivos escolares

No.	Variable	No.	
1	Edad	9	Motivos de baja
2	Sexo	10	Año de ingreso
3	Fecha de nacimiento	11	Año de baja
4	Municipio de procedencia	12	Semestre de baja
5	Ocupación del padre	13	Carrera
6	Ocupación de la madre	14	Promedio anterior o de procedencia
7	Escuela de procedencia	15	Puntaje de examen de selección
8	Especialidad	16	Promedio de curso propedéutico

El número de variables identificadas es de 16, esta es una cantidad limitada para comprender el problema de la deserción escolar en el programa, por esta razón, fue necesario diseñar una herramienta para obtener información de los alumnos considerando aspectos del ámbito personal, familiar, económico y académico del alumno.

3.2 Diseño de la herramienta para obtener información

En el diseño de la herramienta, se consultaron además del archivo de baja de alumnos, los resultados de exámenes de ceneval y solicitudes de becas escolares. También, se realizó un análisis de las variables que se han utilizado en investigaciones relacionadas. La herramienta diseñada consistió en un cuestionario que integra un total de 69 preguntas organizada en cuatro apartados: datos personales del alumno, datos del contexto familiar, datos del contexto socioeconómico y datos del contexto académico. El cuestionario se digitalizo mediante la herramienta de formularios de Google.

3.3 Obtención de información

La información se obtuvo a partir de la aplicación de cuestionarios a alumnos que se encuentran cursando los semestres de tercero a décimo semestre, egresados y también a alumnos que han desertado del programa de licenciatura en informática de la Universidad de la Sierra Juárez. En el caso de los egresados se estableció el contacto vía correo electrónico, para ello se consultó la base de datos de egresados que tiene a resguardo la Jefatura de Carrera. En cuanto a los alumnos que han desertado del programa el contacto fue más complejo al no tener mayor información en los expedientes de baja, se realizó una búsqueda en las listas de correos electrónicos de los profesores, posteriormente se les contacto vía correo electrónico y a través de redes sociales. La obtención de la información de los alumnos que cursan actualmente el programa se realizó a través de sus correos institucionales. En total se obtuvieron 51 registros; 24 corresponden a alumnos activos, 18 a alumnos que se dieron de baja y 9 a egresados del programa.

3.4 Preparación de los datos

Una vez obtenidos los datos, estos se almacenaron en una hoja de cálculo, se simplificaron las preguntas para definirlas como variables (Tabla 2), se verificaron y realizaron correcciones a los datos. Posteriormente, los datos se discretizaron de un formato numérico a formato categórico y de formato nominal a categórico.

Tabla 2. Categorización de variables

Variables de tipo personal	**Variables de tipo familiar**
1. Edad	10. Número de integrantes en la familia
2. Genero	11. Ocupación del padre
3. Estado civil	12. Ocupación de la madre
4. Número de hijos	13. Escolaridad del padre
5. Practica algún deporte	14. Escolaridad de la madre
6. Realiza alguna actividad artística	15. Padres separados
7. Pasatiempo favorito	16. Número de hermanos
8. Considera que eligió la carrera correcta	17. Número de nacimiento
	18. Número de hermanos estudiando
9. Satisfacción por continuar con sus estudios	19. Tiene hermanos cursando una carrera universitaria
	20. Tiene hermanos con licenciatura
	21. Importancia del estudio en el hogar
	22. Primero de la familia en asistir a la universidad
	23. Tipo de problema común en la familia
Variables de tipo económicas	**Variables de carácter académico**

24. Número de personas que dependen del tutor	43. Escuela de procedencia
25. Quién paga sus estudios universitarios	44. Especialidad
26. Está inscrito en alguna beca universitaria	45. Promedio de bachillerato
27. Tiene beca universitaria	46. Reprobó materias en bachillerato
28. Tiene beca del gobierno	47. Número de materias reprobadas en bachillerato
29. Tiene dependientes económicos	48. Tuvo beca de bachillerato
30. Vive con algún familiar	49. Tuvo orientación vocacional
31. Número de personas con quienes vive	50. El programa como primera opción de estudio
32. Vive en casa rentada propia o prestada	51. Motivo de elección de carrera
33. La familia vive en casa propia, rentada o prestada	52. Intentos previos para ingresar a la universidad
34. Tipo de vivienda de la familia	53. Aprobación del Examen EXANI
35. Material de construcción de la casa de la familia	54. Promedio del curso propedéutico
36. Ingreso mensual Familiar	55. Número de materias reprobadas en el curso propedéutico
37. Número de personas que aportaban al ingreso familiar	56. Número de materias aprobadas en el curso propedéutico
38. Cantidad de pago de colegiatura	57. Dominio del idioma ingles al ingresar
39. Trabaja	58. Habla otro idioma o dialecto
40. Horas de trabajo semanalmente	59. Considera que tiene conocimientos necesarios de matemáticas
41. Ingreso por trabajo	60. Tiene hábitos de estudio
42. Cantidad de gastos mensuales	61. Tiene dificultades para estudiar en casa
	62. Tiene espacio específico para estudiar
	63. Tiene apoyo de los padres para mejorar en los estudios
	64. Tiene apoyo de los profesores para resolver dudas de las materias
	65. Acude regularmente a tutorías
	66. Horas al día dedicadas a estudiar fuera de clases
	67. Materia con mayor dominio
	68. Materia en riesgo de reprobar en el primer semestre
	69. Motivo de baja de la licenciatura

3.5 Análisis de los datos

En esta etapa se aplicó la técnica de Análisis de Componentes Principales (PCA) para identificar las variables de mayor representación del problema de la deserción. Este análisis se desarrolló en el software R. El número total de componentes resultantes es de 51 (Tabla 3). Posteriormente, se realizó una selección de los componentes con sus autovalores mayores a 1, el número de componentes seleccionados bajo este criterio es de 23. En la Fig. 2, se muestran de forma gráfica los componentes del 1 al 25, posterior al componente 23 los valores son menores a 1.

Tabla 3. Resultado del análisis de componentes principales

Componentes Principales					
PC1	PC2	PC3	PC4	PC5	PC6
5.433365	5.114822	4.61545	3.872307	3.603424	3.288141

PC7	PC8	PC9	PC10	PC11	PC12
3.200068	3.04904	2.779917	2.43254	2.343699	2.19811
PC13	PC14	PC15	PC16	PC17	PC18
2.098177	1.866843	1.772936	1.64123	1.6211	1.536802
PC19	PC20	PC21	PC22	PC23	PC24
1.258686	1.229283	1.205131	1.0854	1.025008	0.9742785
PC25	PC26	PC27	PC28	PC29	PC30
0.8541977	0.8021706	0.780307	0.7312359	0.6919079	0.6381989
PC31	PC32	PC33	PC34	PC35	PC36
0.5753103	0.5525756	0.5196634	0.4550285	0.4340633	0.4012285
PC37	PC38	PC39	PC40	PC41	PC42
0.3484384	0.3063476	0.2830513	0.2475917	0.2126455	0.1966833
PC43	PC44	PC45	PC46	PC47	PC48
0.1575076	0.1261973	0.1158196	0.09627981	0.08404181	0.05937745
PC49	PC50	PC51			
0.04539975	0.03897338	0.00			

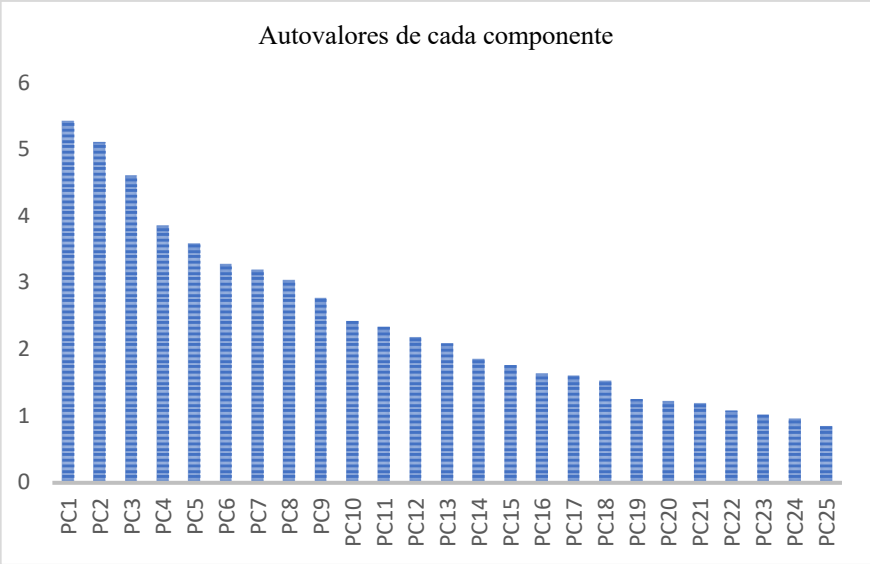


Fig. 2. Autovalores (valor de la varianza) de cada componente principal.

Una vez identificados los componentes principales, se procedió a analizar el valor de los autovectores de cada componente. En la Tabla 4 se muestra el análisis del componente 1 y 2. Los autovectores con mayor valor absoluto de cada componente definen las variables con mayor representatividad, en este caso para el componente 1 y 2 los valores absolutos más elevados son los presentes en V5, V13, V21, V24, V26 y V49. Estas variables corresponden con Número de integrantes en la familia, Número de hermanos estudiando, Número de personas que dependen del tutor, Está inscrito en alguna beca universitaria, Tiene beca del gobierno, Promedio del curso propedéutico. Por lo tanto, para estos dos componentes, estas variables tienen mayor relación con la deserción escolar. Este proceso se realizó con los 23 componentes.

Tabla 4. Resultado del análisis de componentes principales 1 y 2

VARIABLE	PC1	PC2	VARIABLE	PC1	PC2

V1	-0.033576348	0.055522599	V36	0.080798263	0.002043825
V2	-0.133852474	0.042016006	V37	-0.022568203	0.068959681
V3	-0.091427728	0.06556287	V38	-0.031861866	-0.066097053
V4	0.026633995	-0.073347705	V39	0.006102435	0.02640098
V5	0.082481206	**0.33538511**	V40	0.233220164	0.02410719
V6	-0.036931231	0.000664579	V41	-0.244243172	-0.025719939
V7	-0.071940299	-0.148715172	V42	-0.193850362	-0.004641089
V8	-0.096102987	-0.072140854	V43	0.119198643	-0.070916228
V9	-0.09462754	0.046900627	V44	-0.019405398	-0.049549285
V10	-0.133336044	-0.083945686	V45	-0.065468701	0.083874503
V11	0.099845227	0.263326581	V46	-0.072915512	-0.01440683
V12	0.111011757	-0.012564679	V47	-0.060579645	-0.073495573
V13	0.059607156	**0.29641244**	V48	-0.013238976	0.094178346
V14	-0.016414648	0.091023647	**V49**	**0.29319251**	0.003823
V15	0.060250692	-0.046449593	V50	-0.154969316	-0.080088441
V16	-0.050668325	-0.000262698	V51	0.058762534	0.131090934
V17	0.011105683	-0.042564428	V52	0.142235893	0.079688517
V18	-0.05755298	0.031464255	V53	0.139134135	0.054332191
V19	-0.110403016	0.239441698	V54	0.047524496	0.121767708
V20	-0.002164392	0.20864045	V55	0.175197936	-0.013979749
V21	0.109612104	**0.30223227**	V56	-0.034839732	0.002629099
V22	-0.012226608	-0.059142288	V57	0.132098386	-0.042062145
V23	-0.068450876	0.290395877	V58	-0.021987557	-0.085124831
V24	**0.25722645**	-0.155400498	V59	0.054582042	-0.136097473
V25	0.167489493	-0.134933656	V60	0.0161326	-0.017189776
V26	**0.28852597**	-0.090406871	V61	0.183938641	-0.042531737
V27	-0.170148218	-0.039156053	V62	0.205052073	-0.146110089
V28	-0.168653722	-0.029517141	V63	0.064942001	0.119091557
V29	-0.202225328	-0.017174541	V64	0.004421665	-0.017702249

V30	-0.08114845	-0.018348822	V65	0.057550073	0.179512639
V31	-0.157180117	0.228186371	V66	0.148272822	-0.068305615
V32	-0.036866276	-0.175979476	V67	-0.002796641	-0.155691247
V33	-0.139134654	-0.09998319	V68	-0.068599105	-0.099051166
V34	-0.144683348	-0.110255958	V69	-0.011103356	-0.015387664
V35	0.05226217	-0.048262718			

Inicialmente se determinaron 69 variables, una vez realizado el PCA se obtuvo un conjunto de 51 variables. Este nuevo conjunto de variables se adecuo en un archivo .ARFF para su posterior análisis en el software WEKA, además se agregó también una variable más para definir la clase a la que pertenece cada registro de la base de datos. Posteriormente, se aplicaron tres métodos de clasificación conocidos como árboles de decisión para determinar patrones que describieran el comportamiento de las variables relacionadas con la deserción escolar.

4 Resultados

En la aplicación de los tres métodos de árboles de decisión (J48, Randomtree y REPTree) se obtuvieron los siguientes resultados (Tabla 5): El número total de observaciones es de 51, de las cuales el algoritmo J48 muestra un porcentaje de predicción correcta del 70.588 % y un porcentaje del 29.411 % de predicción incorrecta. Este comportamiento se observa de igual forma para el algoritmo RandomTree. En el caso del algoritmo REPTree se observa un porcentaje de predicción correcta superior al 90 % en cuanto a la predicción incorrecta muestra un porcentaje de 9.803 %.

Tabla 5. Predicción correcta e incorrecta de los clasificadores.

Clasificador	**J48**		**RandomTree**		**REPTree**	
No. de observaciones	Correcto	Incorrecto	Correcto	Incorrecto	Correcto	Incorrecto
51	36	15	36	15	46	5
Porcentaje	70.588%	29.411%	70.588%	29.411%	**90.196%**	**9.803%**

El análisis de la matriz de confusión de cada algoritmo (Tabla 6, Tabla 7 y Tabla 8) confirma que el árbol REPTree clasifica de mejor forma las observaciones correspondientes a la clase Baja, para el caso de la clase Egreso si existe mayor confusión al momento de clasificar las observaciones.

Tabla 6. Matriz de confusión de j48.

Clasificado como	Baja	Egreso
Baja	11	7
Egreso	8	25

Tabla 7. Matriz de confusión de RandomTree.

Clasificado como	Baja	Egreso

Baja	8	10
Egreso	5	28

Tabla 8. Matriz de confusión de REPTree.

Clasificado como	Baja	Egreso
Baja	**13**	5
Egreso	0	**33**

Los resultados de la precisión detallada por clase de cada algoritmo (Tabla 9, Tabla 10 y Tabla 11) muestran que el árbol REPTree tiene una alta precisión de 1.000 para la clase Baja, 0.868 para la clase Egreso y un valor promedio de precisión de 0.915, estos resultados son mayores en comparación con los otros dos métodos que muestran un valor promedio de precisión de 0.710 para el algoritmo J48 y 0.694 para el algoritmo RandomTree. Usando REPTree se tiene una mejor representación de los patrones que determinan a los alumnos en riesgo de desertar en el programa de licenciatura en informática como lo muestra el árbol de decisión que se obtiene al aplicar este algoritmo (Fig. 3). En este caso el árbol de decisión tiene un tamaño de 8 y se observa que la variable que tiene mayor representación en el modelo es el promedio del curso propedéutico.

Tabla 9. Tabla que muestra la tasa de verdaderos positivos y falsos positivos de J48.

Clase	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
Baja	0.611	0.242	**0.579**	0.611	0.595	0.364	0.737	0.559
Egreso	0.758	0.389	**0.781**	0.758	0.769	0.364	0.737	0.811
Weighted Avg	0.706	0.337	**0.710**	0.706	0.708	0.364	0.737	0.722

Tabla 10. Tabla que muestra la tasa de verdaderos positivos y falsos positivos de RandomTree

Clase	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
Baja	0.444	0.152	**0.615**	0.444	0.516	0.321	0.651	0.486
Egreso	0.848	0.556	**0.737**	0.848	0.789	0.321	0.651	0.730
Weighted Avg	0.706	0.413	**0.694**	0.706	0.693	0.321	0.651	0.643

Tabla 11. Tabla que muestra la tasa de verdaderos positivos y falsos positivos de REPTree

Clase	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
Baja	0.722	0.000	**1.000**	0.722	0.839	0.792	0.900	0.887
Egreso	1.000	0.278	**0.868**	1.000	0.930	0.792	0.900	0.920
Weighted Avg	0.902	0.180	**0.915**	0.902	0.898	0.792	0.900	0.908

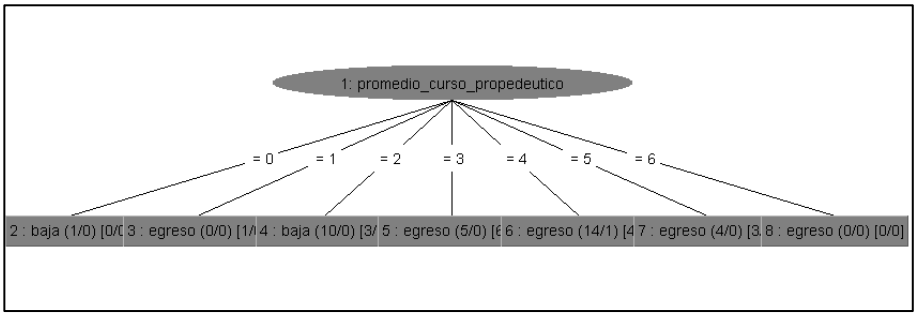


Fig. 3. Árbol de decisión del algoritmo REPTree.

4.1 Desarrollo de un sistema para predecir la deserción escolar

Los resultados que se generaron a partir del análisis de datos mediante la aplicación de técnicas de Minería de Datos, proporcionan las bases para el desarrollo de un sistema de predicción de la deserción escolar. Este sistema tiene dos finalidades, la primera, extender el número de observaciones por medio de un formulario (Fig. 4) para realizar el registro de nuevos datos con la finalidad de incrementar los niveles de precisión y también realizar la predicción de la deserción mediante la aplicación de los resultados que se obtuvieron de las técnicas de Minería de Datos (Fig. 5).

Fig. 4. Formulario para agregar nuevas observaciones.

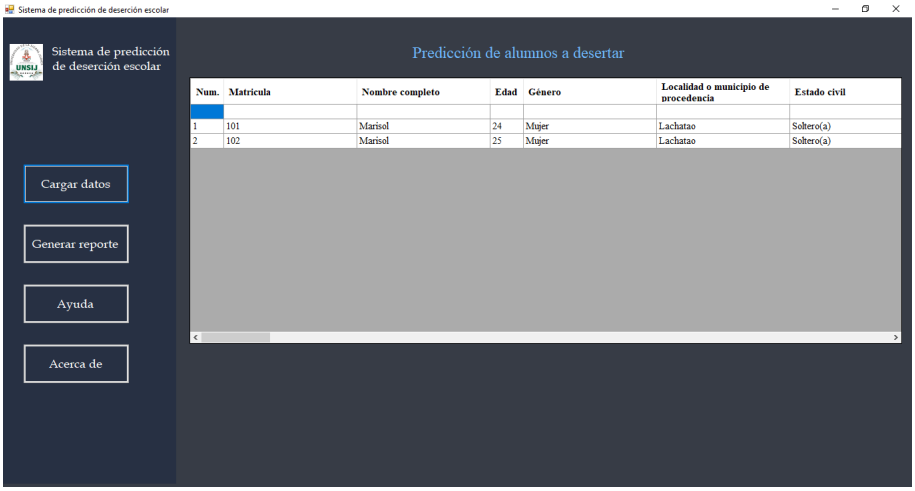


Fig. 5. Sistema para predecir la deserción escolar.

5 Conclusiones y trabajos futuros

De acuerdo a los resultados obtenidos, se determina que el modelo con mayor precisión para determinar los patrones de deserción o permanencia en el programa educativo se obtuvieron empleando el algoritmo REPTree. Se observa que el tamaño del árbol es pequeño, esto responde al número de observaciones que se utilizaron para el análisis, integrar un mayor número de observaciones mejoraría el modelo del árbol de decisión. Otra forma de mejorar el modelo correspondería a la suma de un mayor número de variables para el análisis de componentes principales, en este ejercicio se consideraron de manera inicial 69 variables de las cuales se eliminaron 18. También, es importante destacar que no se utilizaron métodos para el balance de las clases con menor representación como es el caso de los registros que corresponde a la clase Baja.

En la obtención de la información para el análisis, se consideró a los alumnos que cursan actualmente los semestres de tercero a décimo como observaciones de la clase Egreso, esto a razón de que los mayores niveles de deserción se registran en el primer y segundo semestre, por lo tanto los alumnos a partir del tercer semestre tienen mayor posibilidad de culminar su preparación universitaria.

Se espera como trabajo futuro el incremento del número de observaciones, además de la implementación de otros algoritmos de clasificación. Los resultados de esta investigación pretenden formar una base de variables que describan con mayor precisión la deserción de los alumnos del programa. Estos resultados se trasladaron a un sistema computacional para detectar con antelación a los estudiantes que presenten características inclinadas a la deserción y de esta manera proporcionarles atención en una etapa temprana para evitar la deserción.

El tener conocimiento oportuno de los alumnos en peligro de desertar contribuirá a identificar cuáles son los elementos a reforzar para evitar la deserción escolar y crear alternativas de solución al problema para integrarlas a los programas de permanencia educativa que se desarrollan en el programa de licenciatura. También, es importante mencionar que la información que se derive de esta investigación tendrá la finalidad de apoyar en las diferentes áreas de la universidad involucradas en el tránsito de los estudiantes por los programas de estudio.

Referencias

1. Fonseca, G.; García, F.: Permanencia y abandono de estudios en estudiantes universitarios: un análisis desde la teoría organizacional. *Revista de la Educación Superior*, Vol. 45, No. 179, pp. 25-39 (2016)
2. Espinosa-Castro, J.F.; Mariño-Castro, L.M.: Estrategias para la permanencia estudiantil universitaria, Ediciones Universidad Simón Bolívar, pp. 1-55 (2018)
3. Valdez, E.A.; Román Pérez, R.; Cubillas Rodríguez, M.J.; Moreno Celaya, I.: ¿Deserción o autoexclusión? Un análisis de las causas de abandono escolar en estudiantes de educación media superior en Sonora, México. *Revista Electrónica de Investigación Educativa*, Vol. 10, No. 1, pp.1-16 (2008)
4. Araque, F.; Roldán, C.; Salguero, A.: Factors influencing university drop out rates. *Computers & Education*, Vol. 53, No. 3, pp. 563-574 (2009)
5. García, M.E.; Gutiérrez, A.B.; Rodríguez-Muñiz, L.J.: Permanencia en la universidad: la importancia de un buen comienzo. *Aula Abierta*, Vol. 44, No. 1, pp. 1-6 (2015)
6. Antelm Lanzat, A.M.; Gil-López, A.J.; Cacheiro-González, M.L.: Análisis del fracaso escolar desde la perspectiva del alumnado y su relación con el estilo de aprendizaje. *Educ.Educ*, Vol. 18, No. 3, pp. 471-489 (2015)
7. Zonadameni, N.; Canale, S.; Pacifico, A.; Pagura, F.: El abandono en las etapas iniciales de los estudios superiores. *Ciencia, Docencia y Tecnología*, Vol. 27, No. 52, pp. 127-152 (2016)
8. Rodríguez Lagunas, J.; Leyva Piña, M.A.: La experiencia de la UAM. Entre el déficit de la oferta educativa superior y las dificultades de la retención escolar. *El cotidiano,* Vol. 22, No. 142, pp. 98-111 (2007)
9. Bautista Martínez, E.; Briseño Maas, M. L.: El derecho a la educación ante la desigualdad y la pobreza: el caso de Oaxaca, México. *El Cotidiano*, No. 183, pp. 83-90 (2014)
10. Casanova, J. R., Cervero, A.; Núñez, J.C.; Almeida, L.S.; Bernardo. A.: Factors that determine the persistence and dropout of university Students. *Psicothema*, Vol. 30, No. 4, pp.408-414 (2018)
11. Márquez Vera, C.; Romero Morales, C.; Ventura Soto, S.: Predicción del Fracaso Escolar mediante Técnicas de Minería de Datos. *IEEE-RITA*, Vol. 7, No. 3, pp.109-117 (2012)
12. Salvá-Mut, F.; Oliver-Trobat, M.F.; y Comas-Forgas, R.: Abandono escolar y desvinculación de la escuela: perspectiva del alumnado. *Magis. Revista Internacional de Investigación en Educación*, Vol. 6, No. 13, pp. 129-142 (2014)
13. Miranda, M.A.; Guzmán, J.: Análisis de la Deserción de Estudiantes Universitarios usando Técnicas de Minería de Datos. *Formación Universitaria*, Vol. 10, No.3, pp. 61-68 (2017)
14. Sushilkumar, K.: Analysis of WEKA Data Mining Algorithm REPTree, Simple Cart and RandomTree for Classification of Indian News. *International Journal of Innovative Science, Engineering & Technology,* Vol.2, No.2, pp. 438-446 (2015)

Tras la Sintetización de un Conjunto Reducido de Prototipos Óptimos para un Algoritmo de Clasificación: el k-NN Evolutivo.

Edgar Ibis Tacuapan-Moctezuma[1], Salvador E. Ayala-Raggi[1], Aldrin Barreto-Flores[1], José Francisco Portillo Robledo[1]

[1] Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Electrónica, Av. San Claudio y 18 sur, Col. Jardines de San Manuel, C.P. 72570, Puebla, Puebla, México.
edgar.tacuapanmoc@alumno.buap.mx, (salvador.raggi, aldrin.barreto, francisco.portillo)@correo.buap.mx

Resumen. Este artículo presenta un nuevo algoritmo de clasificación de patrones basado en kNN con aprendizaje evolutivo. A diferencia de los algoritmos de reducción de prototipos, en donde se suelen eliminar muestras apartadas de las fronteras de clasificación, y hacer sobrevivir muestras originales ubicadas en dichas fronteras, el algoritmo propuesto aquí, mejora gradualmente las posiciones de un conjunto creciente de prototipos sintéticos, cuyo tamaño aumenta a una tasa inferior a la cantidad de muestras de muestras utilizadas. Los resultados experimentales demuestran que el desempeño en clasificación, con el número de prototipos sintéticos creados, es mucho mejor que realizar una selección aleatoria de un número idéntico de muestras, y utilizarlas como prototipos en un kNN tradicional. Así mismo, también se demuestra que la exactitud de clasificación de nuestro algoritmo es aproximadamente igual, y en algunos casos es mejor, que los métodos actuales de reducción de prototipos.

Palabras Clave: KNN, Prototype Reduction, Gaussian Weighted KNN, Incremental Learning, Evolutive Learning.

1 Introducción

El algoritmo *kNN* es uno de los métodos de clasificación supervisados más utilizados gracias a que presenta buenos resultados, es fácil de entender e implementar, sin embargo, los requerimientos de almacenaje y el costo computacional aumentan a medida que el número de muestras incrementa, esto se debe a que la muestra a clasificar debe de ser comparada con cada una de las muestras almacenadas.

Se han presentado diversas soluciones para enfrentar este problema mediante la selección de prototipos representativos del conjunto de muestras disponibles (selección de prototipos) y mediante la creación de nuevos prototipos aplicando múltiples operaciones sobre el conjunto de datos original (generación de prototipos) [1].

Las mejoras que se han planteado se enfocan en aspectos como: la reducción de instancias, el mejoramiento de la exactitud en la clasificación y en la aceleración del proceso. En [2], presentan un algoritmo memético que es la combinación de un algoritmo evolucionario basado en población con una búsqueda local. Desarrollaron una búsqueda local especialmente diseñada para optimizar la selección de prototipos. Con este método abordan el problema de escalamiento que produce excesivos requerimientos de almacenaje, incrementa el tiempo en procesamiento y afecta la exactitud.

En [3], se propone una selección de prototipos vía relevancia, que se basa en seleccionar los prototipos más notables y los prototipos frontera, la relevancia de un prototipo se la determinan mediante el cálculo de la similitud promedio, una vez que hicieron el cálculo de los pesos de relevancia de cada prototipo del conjunto de entrenamiento, seleccionan los prototipos más relevantes por clase, durante esta selección

se incluyen algunos prototipos frontera, después para cada prototipo relevante, el prototipo más cercano que pertenece a una clase diferente se selecciona como prototipo frontera. Los mismos autores presentan en [4] un método de selección de prototipos para grandes conjuntos de datos basado en agrupamiento, el cual selecciona prototipos frontera y algunos prototipos internos, para realizar la selección dividen el conjunto de entrenamiento en pequeñas regiones para reducir el costo computacional, sobre estas regiones analizan cuales contienen prototipos pertenecientes a otras clases, si la región contiene solo elementos de la misma clase, solo se toma el prototipo más relevante. La división de las regiones se realiza mediante agrupamiento con *c-means*.

En [5], se proporciona una revisión de los métodos de selección de prototipos más destacados propuestos en la literatura, plantean una taxonomía basada en las principales características de los métodos revisados y analizan sus ventajas y desventaja. La taxonomía comienza en tres grupos: condensación edición e híbridos, de los cuales se desprenden otros, dentro de los algoritmos de condensación se encuentran los incrementales, decreméntales y por lote, ejemplos de estos son: CNN, RNN e IKNN respectivamente. En los métodos de edición encontramos ENN como parte de los decreméntales y a AIIKNN por lote. En los métodos híbridos mencionan a IB3 como algoritmo incremental, a VSM como decremental con filtro y a BSE como decremental con métodos *wrapper*, por lote está ICF y en los mixtos encontramos a SSMA. En ese trabajo, realizan comparaciones entre estos y otros métodos en términos de exactitud, capacidad de reducción y tiempos de ejecución mediante técnicas estadísticas no paramétricas sobre 58 conjuntos de datos que se subdividen en dos categorías: pequeños de no más de 2000 instancias y medianos de no más de 20000 muestras. En otra sección realizan un estudio del comportamiento de los algoritmos sobre siete conjuntos de datos de gran tamaño. Por último, comparan y presentan de manera gráfica los resultados obtenidos sobre un conjunto de datos artificial llamado *banana.*

Poda directamente ponderada es un método propuesto en [6] de condensación que utiliza un conjunto de pesos binarios para seleccionar cuales prototipos prevalecen, en el cual es posible controlar la condensación y la exactitud. El entrenamiento del modelo se basa en una optimización evolutiva discreta que busca el óptimo conjunto de pesos. Realiza la búsqueda mediante un algoritmo genético que contiene mecanismos para acelerar su convergencia.

En [7] proponen un algoritmo llamado árbol binario de vecinos más cercanos que genera y selecciona prototipos más relevantes del conjunto de datos mediante árboles binarios, con el cual buscan descartar la mayoría de los prototipos que se encuentran lejos de la frontera. Una vez que forman los árboles, si el árbol es homogéneo, entonces se consideran como prototipos internos, cuando el número de árboles sobrepasa cierto umbral, se genera el centroide del árbol como un nuevo prototipo interior, si no, se genera el centroide como un nuevo prototipo. Si el árbol no es homogéneo, preservan los prototipos frontera.

Algoritmo rápido para la selección de prototipos de margen de confianza presentado en [8], selecciona prototipos frontera confiables, considera que es un prototipo confiable, se considera a un prototipo confiable cuando si entre sus k vecinos existen k' vecinos de la clase opuesta. Si hay más de k' enemigos se toma como prototipo no confiable. Utilizan arboles aleatorios de regiones y bosques de árboles para optimización.

En este artículo, se propone un algoritmo basado en kNN de votación ponderada por distancia. En dicho algoritmo, al igual que en el esquema clásico kNN, se tiene un conjunto de muestras que directamente asumen el rol de prototipos de los cuales un subconjunto de k vecinos más cercanos a una muestra de prueba son utilizados para estimar su etiqueta mediante una votación ponderada que depende de las distancias de los vecinos a la muestra, y una función de base radial como la función gaussiana.

Dado que las muestras utilizadas como prototipos no han sido elegidas mediante algún proceso de selección óptima, no necesariamente se cuenta con el conjunto más adecuado capaz de generalizar mejor las muestras de prueba. Aunque muchos autores se han

esforzado en diseñar algoritmos para realizar selección de prototipos clave para mejorar la clasificación, en la materia de estos trabajos sólo se realiza una selección de los mejores prototipos originales, en vez de sintetizar nuevos prototipos que superen a los originales. Este artículo propone un método que intenta contestar la siguiente pregunta de investigación: ¿Es posible contar con un número reducido de prototipos óptimos que mejore la tasa de clasificación de un algoritmo de kNN?

2 Descripción del algoritmo propuesto

El algoritmo propuesto es básicamente un KNN evolutivo, o simplemente un *e-kNN,* que produce prototipos que cambian o evolucionan durante una etapa de entrenamiento con nuevas muestras que arriban de forma secuencial. Cada prototipo en el algoritmo además de su vector de características, y su respectiva etiqueta de clase, tiene asociado un registro o memoria a corto plazo en el cual se irán almacenado copias de las muestras para las cuales dicho prototipo resultó ser el más cercano. Dicha memoria es de tipo *FIFO* y tiene un tamaño limitado. Además, cada prototipo posee otra memoria más, también de tipo *FIFO*, la cual almacenará valores de una variable que nosotros hemos denominado *CLS* (Classification Strength) y que representa en un rango de 0 a 1 la intensidad de la clasificación de la muestra otorgada por el algoritmo *kNN*:

$$CLS = 1 - \frac{Vl}{Vw} \tag{3}$$

Donde *Vl* es la votación para la clase perdedora, y *Vw* la votación para la clase ganadora.

El algoritmo se inicializa con dos muestras válidas que son tomadas como prototipos iniciales. Visualizando el funcionamiento del entrenamiento en el espacio de características F-dimensional (con *F* siendo el número de características) donde yacen las muestras, cada vez que una nueva muestra se presenta, el prototipo que resulte más cercano se acercará en pasos pequeños a la muestra mientras aquella no sea bien clasificada en el sentido de kNN (por los prototipos existentes en ese momento), y ninguna de las muestras en las memorias resulte mal clasificada debido a dicho acercamiento. Este proceso se realizará con los *k* vecinos más cercanos a la muestra. En caso de que los acercamientos provoquen que la muestra sea bien clasificada, una copia de dicha muestra será agregada a la memoria del prototipo más cercano. Luego se verificará que las muestras almacenadas en las memorias de los prototipos cumplan con el requisito de pertenecer a la memoria de su prototipo más cercano. En caso de no cumplirse dicho requerimiento, se realizará una redistribución apropiada de las muestras almacenadas. Finalmente, se continuará con la siguiente muestra de entrenamiento.

Si los pasos de acercamiento de un prototipo provocaran una incorrecta clasificación de alguna de las muestras almacenadas en las memorias de los prototipos, antes de alcanzar la clasificación correcta de la muestra de entrenamiento, entonces el prototipo "acercado" dará un paso atrás regresando a su posición inmediata anterior, y será el siguiente vecino más cercano (de la misma clase) quien continuará dando pasos de acercamiento. Este proceso continuará hasta el último vecino más cercano *k,* mientras no se haya logrado la clasificación correcta de la muestra de entrenamiento. En caso de que el acercamiento del último vecino más cercano provoque una clasificación incorrecta de alguna de las muestras en las memorias, todos los *k* vecinos acercados (incluyendo el último), regresarán a sus posiciones iniciales y la muestra de entrenamiento se convertirá en un nuevo prototipo con una memoria en la que aparecerá ella misma en la primera posición.

Por el contrario, si al presentarse una nueva muestra de entrenamiento, ésta es bien clasificada (usando kNN), entonces la posición del prototipo más cercano se actualizará

como el promedio de las posiciones de las muestras dentro de su memoria. También, se almacenará el valor *CLS* de dicha clasificación en la memoria de valores *CLS* del prototipo. En este punto el algoritmo, cuando ocurra que la memoria *CLS* haya alcanzado un cierto tamaño, y el promedio de sus valores sea mayor a 0.9, entonces se eliminará el prototipo. Esto último ocasionará que sobrevivan solamente los prototipos en la zona fronteriza de las clases, y se eliminen los prototipos alejados de la frontera de clasificación.

La explicación de esto es que los prototipos en la zona de la frontera de clasificación clasifican con menor fuerza a las muestras de su vecindad. La figura 1 ilustra el caso cuando la muestra es incorrectamente clasificada y se alcanza su correcta clasificación con el acercamiento de sus prototipos vecinos. Por otra parte, la figura 2 ilustra el caso en que antes de lograr una correcta clasificación de la muestra, el movimiento de los prototipos ha ocasionado una incorrecta clasificación de una muestra dentro de alguna de las memorias de los prototipos.

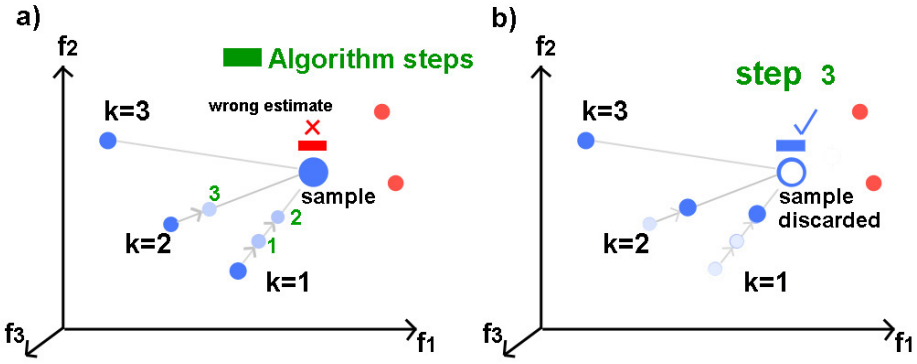


Fig. 8. a) Una muestra, que fue incorrectamente clasificada desde el principio, es alcanzada por sus prototipos vecinos para intentar una clasificación correcta de ésta. b) En el primer paso del segundo prototipo, la muestra ha resultado bien clasificada debido al acercamiento de los prototipos (siempre sin provocar algún error de clasificación en las muestras almacenadas en las memorias de éstos).

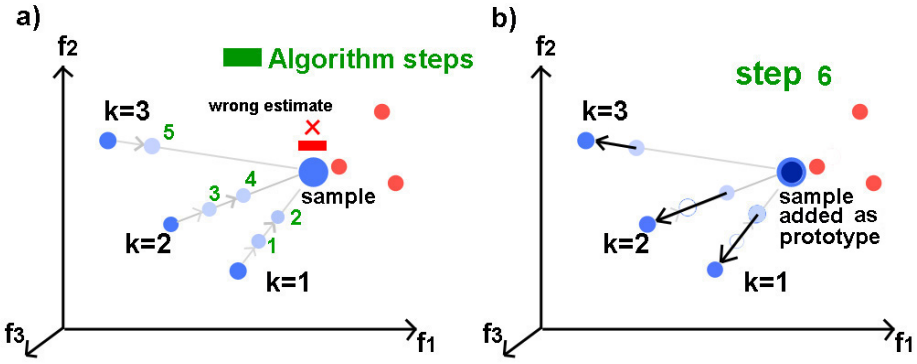


Fig. 9. a) Una nueva muestra, que fue incorrectamente clasificada desde el principio, es alcanzada por sus prototipos vecinos para intentar una clasificación correcta de ésta. b) En el primer paso del tercer prototipo la muestra sigue resultando equivocadamente clasificada a pesar del acercamiento de los prototipos. Sin embargo, el proceso de acercamiento es interrumpido cuando una muestra almacenada en alguna memoria resulta incorrectamente clasificada debido al movimiento del tercer prototipo. En este caso, los prototipos acercados son regresados a sus posiciones iniciales y la muestra de entrenamiento es añadida como un nuevo prototipo.

3. Resultados experimentales

Realizamos experimentos con tres conjuntos de datos de dos clases, obtenidos del repositorio UCI [9], con el conjunto de datos *banana* tomado del repositorio KEEL [10] y con un conjunto de datos artificial que es un grupo de puntos de dos características creados aleatoriamente con valores entre -1 y 1 en donde los puntos que se encuentran dentro de un círculo de radio 0.8 pertenecen a una clase y los que quedan fuera del círculo a otra clase. En la Tabla 1 se describen los conjuntos de datos utilizados y el número de muestras empleadas para entrenamiento y prueba. Para la validación utilizamos *c-fold* con c=10, por lo que el conjunto de datos se dividió en 10 bloques mutualmente exclusivos. Posteriormente se aplicó el algoritmo sobre el conjunto de entrenamiento compuesto de nueve de los bloques creados y el restante se utilizó para validación. Este proceso se aplicó *c* veces en donde cada bloque se utilizó como conjunto de prueba. Los resultados se presentan como el promedio de los resultados de todas las iteraciones en donde se le permite al algoritmo acercarse a la nueva muestra hasta 10 pasos de 30 posibles, es decir la distancia entre la muestra y el prototipo a acercarse se divide entre 30 y para estas pruebas solo tiene permitido dar hasta 10 pasos. Las memorias de los prototipos en donde se almacenan las copias de las muestras y la memoria CLS en la que se almacena la fuerza de clasificación, se fijaron a un tamaño de hasta 40 posiciones.

Tabla 4. Características de los conjuntos de datos y número de muestras utilizadas para entrenamiento (Ent.) y prueba.

Conjunto de datos	Muestras	Características	Clases	No. de muestras	
				Entr.	Prueba
Sonar	208	60	2	188	20
Spambase	4601	57	2	4142	459
Banana	5300	2	2	4771	529
Círculo	5500	2	2	4500	500

En la Tabla 2 comparamos los resultados obtenidos con los métodos DROP3, GCNN, POC-NN, CLU, PSC reportados en [4] para los conjuntos de datos *sonar* y *spambase*. Se presenta la exactitud y la tasa de reducción para cada algoritmo además se muestra el resultado obtenido con el kNN tradicional. Podemos observar que para el conjunto de datos *sonar*, el método propuesto obtiene mejor exactitud que DROP3 y CLU, en términos de la tasa de reducción e-kNN supera a todos excepto a CLU. En el banco de datos *spambase*, obtenemos mejores resultados que CLU y PSC en términos de exactitud. En cuanto a reducción se obtienen mejores resultados que DROP3, POC-NN y PSC.

Tabla 5. Comparación de resultados. Porcentaje de exactitud (Acc.) y porcentaje de reducción (Red.) con el conjunto de datos original y los prototipos seleccionados por DROP3, GCNN, POC-NN, CLU PSC y el método propuesto.

Conjunto	kNN norm		DROP 3		GCNN		POC-NN		CLU		PSC		e-kNN	
de datos	Acc	Red	Acc	Red	Acc	Red	Acc	Red	Acc	Red	Acc	Red	Acc	Red
Sonar	82.82	0	74.60	69.87	83.88	4.39	80.95	45.62	56.73	91.46	79.76	64.59	74.90	86.45
Spambase	78.46	0	78.44	84.29	73.54	98.67	73.37	65.97	69.95	99.72	71.95	75.26	73.26	95.14

En la Tabla 3 comparamos los resultados obtenidos con los mejores resultados presentados en [5] para el conjunto de datos *banana*. Obtenemos una mayor reducción que CNN, FCNN, IB3 y DROP3.

Tabla 6. Comparación de resultados. Porcentaje de exactitud (Acc.) y porcentaje de reducción (Red.) con el conjunto de datos original y los prototipos seleccionados por CNN, FCNN, IB3, DROP3, SSMA, e-kNN.

Conjunto	kNN norm		CNN		FCNN		IB3		DROP3		SSMA		e-kNN	
de datos	Acc	Red	Acc	Red	Acc	Red	Acc	Red	Acc	Red	Acc	Red	Acc	Red
Banana	87.43	0	88.64	77.29	86.55	80.10	84.42	87.11	86.96	91.51	89.64	98.79	83.63	92.22

Las Tablas 4 y 5 muestran el resultado de multiplicar la tasa de reducción por la exactitud, este es un estimador de que tan bueno es el método para seleccionar prototipos considerando una compensación entre la reducción y el éxito de la clasificación [5].

Tabla 7. Producto de exactitud (Acc.) y tasa de reducción (Red.) para los conjuntos de datos sonar y spambase.

Conjunto	Acc. * Red.					
de datos	DROP 3	GCNN	POC-NN	CLU	PSC	e-kNN
Sonar	0.5212	0.0368	0.3692	0.5188	0.5151	**0.6475**
Spambase	0.6611	0.7256	0.4972	0.6676	0.5414	0.6970

Tabla 8. Producto de exactitud (Acc.) y tasa de reducción (Red.) para el conjunto de datos banana.

Conjunto	Acc. * Red.					
de datos	CNN	FCNN	IB3	DROP3	SSMA	e-kNN
Banana	0.6696	0.6932	0.7353	0.7957	0.8855	0.7713

En la figura 3 observamos de manera gráfica el conjunto de muestras totales de *banana* y los prototipos producidos por el *kNN* evolutivo. Podemos observar que la mayoría de los prototipos finales pertenecen a prototipos frontera.

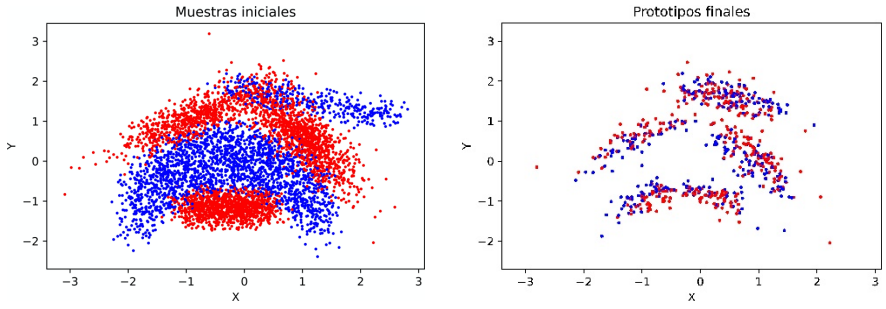


Fig. 3. Muestras iniciales y prototipos finales del conjunto de datos banana.

La figura 4 muestra los prototipos iniciales y los prototipos finales del conjunto de datos circulo. Al igual que en el conjunto de datos *banana*, los prototipos finales son prototipos cercanos a la frontera entre las dos clases. Debido a las características del conjunto de datos, se observa una gran reducción en el número de prototipos finales.

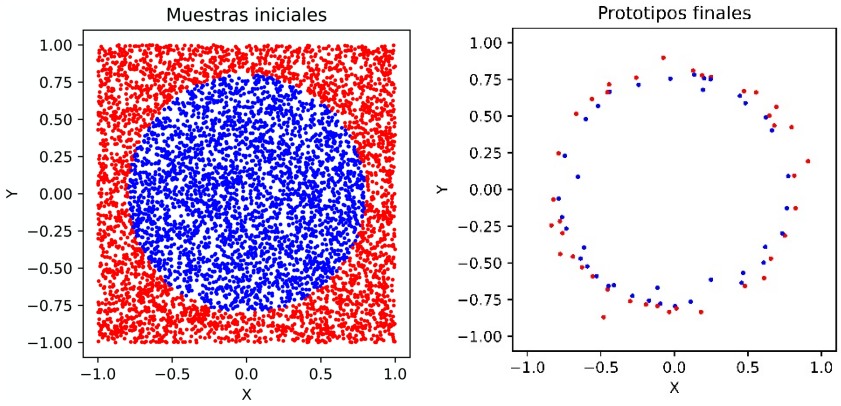
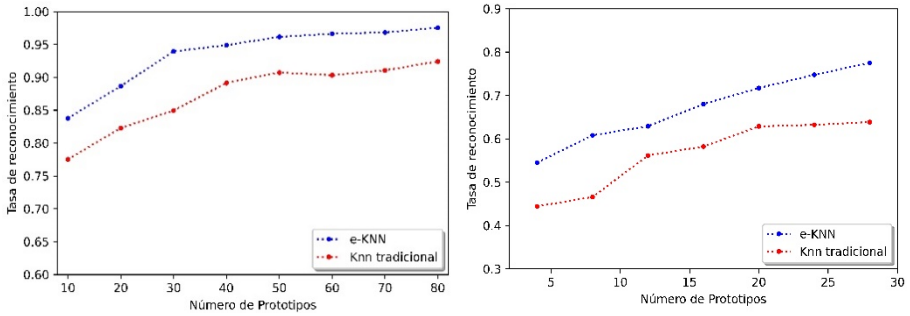


Fig. 4. Muestras iniciales y finales del conjunto de datos circulo.

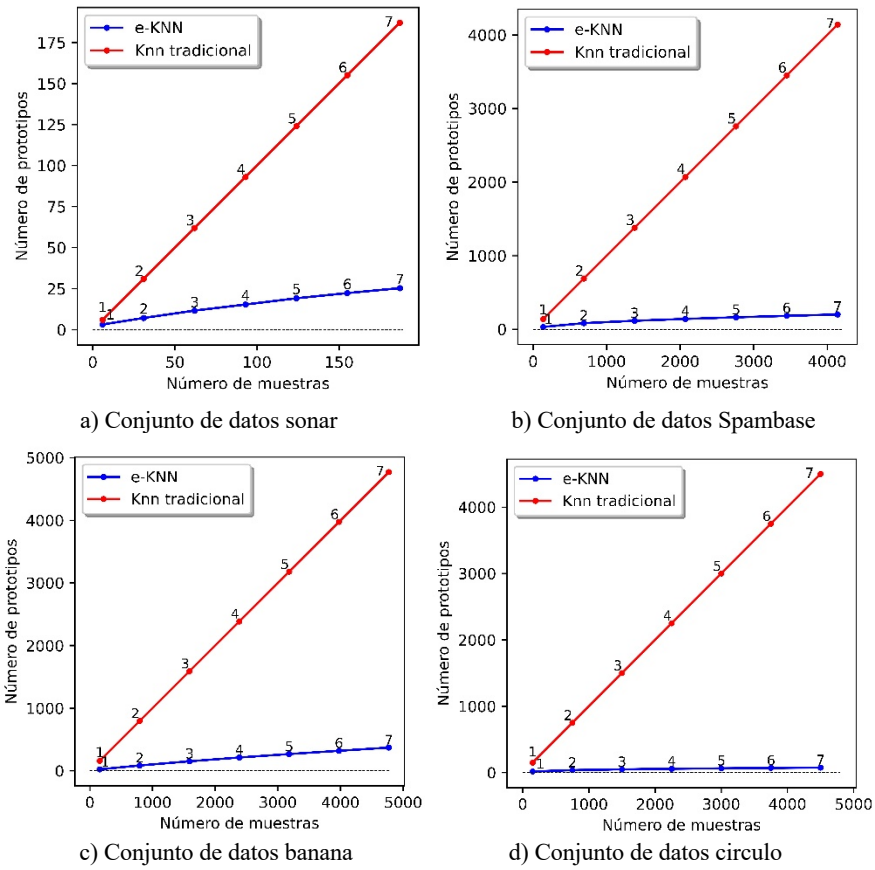
Realizamos un experimento adicional para observar el nivel exactitud alcanzado con diferentes números de prototipos producidos por e-kNN en comparación con el mismo número de prototipos seleccionados de manera aleatoria de los conjuntos datos *Círculo* y *Sonar* (figura 5). En ambos conjuntos de datos, los prototipos producidos por el algoritmo *e-kNN* presentan una mejor tasa de reconocimiento que una selección aleatoria de la misma cantidad de muestras utilizadas como prototipos en un clasificador *kNN* tradicional.



a) Conjunto de datos círculo b) Conjunto de datos sonar

Fig. 5. Exactitud del método propuesto en comparación con el Knn tradicional.

El *e-kNN* es un algoritmo de tipo incremental, en el sentido de que conforme se le van proporcionando muestras de entrenamiento, éste sintetiza gradualmente cierto número de prototipos. La figura 6 muestra el número de prototipos producido por *e-kNN* (color azul) a medida que se incrementa el número de muestras de entrenamiento proporcionadas al algoritmo, y se compara con el *kNN* tradicional que utilizaría cómo prototipos de clasificación todo el conjunto de muestras (color rojo) de entrenamiento que han sido proporcionadas al *e-kNN* hasta un momento dado, véanse los puntos numerados en cada una de las gráficas de la Figura 6, los cuales representan 7 de estos momentos. Para los cuatro conjuntos de datos se obtiene una reducción considerable en el número de prototipos útiles.



a) Conjunto de datos sonar

b) Conjunto de datos Spambase

c) Conjunto de datos banana

d) Conjunto de datos circulo

Fig. 6. Comparación entre el número de muestras utilizadas con el knn tradicional y los prototipos producidos por e-kNN para cada conjunto de datos.

4 Conclusiones y trabajo futuro

En este artículo se presentó un nuevo algoritmo de clasificación de patrones al que hemos llamado KNN evolutivo o simplemente e-KNN, basado en el algoritmo clásico KNN de votación ponderada en base a distancias. Los resultados experimentales demuestran que la exactitud de la clasificación (tasa de aciertos) rivaliza y en algunos casos es mucho mayor que en los métodos actuales de poda o reducción de prototipos. Nuestro método, a diferencia de los métodos actuales, busca mediante pequeños movimientos en el espacio de características, y en un proceso incremental, alcanzar posiciones óptimas intrínsicamente basadas en la distribución probabilística de las posiciones de las muestras de entrenamiento que van arribando una a una al algoritmo. Los resultados muestran que el algoritmo converge hasta un estado donde los prototipos evolucionados van juntándose justo en la frontera de clasificación, que es precisamente la zona más deseable para su ubicación final. Por otro lado, el algoritmo incorpora la definición de un parámetro que mide la intensidad o fuerza de clasificación CLS, que es utilizado para ir eliminando o podando paulatinamente aquellos prototipos que no resultan útiles en la tarea de clasificación de nuevas muestras. Consideramos que nuestra contribución es: un KNN evolutivo cuya cantidad de datos crece a una tasa mucho menor a la cantidad de muestras de entrenamiento presentadas al algoritmo. Dicho algoritmo, es también de aprendizaje incremental, en sentido que se le puede seguir entrenando al presentarle muestras nuevas, mientras que el conjunto de prototipos sintéticos ya evolucionados es utilizable al mismo tiempo para realizar una clasificación KNN clásica. Como trabajo futuro consideramos

realizar pruebas utilizando las mismas muestras de entrenamiento, pero con un ordenamiento aleatorio diferente a cómo fueron inicialmente presentadas al algoritmo.

Referencias

1 Triguero, I.; Derrac, J.; Garcia, S.; Herrera, F.: A taxonomy and experimental study on prototype generation for nearest neighbor classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol.42, pp. 86-100 (2012)

2 García, S.; Cano, J.R.; Herrera, F.: A memetic algorithm for evolutionary prototype selection: A scaling up approach. *Pattern Recognition,* vol. 41, pp.2693–2709 (2008)

3 Olvera-López, J.A.; Carrasco-Ochoa, J.A.; Martínez-Trinidad, J.F.: Prototype selection via prototype relevance. *Progress in Pattern Recognition, Image Analysis and Applications, Berlin, Heidelberg, Springer Berlin Heidelberg*, pp.153–160 (2008)

4 Olvera-López, J.A.; Carrasco-Ochoa, J.A.; Martínez-Trinidad, J.F.: A new fast prototype selection method based on clustering. *Pattern Analysis and Applications*, vol.13, pp. 131-141 (2010)

5 Garcia, S.; Derrac, J.; Cano, J.; Herrera, F.: Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol.34, pp.417–435 (2012)

6 Nikolaidis, K.; Mu, T.; Goulermas, J.: Prototype reduction based on direct weighted pruning. *Pattern Recognition Letters,* vol.36 22-28 (2013)

7 Li, J., Wang, Y.: A new fast reduction technique based on binary nearest neighbor tree. *Neurocomputing,* vol.149, pp.1647-165 (2015)

8 Jankowski, Norbertand Orlinski, M.: Fast algorithm for prototypes selection trust-margin prototypes. *In: Artificial Intelligence and Soft Computing, Cham, Springer International Publishing,* 583–594, (2019)

9 Dua, D., Graff, C.: UCI machine learning repository (2017)

10 Alcalá-Fdez, J.; Fernandez, A.; Luengo, J.; Derrac, J.; García, S., Sánchez, L.; Herrera, F.: Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework (2011)

Localización Automática de *Landmarks* con Robustez a la Traslación, Postura Angular y Escala, y su Aplicación en la Región del Tercer Hueso Metacarpiano en Imágenes Radiográficas de la Mano

Jamid Israel Saenz Giron, Salvador E. Ayala-Raggi, Mauricio Longinos Garrido, Aldrin Barreto-Flores, José Francisco Portillo-Robledo
Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Electrónica, Av. San Claudio y 18 Sur, Jardines de San Manuel, C.P. 72570 Puebla, Puebla,México
jamid.saenz@alumno.buap.mx
salvador.raggi@correo.buap.mx longinos.200918526@gmail.com
aldrin.barreto@correo.buap.mx francisco.portillo@correo.buap.mx

Resumen En este artículo se propone un método nuevo de localización automática de *landmarks* en imágenes radiográficas que es robusto a cambios en posición, postura angular, tamaño y contraste. Proponemos un método de estimación de la posición de la mano en una imagen radiográfica, basado en una mejora iterativa del cálculo del centro de masa. Se propone, además, un algoritmo basado en cambio de densidad para estimar el tamaño de la mano en la imagen. Así mismo, pre- sentamos un método fundado en la dirección de los gradientes de la imagen para realizar la estimación de la postura angular de la mano. Finalmente, presentamos un enfoque multiescala, que utiliza eigenespacios, para realizar la localización automática de las *landmarks*. Los resultados experimentales demuestran que el método propuesto consigue un error similar al de los métodos actuales, pero con la ventaja de la robustez a las variaciones en posición, postura angular, escala, y contraste de la región de interés.

Keywords: *Landmark* localisation, eigenfaces, radiographic image registration

1. Introducción

La localización exacta de landmarks y regiones de interés en imágenes radiográficas representa un problema desafiante. Esto debido a diversos factores como el contraste, el cual no permite distinguir correctamente los huesos, el tamaño absoluto de las estructuras óseas en la imagen, la posición, la postura angular, la variación particular de cada individuo en la forma de los huesos, y la intromisión de artefactos y otros objetos ajenos alrededor de las regiones de interés. La localización de *landmarks* es utilizada para buscar regiones de interés en distintas imágenes médicas tanto en 2D como son las radiografías y en 3D que corresponden las imágenes de resonancia magnética. En lo que respecta a la detección automática de *landmarks* en imágenes de manos, se han propuesto métodos como en [1] donde proponen la localización de puntos en radiografías al realizar de una búsqueda a diferentes escalas de la misma imagen, por medio de parches y al proyectarlos en un subespacio lineal para determinar si es semejante a las imágenes utilizadas en el entrenamiento realizado, y así ubicar las *landmarks*, este método lo utilizan dichos autores en imágenes 2D y 3D. En [2] se utiliza el método de machine learning de bosques aleatorios en conjunto con el método de modelo de forma precisa y ampliamente utilizado en imágenes de rostros, para ello crean distintos modelos de apariencia de la mano y por medio de los bosques aleatorios se va seleccionando la región donde se encuentran las *landmarks*. De manera similar, en [3] utilizan los bosques aleatorios para localizar

landmarks e identificar las zonas similares en las cuales los puntos podrían localizarse, y hacen uso de imágenes tanto en 2D como en 3D. Siguiendo el uso de bosques aleatorios, en [4], lo utilizan para modelar su regresor local basado en apariencia, combinar la información de apariencia de la imagen con la configuración geométrica de *landmarks* para ir localizando distintos puntos, también en imágenes 2D y 3D tanto en manos como en radiografías cefálicas laterales e imágenes de cuerpo completo. En [5] se utiliza otro método de machine learning, las redes neuronales convolucionales (Convolutional Neural Networks) en conjunto con el método de mapas de calor por regresión (Regressing Heatmaps) para imágenes de manos en 2D y 3D, en el cual proponen una nueva arquitectura con las redes neuronales convolucionales llamada Spatial Configuration-Net (SCN). En años recientes los autores en [6] utilizan su arquitectura SCN al localizar *landmarks* con un error bajo y además con un conjunto limitado de entrenamiento para dicha localización. Los métodos mencionados, realizan la localización de las *landmarks* y reportan errores promedio de unos pocos milímetros, sin embargo, la mayoría de ellos resuelven el problema de la localización cómo una optimización local [2], y no global, en donde la búsqueda debe realizarse muy cerca del lugar donde yacen las *landmarks*. Algunos otros, aunque con localización global [1], no reportan robustez a la escala y la rotación. Por esta razón el método aquí propuesto busca primero encontrar la mano completa en la imagen, normalizarla en tamaño y en postura angular para finalmente realizar la búsqueda de las *landmarks*. Por lo tanto proponemos un método nuevo multiresolución para la localización automática de *landmarks* que es robusto a la posición, postura angular, y escala de la mano. Hemos evaluado el nuestro método localizando automáticamente dos *landmarks* en el tercer hueso metacarpiano, uno ubicado en la parte superior y otro en la parte inferior del hueso mencionado.

2. Descripción general del Sistema

El método propuesto consiste en dos etapas principales: entrenamiento y evaluación. Para las dos etapas se determina la ubicación del baricentro de la mano, el cual nos permite estimar la región de la mano para eliminar objetos innecesarios que puedan aparecer en la región de la mano. El siguiente paso es encontrar automáticamente la región de interés donde se encuentre la mano para posteriormente aplicar una corrección de contraste en la región central de la mano que permita tener un contraste parecido en dicha región de la mano tanto en las imágenes del conjunto de entrenamiento como en la imagen de evaluación en la que se desea realizar el proceso automático de la localización de las *landmarks*. El paso siguiente será corregir de forma automática la postura de la mano (ángulo) utilizando para ello un algoritmo de reconocimiento automático del ángulo dominante de los gradientes de intensidad de la imagen dicho algoritmo es propuesto en este artículo. En la etapa del entrenamiento cada imagen se marca manualmente con dos *landmarks*, una en la parte superior y la otra en la parte inferior del tercer hueso metacarpiano. Dichas landmarks servirán como puntos de referencia centrales para extraer dos regiones de interés alrededor de cada una de estas dos landmarks. El algoritmo de detección que se propone en este artículo es multiresolución, con el objetivo de agilizar el proceso de búsqueda de cada una de las dos *landmarks*. Primero, se realiza la búsqueda de una de las *landmarks* en una resolución muy baja y posteriormente en resoluciones cada vez más altas. En la práctica utilizaremos una pirámide de tres resoluciones distintas. Por tanto, tendremos una región de interés alrededor cada la *landmark* por cada una de las tres resoluciones. Se utilizará el método de las eigenfaces [7] para representar cada una de las tres regiones de búsqueda para cada una de las dos *landmarks*. Con dicho método podemos comparar o proyectar en el subespacio lineal generado por el método de las eigenfaces alguna región candidata al momento de hacer la búsqueda y comprobar si el subespacio de las eigenfaces es capaz de generar una reconstrucción adecuada de la región. La búsqueda será comenzada en la menor resolución y terminara en la de mayor

resolución, y las regiones de búsqueda alrededor de la *landmark* en cuestión abarcaran una región más pequeña de la mano conforme cambiemos a una resolución mayor. Esto se observa en la figura 1.

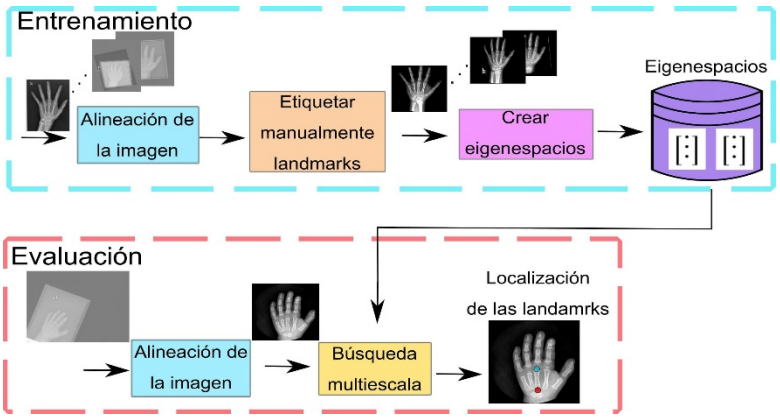


Fig. 1. Diagrama a bloques del método propuesto para la localización automática de *landmarks*.

3. Método propuesto

3.1. Descripción General del proceso de Alineación de Imagen

Debido a que el método de las *eigenfaces* funciona mejor cuando las imágenes de entrenamiento son parecidas entre sí, se requiere realizar un pre acondicionamiento de las imágenes de entrenamiento con el objetivo de que todas ellas tengan el mismo tamaño de imagen, y además, la región de la mano aparezca con la misma postura (tanto en posición, cómo en ángulo) y con una escala uniforme en todas. Así mismo, también se busca uniformizar el contraste de la región de interés dentro de la mano. En pocas palabras, se realizara´ una alineación de las imágenes de entrenamiento en cuatro aspectos diferentes: traslación, rotación, escala y contraste. De esta manera,

llamaremos $I(m, n)$ a la imagen original, y $I^s(m, n)$ a una imagen intermedia alineada en posición, escala y contraste. La imagen de salida de dicho algoritmo es una imagen cuadrada de 256x256 pixeles donde la mano se encuentra centrada y corregida en contraste. El último paso de la alineación de la imagen es realizar una corrección de la postura angular de la manoy tener la imagen $I^s(m, n)$.

3.2. Segmentación de la mano

En esta sección se busca obtener imágenes segmentadas que nos permitan disminuir las diferencias entre cada una de ellas. Primero se modifica el tamaño de la imagen de entrada $I(m, n)$, al determinar su dimensión más pequeña y cambiarla a 256 pixeles. La otra dimensión se modifica a un valor proporcional p, para obtener una nueva imagen $I^p(m,n)$. Lo siguiente es localizar el baricentro de la mano b_n, (sección 3.3). Después se eliminan los objetos alrededor de la mano (sección 3.4) para obtener la imagen $I^c(m, n)$ y el radio de la mano r_c. Se continua al ubicar el baricentro en la imagen de entrada $I(m, n)$ al multiplicar b_n por el valor proporcional p y multiplicar dicho valor por el radio r_c para aplicarle una máscara binaria $M(m,n)$ circular a la imagen. Se procede a realizar un corte sobre $I(m,n)$ y obtener la región $C(m, n)$ siendo el centro geométrico de la imagen el baricentro (m_f,n_f). Se cambian las dimensiones de la región a 256x256 pixeles para obtener la imagen $F(m, n)$. Finalmente la imagen se corrige en contraste para obtener $I^g(m, n)$. Lo anterior se muestra en la figura 2.

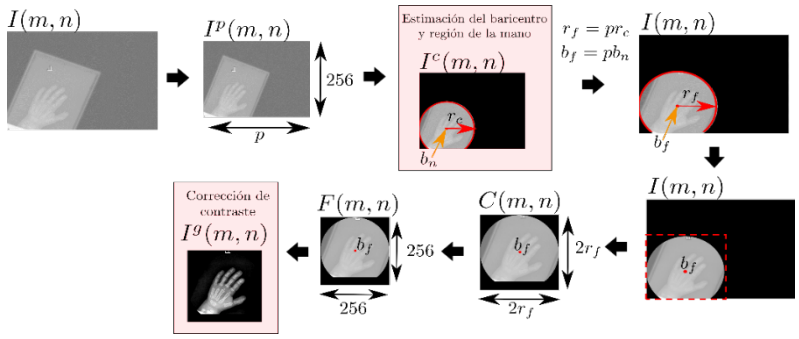


Fig. 2. A partir de *I*(*m, n*) se modifican las dimensiones para localizar el baricentro *bn,* la región de la mano en conjunto con *rc* y trasladarlos en *I(m,n)*.Se continua aplicando un corte dentro de la imagen *I*(*m, n*) creando *C*(*m, n*), además se cambian sus dimensiones para tener *F*(*m, n*). Por último se realiza una corrección de contraste y obtener la imagen I^g(*m, n*).

3.3. Estimación del baricentro de la región de la mano

Se propuso localizar el baricentro, ya que a partir de éste se podrá dibujar una circunferencia que nos permitirá obtener solo la región de la mano. Para ello se seleccionó un radio inicial al considerar las manos más grandes, con el fin de ir reduciéndolo. Lo siguiente es obtener los bordes y localizar el baricentro por primera vez para tener una aproximación y trasladarlo con la misma posición en la imagen inicial *I(m,n)*. Se continúa con corregir el contraste de la imagen. A continuación se repite la detección de bordes con la imagen resultando una nueva imagen $E_2(m, n)$. Además de volver a localizar por segunda vez el baricentro en $E_2(m, n)$. A partir del punto $b_a = (m_{b2}, n_{b2})$ y siendo éste el centro, se dibuja una circunferencia de radio *r* para comenzar el ajuste en la posición del baricentro. Así se inicia por localizar dentro de la circunferencia y guardar la posición del baricentro en la variable b_a, e ir reduciendo el radio en 10 pixeles, repitiendo la localización del nuevo baricentro y guardarlo en b_n=(m_{b3},n_{b3}), como se muestra en la figura 3. Además de tener dos condiciones de paro al ubicar el punto. Para la primera se mide la distancia euclidiana entre b_a y b_n y si es menor o igual a 1 píxel, se detiene la localización. Para la segunda condición se tomó en consideración que a partir de que el radio llegara al valor de 30 pixeles se detenga la localización. Al cumplirse una de las dos condiciones se mantiene la ubicación final de b_n.

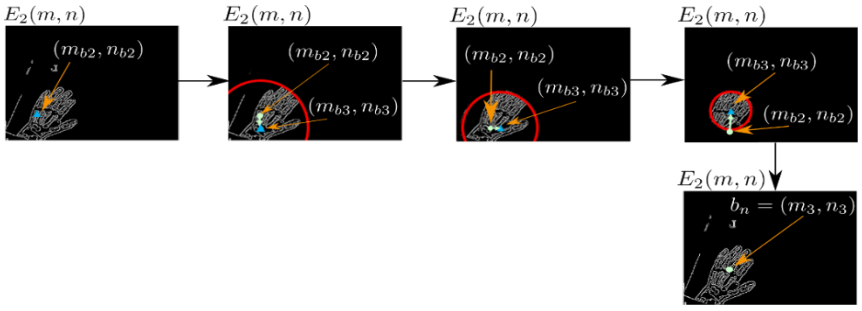


Fig. 3. Localización del baricentro en la región de la mano en la imagen *E*2(*m,n*). El punto (m_{b2},n_{b2}) representa el punto anteriormente ubicado y (m_{b3},n_{b3}) el nuevo baricentro localizado, además de la reducción de la circunferencia para la búsqueda del baricentro final *bn*.

3.4. Estimación de la región de la mano

Se buscó eliminar objetos alrededor de la mano ya que no aportan información en la búsqueda de *landmarks*. Se comienza al asignar a la constante *c* un valor de 1.32. Continuamos creando dos conjuntos, uno de radios *R* con 18 elementos, por lo que para

$r_i \in R$ se calcula $r_i=30c^n$ y el segundo para áreas de cada elemento del conjunto de radios. Se prosigue con un tercer conjunto M que contiene las 18 imágenes rectangulares conformadas por máscaras binarias donde el interior de la circunferencia se conforma por valores de 1 y en el exterior valores de 0. Continuando se multiplica la imagen de bordes $E(m, n)$ por cada elemento del conjunto M para crear el conjunto M_E . Para formar el conjunto H que contiene coronas circulares formadas por la resta absoluta de la imagen binaria rectangular de bordes M_{Ei} por su elemento posterior M_{Ei+1} (figura 4) a excepción del primero elemento H_1 conformado por la primera imagen de bordes binaria con la máscara circular M_{E1}. De manera similar se crean coronas circulares pero en el conjunto de áreas A, donde el primer elemento del nuevo conjunto corresponde a $H_{A1} = A_1$ y realizando $|A_i - A_{i+1}|$. Por lo tanto para cada $H_i \in H$ y $H_{Ai} \in H_A$ se calcula $D_i = H_i/H_{Ai}$ para $D_i \in D$. Por consiguiente se seleccionó un valor umbral que permita conocer en que circunferencia se encuentra la mano. Para facilitarlo se normalizaron todos los valores del conjunto D al ordenarlos de mayor a menor y se encontró que al considerar los dos primeros máximos del conjunto siendo D_1, D_2, se tiene una mejor consideración del umbral, así se obtiene el valor promedio $\overline{m}$ de los valores antes mencionados y se normaliza todo el conjunto D al dividir cada elemento D_i entre $\overline{m}$, creando el conjunto E. Para poder concluir con un valor umbral de 0.2. Así para cada nueva imagen, se obtiene su conjunto E y conocer que circunferencia debe aplicarse. Esto se realiza con las variables *contador* y r_c. Los pasos posteriores se muestran en el algoritmo 1.

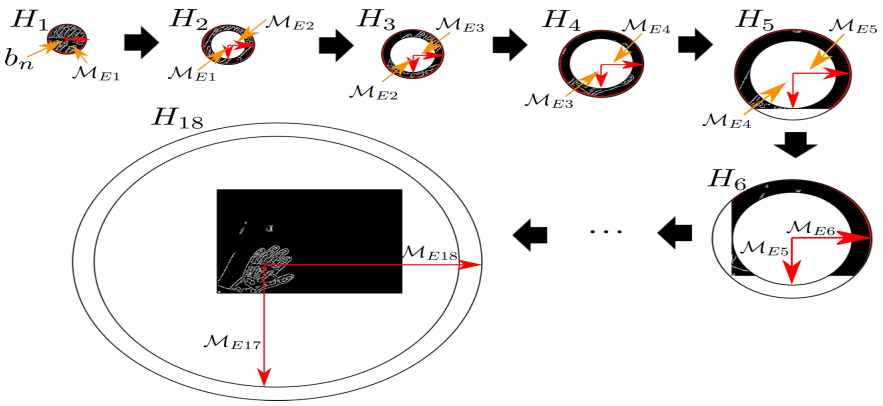


Fig. 4. Creación de las coronas circulares con la imagen de bordes $E(m, n)$, la cual se inicia con el baricentro b_n siendo el centro y comenzar con el radio M_{E1} para formar el primer elemento H_1, se continua con la corona formada por H_2 hasta formar la corona H_{18}.

Algoritmo 1: Estimación de la región de la mano basada en cambios de densidad usando una región circular creciente.

Entradas: Imagen $I(m, n)$, Imagen de bordes $E(m, n)$ y baricentro (m_b, n_b).
Salidas: Imagen de la mano con su circunferencia contenida IE (m, n), radio de la circunferencia rc.

1 inicio
2 Inicializar $c = 1.32$.
3 Crear el conjunto de radios R = $\{r1, r2, r3, ..., r18 | ri = 30c^n\}$.
4 Crear el conjunto de áreas A = $\{A1, A2, A3, ..., A18 | Ai = \pi r^2\}$.
5 Crear el conjunto de imágenes binarias
$M = \{M1(m.n), M2(m, n), ..., M18(m, n)\}$.
6 Realizar $ME = \{E(m, n)Mi(m, n) | Mi(m, n) \in M\}$.
7 Crear el conjunto de coronas circulares $H = \{H1, H2, ..., H18\}$.
8 Crea el conjunto de áreas de las coronas circulares $HA = \{HA1, HA2, ..., HA18\}$.
9 Calcular la densidad de área de cada corona circular $D = \{D1, D2, ..., D18\}$

mediante: *Di = Hi/HAi*
10 Ordenar de mayor a menor las densidades D.
11 Obtener el primer *D*1 y segundo *D*2 máximos de D.
12 Calcular el valor promedio $\overline{m}$ al realiza: $\overline{m}$=*(D1+D2)/2* .
13 Normalizar todas las densidades de *D* con *E* = {*E*1, *E*2, ..., *E*18|*Ei* = D_i/m}.
14 Sea *contador* = 0 y *rc* = 0 .
15 **para cada** *elemento de E* **hacer**
6 **si** *para cada elemento Ei* =< 0.2 **entonces**
17 *rc* ← *contador*.
18 **break**
19 **sino**
20 *contador* ← *contador* + 1
21 **fin**
22 **fin**
23 Seleccionar el número de circunferencia *rc* final, en *R*.
24 Realizar *IE* (*m, n*) = *M* (*m, n*)*I*(*m, n*), donde *M* (*m, n*) tiene radio *rc*.
25 fin

3.5. Corrección angular de la imagen de la mano

Se propuso un método que nos permita corregir el ángulo de la imagen de la mano. Se inicia con obtener los gradientes verticales y horizontales de la imagen $I^g(m, n)$ también se aplica una máscara circular binaria a la imagen y obtener $I^m(m, n)$. Lo siguiente es calcular su histograma polar con un total de 360 vectores, después se crean los vectores $\mathbf{h}_i$ para $i = 1, 2, ..., 360$, donde cada uno corresponde a la suma ponderada de una ventana que comprende a los vectores de i a $i + 90^\circ$ del histograma polar. A partir de los vectores $\mathbf{h}_i$ se busca el vector con magnitud máxima $\mathbf{h}_{max1}$. Se prosigue por dibujar una línea perpendicular y localizar un segundo vector máximo $\mathbf{h}_{max2}$ en la región opuesta al primer vector máximo. Se continúa por hallar el vector $\mathbf{h}_M$ de los vectores $\mathbf{h}_{max1}$ y $\mathbf{h}_{max2}$. En algunas ocasiones al hallar el vector máximo y corregir su ángulo no se tenía el resultado esperado. Por lo tanto se decidió crear un método que permita medir la densidad en la región de la mano y poder corregir el ángulo de manera correcta. Así se inicia por aplicar una máscara circular binaria a la imagen $I^g(m, n)$, con diámetro d_c de longitud un cuarto de dicha imagen y centrado en el centro geométrico para obtener $I^d(m, n)$ y se divide en dos mitades al dibujar una línea perpendicular al vector $\mathbf{h}_M$ el cual se ubica en el centro geométrico de $I^d(m, n)$, donde la primera mitad es $C_1(m, n)$ y la segunda $C_2(m, n)$, así mismo se selecciona un parche $P(m, n)$ de tamaño un cuarto del tamaño de la imagen $I^d(m, n)$. Para todos los pixeles (m, n) de $C_1(m, n)$, se centra el parche $P(m, n)$ en (m, n) y se calcula el valor p solo si el píxel cumple que $(m, n) > \bar{g}$ donde $\bar{g}$ e ir guardando cada valor p en el conjunto P_1. Para $C_2(m, n)$ también se realiza el mismo proceso anteriormente mencionado para formar el segundo vector $_2$. Por otra parte se obtiene el valor promedio $\bar{r}_1$ del vector $\mathbf{r}_1$ y el valor promedio $\bar{r}_2$ del vector $\mathbf{r}_2$. Finalmente si $\bar{r}_1 > \bar{r}_2$ y además el vector $\mathbf{h}_M$ se encuentra en la mitad $C_1(m, n)$ se rota la imagen 180° y se tiene la imagen $I1^s(m, n)$, en otro caso si $\mathbf{h}_M$ se encuentra en la mitad $C_2(m, n)$ se rota la imagen 180° y se tiene la imagen $I1^s(m, n)$. Se muestra en el algoritmo 2 con lo anteriormente mencionado.

Algoritmo 2: Corrección angular de la imagen de la mano

Entrada: Imagen $I^g(m, n)$.
Salida: Imagen corregida en ángulo $I^s(m, n)$.
1 inicio
2 Aplicar una máscara circular binaria $I^m(m, n)$ a $I^g(m, n)$.
3 Obtener el histograma polar de los vectores de gradiente de cada uno de los pixeles de $I^m(m, n)$.

4 Crear y obtener su histograma polar del conjunto H donde *hi* corresponde a la suma ponderada de los vectores del histograma de paso 2, al tomar una ventana de vectores i a $i + 90^{\circ}$.
5 Obtener el vector con mayor magnitud **h***max*1 del histograma del paso 4.
6 A partir de **h***max*1 dibujar una línea perpendicular y localizar **h***max*2 en la región opuesta al primero vector máximo.
7 Hallar el vector **h***M* con máxima magnitud de los vectores **h***max*1 y **h***max*2.
8 Aplicar una máscara circular binaria de radio *dc* y dimensiones de un cuarto de sus dimensiones a $I^m(m, n)$ para obtener $I^d(m, n)$.
9 Ubicar en el centro geométrico de la imagen $I^d(m, n)$ a **h***M* para dibujar una línea perpendicular a dicho vector y dividir en dos mitades $C1(m, n)$ y $C2(m, n)$ la máscara circular del paso 8.
10 Obtener el valor promedio g^- de todos los niveles de gris de $I^d(m, n)$.
11 Sea $j = 1$ que corresponde a la primera mitad de la circunferencia del paso 9.
12 **repetir**
13 **para cada** (m, n) *de* Cj (m, n) **hacer**
14 **si** $(m, n) >= g^-$ **entonces**
15 Tomar $P(m,n)$ de tamaño 1/4 de las dimensiones de $I^d(m, n)$ y centrada en (m, n).
16 Hacer $s = \sum_{m,n} P(m, n)$.
17 Almacenar s en **S***j* .
18 **fin**
19 **fin**
20 $j = j + 1$
21 **hasta que** $j = 2$;
22 Obtener el valor promedio p^-1 de **S**1 y p^-2 de **S**2.
23 **si** $p^-1 > p^-2$ **entonces**
24 **si** ***h****M está ubicado dentro de* $C1(m, n)$ **entonces**
25 Rotar 180° la imagen $I^g(m, n)$ y obtener $I^s(m, n)$.
26 **fin**
27 **sino´**
28 **si** ***h****M está ubicado dentro de* $C2(m, n)$ **entonces**
29 Rotar 180° la imagen $I^g(m, n)$ y obtener $I^s(m, n)$
30 **fin**
31 **fin**
32 fin

3.6. Etapa de entrenamiento

Por cada *landmark*, se realiza la construcción de 3 subespacios lineales usando el método de las *eigenfaces*. Se deben etiquetar manualmente un conjunto de imágenes con las 2 *landmarks* propuestas. Este etiquetado se realiza en las imágenes de mayor resolución. Cada subespacio o *eigenespacio* está formado por *eigenimagenes* de 32*x*32 pixeles para cada uno de los tres niveles de búsqueda. Las imágenes en cada nivel de búsqueda son de 64*x*64, 128*x*128 y 256*x*256 pixeles. Por las imágenes de entrenamiento, sabemos que la posición probable de la *landmark* está dentro de una región rectangular cuyos límites podemos determinar. El límite izquierdo corresponde a la coordenada horizontal de la *landmark* posicionada más hacia la izquierda. El límite derecho corresponde a la coordenada horizontal de la *landmark* posicionada más hacia la derecha. El límite superior corresponde a la coordenada vertical de la *landmark* posicionada más hacia arriba. El límite inferior corresponde a la coordenada vertical de la *landmark* posicionada más hacia abajo en las imágenes de ejemplo. Esta delimitación nos permite crear una región de búsqueda rectangular $R_3(m, n)$ dentro de la imagen $I3^s(m, n)$ tomando los siguientes pixeles como vértices para construir la región: superior izquierdo (29, 12), superior derecho (38, 12), inferior izquierdo (29, 36), inferior derecho (38, 36), teniendo dimensiones de 24*x*9 pixeles, para la landmark superior. Para la inferior, selecciona una

región de búsqueda rectangular $D_3(m, n)$ dentro de la imagen $I3^s(m, n)$ tomando los siguientes pixeles como vértices para construir la región: superior izquierdo (27, 29), superior derecho (38, 29), inferior izquierdo (27, 49), inferior derecho (38, 49), teniendo dimensiones de 20*x*11 pixeles.

3.7. Búsqueda de la *landmark* en diferentes niveles de resolución

En esta etapa se tiene como objetivo la búsqueda de la *landmark* en diferentes escalas, ya que se cuenta con la imagen de la región de la mano en 3 diferentes tamaños. Primero se inicia con la imagen $I^s(m,n)$ de 64*x*64 pixeles, también se tiene definida la región de búsqueda $R_3(m, n)$ dentro de dicha imagen. Se realiza la primera localización de la *landmark* $\boldsymbol{l}_3$ (sección 3.7.1). Lo siguiente es redimensionar $\boldsymbol{l}_3$ en la imagen $I_2^s(m, n)$ de 128*x*128 pixeles, al realizar esto la landmark equivale a una región cuadrada de 2*x*2 pixeles, pero agregando 2 pixeles a cada lado de la región siendo de dimensiones 6*x*6 pixeles que se utilizará como región de búsqueda $R_2(m, n)$ para localizar $\boldsymbol{l}_2$. Repetimos la búsqueda teniendo una mejor posición de la *landmark*. A continuación $\boldsymbol{l}_2$ se redimensiona en la imagen $I_1^s(m, n)$, por lo tanto se obtiene una región de búsqueda también de 2*x*2 pixeles, agregando 2 pixeles a cada lado para formar $R_1(m, n)$. Finalmente se repite la localización de la *landmark* en la imagen $I_1^s(m, n)$ y la región de búsqueda $R_1(m, n)$, localizando la *landmark* $\boldsymbol{l}_1$ dentro de la imagen con mayores dimensiones $I_1^s(m, n)$ y ubicada en el hueso medio metacarpiano correspondiente. Se muestra de manera gráfica la búsqueda de la *landmark* en diferentes escalas en la figura 5.

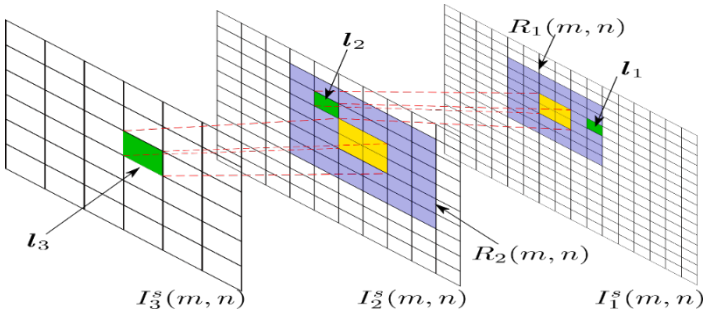


Fig. 5. Al localizar la posición de la *landmark* $\boldsymbol{l}3$ en la imagen $I_3^s(m, n)$ se redimensiona el punto en la imagen I_2^s siendo una región de búsqueda $R2(m, n)$, y se repite la búsqueda, para localizar el punto $\boldsymbol{l}2$, el cual se redimensiona en la imagen $I_1^s(m, n)$ obteniendo una nueva región de búsqueda $R1(m, n)$, repitiendo la búsqueda del punto en $R1(m, n)$, para localizar la *landmark* final $\boldsymbol{l}1$.

3.7.1 Proyección de parches en el subespacio de las eigenfaces

La búsqueda de la *landmark* solo se realiza en $R(m, n)$. Lo primero es a partir del primer píxel de $R(m, n)$ y siendo éste el centro se toma un parche $P(m,n)$ de tamaño de 32*x*32 pixeles, lo siguiente es aplicar el algoritmo *PCA* al convertir $P(m, n)$ en un vector columna Φ y restarle el vector promedio $\bar{\Psi}$, después se proyecta en el subespacio lineal correspondiente **Q** de acuerdo al nivel de resolución de $I^s(m, n)$ y sumarle el vector $\bar{\Psi}$ para obtener el vector columna $\hat{\Phi}$. Posteriormente se obtiene un error de reconstrucción $e = \|\Phi - \hat{\Phi}\|$. Esto se realiza con todos los pixeles de $R(m, n)$ para cada uno se obtiene un error de reconstrucción y se guarda en un mapa de errores E, donde la posición (m, n) de dicha mapa corresponde al píxel utilizado en $R(m, n)$. Una vez finalizada la búsqueda se determina la posición mínima al calcualr $\boldsymbol{l} = \text{argmin}_{(lm,ln)}\, E(lm, ln)$. por lo cual $\boldsymbol{l}$ es la posición donde se ubica la *landmark* dentro de la imagen $I^s(m, n)$. Estos pasos se muestran en la figura 6.

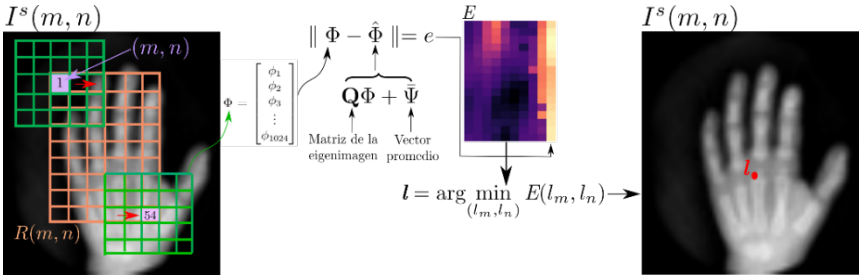


Fig. 6. A partir de la región *R*(*m, n*), se inicia en el píxel 1 y siendo éste el centro se toma un parche que se convierte en un vector columna *Φ* el cual se proyecta en **Q**, para obtener $\hat{\Phi}$ y calcular la norma *e* de los dos vectores y guardarlos en *E*. Una vez finalizada toda la reconstrucción de la región *R*(*m, n*) se busca la posición mínima en *E* que corresponderá a la *landmark* ***l*** en el hueso medio metacarpiano.

3.8. Métricas de evaluación

Para determinar el error se tuvo que buscar información que nos pudiera dar la longitud en milímetros del tercer hueso metacarpiano en diferentes rangos de edad y por sexo, el cual se obtuvo de [8]. Un problema es que el rango de edad en el que se trabajó para el algoritmo fue de 1-19 años, pero la información proporcionada es de 3-19 años, así se tuvo que realizar un ajuste por mínimos cuadrados. La función obtenida para determinar la longitud del tercer hueso metacarpiano es de la forma:

$$l = a - be^{-ce_o} \tag{1}$$

donde e_o corresponde a la edad ósea. Así se tiene una función de longitud del hueso para mujeres $L3M_{mm}^{f}$ y de hombres $L3M_{mm}^{m}$ Quedando de la siguiente manera:

$$L3M_{mm}^{m} = 86.85 - 66.94e^{-0.06e_o} \tag{2}$$
$$L3M_{mm}^{f} = 66.77 - 44.15e^{-0.11e_o} \tag{3}$$

Al conocer la longitud del hueso en milímetros con (2) y (3) se buscó el número de milímetros en un píxel con su respectiva longitud en pixeles obtenida por las *landmarks* marcadas manualmente en su correspondiente sexo donde para hombres es $L3M_{px}^{m}$ y para mujeres $L3M_{px}^{f}$, quedando de la siguiente manera:

$$mm\_per\_px_m = L3M_{mm}^{m} / L3M_{px}^{m} \tag{4}$$
$$\text{mm\_per\_px}_\text{m} = L3M_{mm}^{f} / L3M_{px}^{f} \tag{5}$$

Por lo tanto para cada imagen evaluada se calculó su respectivo valor *mm per px* dependiendo del sexo y se multiplica por su error en pixeles e_{px} por lo que de esta manera obtenemos el error en milímetros e_{mm} como:

$$e_{mm} = (mm\_per\_px)e_{px} \tag{6}$$

Finalmente se obtiene el error promedio e^p en milímetros de acuerdo a $e^p=(1/n)\sum_{i=1}^{n} e_{imm}$, donde $n = 250$.

4. Resultados experimentales

En la tabla 1 se muestra el error estimado al marcar las *landmarks* automáticamente por nuestro algoritmo en conjunto con otros errores obtenidos por autores con métodos en localización de *landmarks* en imágenes radiográficas de manos. Para obtener el error se

obtuvo la distancia residual que consiste en comparar las imágenes marcadas automáticamente con imágenes marcadas manualmente por un humano. Para esto se ocuparon 250 radiografías que no pertenecen al conjunto de entrenamiento.

Tabla 1. Error obtenido y los que se presentan en el estado del arte.

Autores	Error (Promedio ± Desviación estándar) mm
Doner et al. [1]	0.77±0.64
Lindner et al. [2]	0.61
Payer et al. [9]	1.13 ± 0.98
S˘tern et al. [5]	0.80 ± 0.91
Urschler et al. [4]	0.80 ± 0.93
Payer et al. [6]	0.66±0.74
Propuesto	1.15±1.07
Autores	Error (Promedio ± Desviación estándar) mm
Doner et al. [1]	0.77±0.64
Lindner et al. [2]	0.61
Payer et al. [9]	1.13 ± 0.98
S˘tern et al. [5]	0.80 ± 0.91
Urschler et al. [4]	0.80 ± 0.93
Payer et al. [6]	0.66±0.74
Propuesto	1.15±1.07

5. Conclusiones

En este artículo se ha presentado un método para la localización automática de *landmarks* en radiografías de manos, el cual ubica dos *landmarks* en el tercer hueso metacarpiano, lo cual es posible al tener las imágenes lo más parecidas posibles entre sí. Para esto se tomaron en consideración características como el contraste, la rotación, traslación y escala de cada imagen. Nuestros resultados muestran que nuestro método tiene un error de 1.15 1.07 mm similar a los del estado del arte como son: [1], [2], [3], [4], [6]. Así consideramos que nuestras contribuciones son: 1.Un método para la ubicación de la región de la mano en las radiografías, 2.Un método para corregir la posición angular de la mano, 3.Un método para la corrección de contraste. Como trabajo futuro se pretende ocupar este método para obtener regiones de interés que contengan las zonas de las epífisis en los huesos metacarpianos y los huesos carpa- les, los cuales son útiles para la estimación de la edad ósea. Donde se ha observado que un error mínimo en la localización de *landmarks* permite tener una mejor estimación de la edad ósea.

Referencias

1. Donner, R.; Menze, B.H.; Bischof, H.; Langs, G.: Fast anatomical structure localization using top-down image patch regression. In: Menze, B.H., Langs, G., Lu, L., Montillo, A., Tu, Z., Criminisi, A. (Eds.) *Medical Computer Vision. Recognition Techniques and Applications in Medical Imaging.* pp. 133–141. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
2. Lindner, C.; Bromiley, P.A.; Ionita, M.C.; tes, T.F.: Robust and accurate shape model matching using random forest regression-voting. *IEEE Transactions on Pattern Analysis and Machine Intelligence 37(9)*, 1862–1874 (2015)

3. S˘tern, D.; Ebner, T.; Urschler, M.: From local to global random regression forests: Exploring anatomical landmark localization. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (Eds.) *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016.* pp. 221–229. Springer International Publishing, Cham (2016)
4. Urschler, M.; Ebner, T.; A˚ tern, D.: Integrating geometric configuration and appearance information into a unified framework for anatomical landmark localization. *Medical image analysis 43*, (January 2018), https://doi.org/10.1016/j.media. 2017.09.003
5. Payer, C.; S˘tern, D.; Bischof, H.; Urschler, M.: Regressing heatmaps for multiple landmark localization using CNNs. In: *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2016.* pp. 230–238 (2016)
6. Payer, C.; S˘tern, D.; Bischof, H.; Urschler, M.: Integrating spatial configuration into heat- map regression based CNNs for landmark localization. *Medical Image Analysis 54*, 207–219 (may 2019)
7. Turk, M.; Pentland, A.: Eigenfaces for recognition. *Journal of Cognitive Neuroscience 3(1)*, 71–86 (Jan 1991)
8. Haghnegahdar, A.; Pakshir, H.; Ghanbari, I.: Correlation between skeletal age and metacarpal bones and metacarpophalangeal joints dimensions. *Journal of dentistry (Shiraz, Iran) 20*, 159–164 (09 2019)
9. Payer, C.; S˘tern, D.; Bischof, H.; Urschler, M.: Regressing heatmaps for multiple landmark localization using cnns. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (Eds.) *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*. pp. 230–238. Springer International Publishing, Cham (2016)

Acompañamiento Virtual una Experiencia para la Tutoría

Giovanna Palacios, María C. Santiago, Ana C. Zenteno, Gustavo T. Rubín,
Yeiny Romero, Antonio E. Álvarez
Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla
Avenida San Claudio y 14 sur, CU. Col San Manuel.
{giovanna.palacios, marycarmen.santiago, ana.zenteno, gustavo.rubin,
yeiny.romero}@correo.buap.mx , antonio.alvarez@alumno.buap.mx

Resumen: Una estrategia de apoyo en la educación es la tutoría académica. Ésta se interpreta como el proceso de orientación por medio de estrategias formativas desarrolladas por docentes de la institución con el fin de dar seguimiento a los estudiantes en el proceso formativo y de egreso, para ello se debe proveer de espacios presenciales y virtuales. Se presenta una experiencia para tutoría por medio de acompañamiento virtual que provea de mecanismos de encuentros: agenda y herramientas para dar seguimiento a los estudiantes. El caso de estudio la generación 2014. Los resultados son favorables dado que en la presencialidad era nulo o mínimo en el mejor de los casos. Por ejemplo, en la Fac. de Cs química logramos que más del 50% se conectaran para llevar acabo la tutoría, sin embargo, también se identificaron casos de resistencia al uso de la tecnología y principalmente por los docentes.

Palabras clave: Educación a Distancia, Ambientes Virtuales, Herramientas Web 2.0, Plataformas Instruccionales, Diseño Instruccional, Moodle, Tutoría Académica.

1. Introducción

En México como en casi todas las Universidades de América latina, en los últimos años han incorporado programas de tutoría, como una estrategia para mejorar la calidad de educación, ya que, en un gran número de instituciones de educación superior, en su mayoría universidades públicas se presenta un alto índice de rezago, abandono y deserción, lo cual representa una gran pérdida de recursos humanos y económicos.

Es de suma importancia prestar la mayor atención a los dos primeros semestres de la carrera, ya que es cuando los estudiantes experimentan la transición de rupturas diversas y el reto de adaptarse a situaciones nuevas, incluso a la necesidad de rectificar decisiones que pueden ser trascendentales en su vida, dado que la mayor proporción de los abandonos escolares o de deserción de los estudiantes se da precisamente en el primer año de la formación universitaria. Consideramos que la tutoría implica procesos de comunicación y de interacción de parte de los profesores; implica una atención personalizada a los estudiantes, en función del conocimiento de sus problemas, de sus necesidades y de sus intereses específicos.

La última etapa del siglo XX ha marcado en nuestro entorno universitario un momento crucial en la redefinición de los procesos de formación, y en el ajuste de los mecanismos para facilitar la transición entre los diferentes sistemas de formación y trabajo. Como consecuencia, los procesos de orientación y tutoría se consideran uno de los indicadores de calidad de las instituciones de enseñanza superior.

1.1. La Tutoría

La Asociación Nacional de Universidades e Instituciones de Educación Superior (ANUIES) define la tutoría como un proceso de acompañamiento durante la formación de los estudiantes, que se concreta mediante la atención personalizada a un alumno o a un grupo reducido de alumnos, por parte de académicos competentes y formados para

esta función, apoyándose conceptualmente en las teorías del aprendizaje más que en la enseñanza.

La Benemérita Universidad Autónoma de Puebla define la tutoría como el proceso de acompañamiento a estudiantes pertenecientes a alguno de los programas académicos de la modalidad escolarizada y de modalidades alternativas de los niveles, Técnico, Técnico Superior Universitario y Licenciatura de la Benemérita Universidad Autónoma de Puebla, con el objetivo de apoyarles a lo largo de su trayectoria académica, hasta su conclusión y promover su formación integral en los campos del conocimiento, habilidades, destrezas, actitudes y valores éticos.

1.2. La Tutoría en las Universidades

En la **Facultad de Medicina de la Universidad Nacional Autónoma de México**, definen a la tutoría como un recurso estratégico que permite fortalecer los programas educativos a través del acompañamiento a los alumnos. Esta actividad, inherente a la función docente, y a través del encuentro y comunicación entre tutor-alumno, permite potenciar el desempeño académico y refuerza las capacidades a lo largo de la estancia en la institución. Los estudiantes interesados en la tutoría realizan el registro en la coordinación posteriormente el tutor se pondrá en contacto con el alumno para agendar una cita con el fin de recabar información como formas de contacto, detección de futuros problemas que se pudieran presentar durante la trayectoria académica y establecer un plan de trabajo, toda esta información es anotada en un cuadernillo para realizar una evaluación sobre el desempeño del alumno.

En la **Universidad Autónoma de Ciudad Juárez (UACJ)**, el programa de tutorías opera desde el ingreso a la universidad, el tutor convoca a reuniones a sus tutorados. Uno de los principales problemas que enfrenta la universidad es cuando el tutor no coincide con los horarios disponibles que tiene su tutorado.

En la **Universidad Autónoma de Tabasco, e**l Programa Institucional de Tutoría contribuye a la formación integral del alumno, mejorando la calidad de su proceso educativo, la tutoría individual al estudiante se lleva a cabo en los primeros ciclos escolares, por ser la etapa de adaptación al sistema educativo universitario, y respecto a la tutoría grupal la atención que brinda el tutor es a grupos de estudiantes, de entre 15 y 25 personas, cifra que podrá variar según necesidades emergentes y generales para detectar los posibles casos de tutoría individual. Una vez que el alumno conoce a su tutor por medio del curso de inducción el tutor realiza una prueba de habilidades y destrezas (EDAOM), el tutor obtiene los resultados y moldea estrategias de aprendizaje para lograr las metas académicas.

En la **Universidad Veracruzana,** la tutoría académica consiste en el seguimiento que le da un tutor académico a la trayectoria escolar de los estudiantes durante su permanencia en el programa educativo, una vez que formas parte de la comunidad formas parte del Modelo Educativo Orientado a la Formación Integral, esto con el fin de adquirir habilidades y aptitudes para tener una mejor formación académica.

El sistema de tutorías de la **Benemérita Universidad Autónoma de Puebla (BUAP)** se encuentra inmerso en un nuevo siglo caracterizado por grandes y rápidos cambios en todos los ámbitos del desarrollo humano, social, científico, político y económico que exploran los egresados a la toma de decisiones críticas y razonadas y a la necesidad emergente de habilidades y destrezas que van más allá del aprendizaje sustentado por los planes y programas de estudio.

En la actualidad, la BUAP ha logrado consolidar un proyecto de desarrollo y mejoramiento permanente que le ha dado el reconocimiento de diversos sectores de la sociedad poblana y que, no obstante, enfrenta desafíos importantes que es necesario conocer.

1.3. Herramientas de apoyo para la educación a distancia y la tutoría

Herramientas Web 2.0
Las Herramientas Web 2.0 proveen la interacción a través de las plataformas instruccionales, estas herramientas fortalecen las estrategias pedagógicas y evaluativas como por ejemplo en el chat se pueden establecer sesiones síncronas y en el foro, donde se establece una pregunta detonadora, en tiempos asíncronos.

Plataformas Instruccionales
La plataforma Instruccional o plataformas de e-learning, campus virtual o Learning Management System (LMS) es un espacio virtual de aprendizaje orientado a facilitar la experiencia de capacitación a distancia, tanto para empresas como para instituciones educativas.

También se puede decir que son sistemas integrados de aprendizaje, gestión y seguimiento de cursos en línea, el objetivo es administrar los recursos educativos disponibles organizando el acceso a ellos, permitir la comunicación y fomentar el trabajo colaborativo.

Algunas plataformas son

- Moodle
- Chamilo
- Blackboard
- Canvas

Estas plataformas instruccionales proporcionan diversas opciones y capacidades para la administración, seguimiento, gestión y desarrollo de distinto tipo de cursos en línea. Es usual que la falta de capacitación, formación y experiencia en el uso de estos ambientes de aprendizaje se convierta en una limitante que dificulte esta tarea. No obstante, hoy en día existen muchas alternativas y opciones para lograr una capacitación básica en estos menesteres.

Por otro lado, si bien es cierto que, por lo general, la estructura de los cursos en línea en los sistemas integrados está definida por la misma plataforma, el diseño instruccional y la producción de los contenidos, la operación de éstos es responsabilidad del docente interesado en estas modalidades educativas.

Diseño Instruccional
Es el proceso a través del cual se crea un ambiente de aprendizaje, así como los materiales necesarios y actividades, con el objetivo de ayudar al tutorado a desarrollar la capacidad necesaria para lograr el aprendizaje.

Ambientes virtuales
Se conocen como ambientes virtuales al sistema o software en el cual se permite la distribución o desarrollo de contenidos para cursos online, con el fin de enriquecer la interacción de enseñanza – aprendizaje.

2. Diseño e Implementación de un ambiente virtual de tutoría.

En el modelo actual de tutoría se pueden observar las siguientes, principales, debilidades: seguimiento a tutorados, habilidades en los profesores de manejo de nuevas tecnologías, falta de claridad en las actividades propias de la tutoría y coincidencia en tiempo y espacio del tutor y tutorado. Por lo anterior, se propone una redefinición de la función del tutor y tutorado a través de la gestión, mediante una plataforma y estrategia de contacto, basada en una comunicación síncrona y asíncrona apoyándonos del uso de nuevas tecnologías y definición de actividades.

2.1 Modelo de Espacios Virtuales de Tutoría Académica

Con el uso de la tecnología se genera un espacio con herramientas web 2.0 para lograr la interacción y así eliminar el problema del tiempo y espacio, los principales elementos son los tutores y tutorados en conjunto con un modelo de comunicación y de los ambientes virtuales.

Los elementos principales de este modelo son los tutorados en conjunto con los tutores y los espacios virtuales que van a permitir la comunicación y los servicios de apoyo, vea Fig. 1.

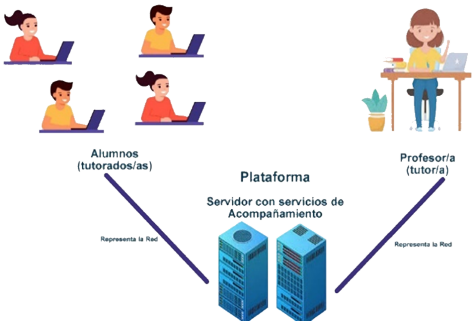


Fig. 1. Modelo de Acompañamiento Virtual

El modelo está compuesto por los siguientes elementos vea Fig. 2, la gestión de los espacios virtuales de tutoría académica, esto provoca la producción de las agendas se cuenta con dos tipos de agendas una para los tutorados y otra para el tutor así mismo la producción de las actividades propias para el tutor y tutorado, esto va a producir espacios virtuales de tutoría académica vía un diseño instruccional, finalmente a esta producción se realiza la inscripción del tutor y tutorado para que inicie la tutoría en línea.

Fig. 2. Modelo de Espacios Virtual de Tutoría Académica

a) Gestión

La gestión es el formalismo para realizar este proceso, la asignación de tutores y tutorados es un trabajo en conjunto con el coordinador de tutores, secretario académico hace la solicitud a la Dirección de Acompañamiento Universitario, BUAP, para la producción de los espacios virtuales.

b) Producción de Agendas

Se crean contenidos acordes al periodo vigente, con diseño instruccional y la planificación de actividades síncronas y asíncronas que sirvan como guía al trabajo de la tutoría, apoyándonos de herramientas que permitan la interacción y facilidad de comunicación entre tutor y tutorado resolviendo así el problema de tiempo y espacio.

La agenda del tutor contiene información sobre los servicios y programas, se solicita a su vez replique ésta a sus tutorados y así mismo se le da instrucción para que recupere información a los tutorados para que el tutor pueda realizar el seguimiento académico. La agenda del tutorado define las actividades y atención a las solicitudes del tutor a lo largo del periodo.

La primera actividad es de suma importancia ya que es necesario que el tutor realice la personalización de su perfil y coloque su fotografía con el objetivo de que el tutorado identifique a su tutor. Como actividad integradora la participación en el foro me presento y conozco a los demás con el fin de que el tutor tenga un poco más de información como los lugares de procedencia de cada uno de los tutorados. Respecto a la actividad del envió de la constancia del expediente único para tener un expediente digital de los tutorados, también es necesario que el tutor establezca horarios de tutoría presencial y virtual para poder realizar un seguimiento académico de los tutorados vea Fig. 3.

No.	AGOSTO - DICIEMBRE	ACTIVIDADES DEL TUTOR
1	Del 01 al 04 de septiembre	Personalizar su perfil - Nombre (s) Apellidos, mail y fotografía.
		Cambiar contraseña.
		Personalizar el espacio de "presentación y formas de contacto con mi tutor."
		Publicar un anuncio de bienvenida y primeras indicaciones al tutorado.
		Inicializar el espacio (foro) de dudas o comentarios.
2	Del 07 al 15 de septiembre	Apertura del espacio "Me presento y conozco a los demás."
		Guía de participación del espacio "Me presento y conozco a los demás."
		Verificar que los alumnos hayan personalizado su perfil con su fotografía, si no lo han hecho invitarlos a realizar esta actividad a través de un mensaje interno.
	Del 17 al 18 de septiembre	Cierre de participación en el espacio "Me presento y conozco a los demás."
3	Del 21 al 25 de septiembre	Publicar un anuncio para solicitar la "constancia de llenado del expediente único."
	Del 21 de septiembre al 05 de octubre	Revisar la actividad y confirmar de recibido: "constancia de llenado del expediente único."
4	Del 05 al 09 de octubre	Publicar un anuncio para comunicar a los alumnos el día (s) y horario (s) de la sesión de chat, con la finalidad de comentar con los alumnos información respecto a: plan de estudios, ruta crítica y oferta académica.
	Del 05 al 16 de noviembre	Realizar la sesión de chat en el día (s) y horario establecido para comentar con los alumnos información respecto a: plan de estudios, ruta crítica y oferta académica
5	Del 19 al 23 de octubre	Publicar un anuncio para solicitar el llenado del formato: "Proyección de materias."
	Del 19 al 30 de octubre	Revisar la realización de la actividad: "Proyección de materias", en caso de ser necesario realimentar al alumno.

Fig. 3. Contenido de la Agenda del Tutor

No.	Fechas	Actividades del tutorado: Agosto-Diciembre 2015
1	01 al 04 de septiembre	Personaliza tu perfil - Nombre (s) Apellidos, mail y fotografía.
		Cambia tu contraseña.
		Revisa la sección de anuncios para conocer las primeras instrucciones del Tutor.
2	07 al 15 de septiembre	Participa en el espacio denominado: "Me presento y conozco a los demás."
	07 al 18 de septiembre	Revisa las participaciones de los compañeros y comenta al menos a tres de ellos en relación a su presentación en el espacio: "Me presento y conozco a los demás."
3	21 al 25 de septiembre	Revisa la sección de anuncios para consultar las instrucciones o comentarios del tutor
	21 de septiembre al 05 de octubre	Envía la "constancia de llenado del expediente único." en el espacio correspondiente del apartado de "Mis actividades" en Ev@ tutor.
4	05 al 09 de octubre	Revisa la sección de anuncios para consultar las instrucciones o comentarios del tutor
	05 al 16 de octubre	Participa en la sesión de chat en el día (s) y horario (s) establecido por el tutor (a) para conocer la información respecto a tu plan de estudios, ruta crítica y oferta académica. El acceso a la sala de chat se encuentra en la sección de "Mis actividades" en Ev@ tutor.
5	19 al 23 de octubre	Revisa la sección de anuncios para consultar las instrucciones o comentarios del tutor
	19 de al 30 de octubre	Realiza la actividad: "Proyección de materias", se encuentra en la sección de "Mis actividades" en Ev@ tutor.
6	03 al 06 de noviembre	Revisa la sección de anuncios para consultar las instrucciones o comentarios del tutor
	09 al 13 de noviembre	Realiza la actividad "conoce y explora los servicios de la universidad", se encuentra en la sección de "Mis actividades" en Ev@ tutor.
7	24 al 30 noviembre	Responde la encuesta: "Acompañamiento de mi tutor.", se encuentra en la sección de "Mis actividades" en Ev@ tutor.

Fig. 4. Contenido de la Agenda del Tutorado

El tutorado también debe de personalizar su perfil para que el tutor pueda identificarlo y atender a las actividades establecidas por la agenda vea Fig. 4, participación del foro me presento y conozco a los demás como actividad de integración con sus compañeros y tutor académico, responder a la solicitud del llenado del expediente único con el fin de que el tutor pueda tener un registro digital y pueda ser consultado de manera inmediata para realizar el seguimiento.

El contenido de las agendas está basado en un diseño instruccional, se han definido un conjunto de actividades de acuerdo al caso de estudio, se cuentan con fechas de inicio y fin cada una de las actividades y se encuentran claramente identificadas que actividades deben de realizar el tutor y el tutorado

c) Producción de Espacios Virtuales de Tutoría Académica (Caso de estudio de la generación 2014)

La organización de los Espacios Virtuales está conformado por 4 DES, la económico-administrativa cuenta con 3 unidades académicas y 10 programas educativos, ingeniería y ciencias exactas conformada por 6 unidades académicas y 26 programas educativos, ciencias sociales y humanidades con 8 unidades académicas y 26 programas educativos y por último ciencias sociales y humanidades conformada por 7 unidades académicas y 17 programas educativos con un total de 418 bloques vea Fig. 5.

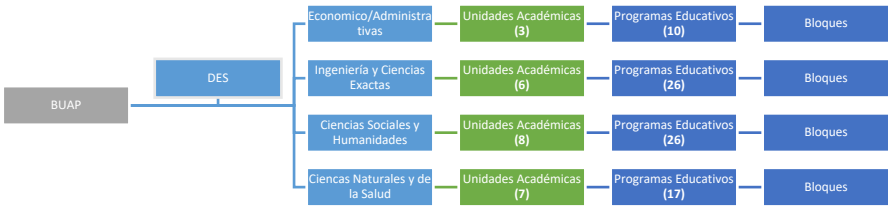


Fig. 5. Producción de Espacios Virtuales de Tutoría Académica

d) Roles

Las actividades que desempeña el tutor académico dentro de la plataforma es que este rol puede realizar cualquier acción dentro de un Espacio Virtual, agregar actividades y calificar a los tutorados, de acuerdo a la estructura de algunas actividades de la plataforma el tutor debe tener estos permisos dentro de su rol. Respecto a los tutorados los permisos son un poco más limitados como lo marcan las actividades en agenda, solo deben realizar actividades como entregas, responder test o bien realizar aportaciones a foros o chats.

El Coordinador de tutores, que en la agenda no se observan actividades definidas de éste, tiene funciones como el monitoreo de los espacios virtuales de su unidad académica, así como el monitoreo de la participación del tutor y tutorado, con el fin de poder dar seguimiento a esta importante actividad que es la tutoría académica.

e) Producción de Espacios Virtuales de Tutoría Académica

En los espacios virtuales de tutoría académica podemos observar, Fig. 6, en el apartado de color Azul se muestra la nomenclatura conformada por el bloque, programa educativo y la generación a la que pertenece, posteriormente en el apartado de color rosa se observa un mensaje de bienvenida, seguido de apartado del tutor lo podemos identificar de color amarillo aquí se encuentra la agenda del tutor y los reglamentos y normativas institucionales, en la parte derecha identificada de color rojo se muestra el nombre del tutor y coordinador de tutores y los avances obtenidos por el tutorado, de la misma manera bajo este esquema instruccional las actividades mínimas para realizar la tutoría académica.

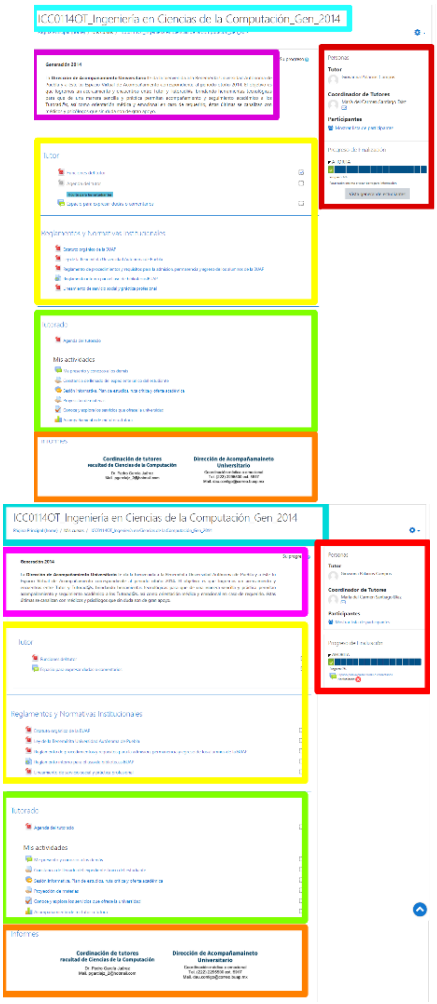


Fig. 6. Contenido del Espacio Virtual de Tutoría Académica a) Tutor y b) Tutorado

3 Pruebas

El estudio se realizó a una muestra universitaria que comprendió a 13,235 tutorados, lo que requirió la producción de 440 espacio virtual con 403 tutores, vea Tabla 1. Es importante reconocer que antes existía poco o nulo contacto con el tutorado por tanto no existía la interacción y el seguimiento.

Tabla 1. Caso de Estudio Generación 2014

Nombre	Número
DES	4
Unidades Académicas	24
Licenciaturas / Programas educativos	79
Bloques	418
Espacios Virtuales	440
Tutores	403

Tutorados	13,235

Al corte de 2 meses de uso de los espacios virtuales, se observa que en la facultad de Ciencias Químicas se tuvo una participación del 83% de los tutores, vea reporte de la Fig. 7. Además, el 44% personalizaron su perfil el 50% coloco un mensaje de bienvenida y primeras indicaciones a sus tutorados, y el 56% realizó la actividad del foro me presento y conozco a los demás.

Tutores				
Total de Tutores	18			
Acceso de Tutores	15	83%		
Actividades	Valoración de participación y cumplimiento			
	0	1	2	NA
	No se ha realizado %	Parcialmente %	Se cumplió %	%
Personaización de perfil.	44%	11%	**44%**	
Cambio de contraseña.	44%	0%	**56%**	
Anuncio de bienvenida y primeras indicaciones al tutorado.	50%	0%	**50%**	
Foro dudas o comentarios.	11%	0%	**0%**	89%
"Me presento y conozco a los demas".	44%	0%	**56%**	

Tutorados		
Total	327	
Acceso	183	**56%**
Actividades		
Perfil personalizado.		
Participación: "Me presento y conozco a los demas".		
Constancia: Expediente único.		

Fig. 7. Resultados de la Facultad de Ciencias Químicas

Se observa como caso de éxito, de 327 tutorados el 56% se conectó a la plataforma y personalizó su perfil, participó en el foro de me presento y conozco a los demás y envió la constancia del llenado del expediente único, con esto se permitió atender la problemática de tiempo y espacio, vea Fig. 7.

Ahora bien, el caso especial donde los tutorados ingresaron a plataforma solicitaron la atención del tutor como no hubo respuesta, se intervino y las acciones que se implementaron, fue la sensibilización del tutor y se les impartió un curso de capacitación en el uso de la plataforma, se observa que el 100% de los tutorados ingresaron a plataforma, personalizaron su perfil, establecieron las formas de contacto virtual y presencial vea Fig. 8.

Lamentablemente en este segundo corte perdimos a los tutorados ya que solo el 25% realizó las actividades de personalización de perfil, participación en el foro de me presento y conozco a los demás y realizaron el envío de la constancia del llenado del expediente único vea Fig. 8.

Total de Tutores	7			
Acceso de Tutores	7	100%		
Actividades	Valoración de participacion y cumplimiento			
	0	1	2	NA
	No se ha realizado %	Parcialmente %	Se cumplió %	%
Personaización de perfil.	0%	14%	**86%**	
Cambio de contraseña.	29%	0%	**71%**	
Personalizar el espacio de "presentación y formas de contacto con mi tutor."	0%	29%	**71%**	
Anuncio de bienvenida y primeras indicaciones al tutorado.	43%	0%	**57%**	
Foro dudas o comentarios.	14%	0%	**29%**	57%
"Me presento y conozco a los demas".	14%	0%	**86%**	

Total	257	
Acceso	64	25%
Actividades		
Perfil personalizado.		
Participación: "Me presento y conozco a los demas".		
Constancia: Expediente único.		

Fig. 8. Resultados Facultad de Ciencias de la Comunicación

4 Conclusiones

Se cuenta con una metodología que integra estrategias, funciones y actividades claras apoyándonos de las TIC para llevar acabo la tutoría académica.

Se Implementó una plataforma digital con uso de herramientas web 2.0 y con el apoyo de software libre en la cual permite tener encuentros entre tutor y tutorado, ésta tiene implementada una estrategia mediante una agenda de actividades con la finalidad de guiar el trabajo de tutoría para lograr encuentros entre tutores y tutorados de manera síncrona y asíncrona.

Se fortaleció la tutoría con el uso de nuevas tecnologías y sensibilización mediante capacitación en el uso de la plataforma a tutores, con el objetivo de aumentar el impacto de la tutoría académica.

Importante hay que comentar que se redefinieron las funciones del tutor y tutorado a través de la gestión y creación de actividades en plataforma, lo que permitió tener un seguimiento académico oportuno y evidencia de la actividad de tutoría académica.

Finalmente, hay que comentar que es muy satisfactorio decir que gracias a este trabajo se tuvieron verdaderos encuentros e interacción entre tutor y tutorado para poder realizar el seguimiento a los tutorados/as, por medio del uso de la plataforma.

Referencias

1. Adolfo Obaya V. y Yolanda Marina Vargas R.. (2014). La tutoría en la educación superior. 2020, de scielo Sitio web: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S0187-893X2014000400012
2. Alejandra Romo López. (2011). La tutoría: una estrategia innovadora en el marco de los programas de atención a estudiantes. México, D.F.: Consejo Editorial de Publicaciones ANUIES.
3. Aníbal Barrera García, Ing. Irina Peña Sklyar, Dr. C. Maximino Peña Matos. (2016). Diseño e implementación de un Entorno Virtual de Aprendizaje (EVA) utilizando la plataforma educativa
4. Moodle. 2020, de scielo Sitio web: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2218-36202016000200004
5. Benemérita Universidad Autónoma de Puebla. (2013). Plan de Desarrollo Institucional 2013-2017. Recuperado de http://www.pdi.buap.mx
6. Juárez, U. A. (05 de 08 de 2019). Universidad Autónoma de Ciudad Juárez. Obtenido de http://erecursos.uacj.mx/handle/20.500.11961/3718
7. Lázaro Vázquez y Romero. (2010). Diseño instruccional. 2020, de Programa de Estudios Universitarios BUAP Sitio web: http://www.peu.buap.mx/web/fes/10%20FES%20Ano%202%20No%2010/07%20Diseno_instruccional.pdf
8. Server, T. (2020, 18 marzo). ¿Qué es Moodle? La Guía Definitiva (2020). Tropical Server. https://www.tropicalserver.com/ayuda/que-es-moodle/
9. Tobar, E. (2019, 22 julio). Funciones de Blackboard y principales usos de la plataforma. Comunidad eLearning Masters | edX. http://elearningmasters.galileo.edu/2017/10/18/funciones-de-blackboard/
10. Universidad Juárez Autónoma de Tabasco. (09 de 08 de 2019). Obtenido de http://www.ujat.mx/tutorias
11. UV. Universidad Veracruzana. Obtenido de https://www.uv.mx/dgdaie/tutorias/tutoria-academica/
12. Yahaira Gamboa Villalobos. (2013). La tutoría virtual. Quehaceres para el buen desempeño. 2020, de uned Sitio web: https://www.uned.ac.cr/academica/edutec/memoria/ponencias/yaha_80.pdf

Comparativa de Herramientas para Contrarrestar Vulnerabilidades en Kali Linux

Yeiny Romero Hernández, Maria del Carmen Santiago Díaz, Gustavo Trinidad Rubín Linares, Judith Pérez Marcial, José Alberto Rendón Soto, Miguel Pérez
Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla
Avenida San Claudio y 14 sur, CU. Col San Manuel
{yeiny.romero, marycarmen.santiago, gustavo.rubin, judith.perez}@correo.buap.mx,
albertors280996@gmail.com, miguel.perezxi@alumno.buap.mx

Resumen. Hoy en día el uso del internet ha aumentado considerablemente y a la par de ello han aumentado los ataques en la red, cualquier dispositivo conectado a la red puede ser víctima de algún ataque por lo que es de gran importancia detectar las vulnerabilidades de nuestro dispositivo antes de que sufra un ataque, en este trabajo se exponen los diferentes tipos de ataques a los que se está expuesto al conectarse a la red al igual que se prueban algunas herramientas en Kali Linux para detección de vulnerabilidades con el objetivo de proteger los dispositivos una vez que se analizan algunas de ellas.

Palabras Clave: Ataque, Vulnerabilidades, Herramientas Kali.

1 Introducción

Se considero que las vulnerabilidades de los sistemas de información son el principal problema por solucionar dado que conlleva fuertes consecuencias la fuga de información y por nada se debe permitir que esto suceda. En el presente trabajo y bajo ese contexto se hizo la investigación de las vulnerabilidades a las que se encuentra expuesta la red y la información, asimismo las herramientas que evalúan y refuerzan la seguridad poniendo en práctica algunas de ellas.

1.1 Antecedentes

Uno de los principales problemas que hoy en día inquietan a los usuarios y al mismo tiempo a grandes empresas, es saber si su información está completamente segura o si su sistema está expuesto a vulnerabilidades que permitan sufrir un ataque el cual pone en riesgo la información.

Una vulnerabilidad es el riesgo que una persona, sistema u objeto puede sufrir frente a peligros inminentes. En ámbitos computacionales se refiere a los puntos débiles de un sistema donde su seguridad informática no tiene defensas necesarias en caso de un ataque.[1]

Los encargados de distribuir los sistemas operativos suelen ser los principales responsables de una mala configuración, dejando la seguridad completamente abierta. El aseguramiento de las contraseñas de cuentas de usuarios es de suma importancia porque pueden no estar bien establecidas o incluso tener combinaciones de inicios de sesión y contraseñas fácilmente descifrables. Los ataques y las filtraciones de datos han causado estragos en grandes compañías. En el verano de 2018, el portal de búsqueda de empleo Jobandtalent sufrió un ataque que dejó descubierto los datos de más de 10 millones de

usuarios españoles. Los datos que robaron fueron nombres, apellidos, direcciones de correo y una versión cifrada de sus contraseñas.[2]

Equifax una de las mayores firmas financieras alrededor del mundo sufrió una filtración de datos que equivalía a 143 millones de usuarios afectados en sus cuentas de crédito o de débito.[3]

Se hace mención que a partir de las denuncias realizadas por Edward Snowden (2014) "Si resides en Nueva Zelanda, te están vigilando", "Nuestras palabras son interceptadas, almacenadas y analizadas por algoritmos incluso antes de que las pueda leer el receptor", en dicha obra se ponía en evidencia la capacidad de recolección de datos y de intersección de las comunicaciones con que cuenta el gobierno de Estados Unidos realizadas sin el consentimiento de los ciudadanos [4]

Como mostró Stoll Clifford (1988), en su artículo "Stalking the Wily Hacker" publicado en el Profesional Journal Communications of the ACM se trata de un artículo científico que subraya las técnicas utilizadas por el hacker para infiltrarse en el sistema, se menciona como el intruso obtuvo el acceso a los sistemas mediante una combinación de inicio de sesión y contraseñas como 'Sistema / administrador'.[5]

Por esta razón, las vulnerabilidades de los sistemas son el principal problema por solucionar puesto que conlleva fuertes consecuencias la fuga de información.

El uso de las tarjetas de crédito, correos electrónicos, aplicaciones de compra y venta, es decir, toda aquella actividad que utilice de manera electrónica datos personales requiere que la privacidad de la información sea protegida durante el uso de estos servicios.[6][15]

Para afrontar este problema se necesita estar bien informado acerca de las técnicas de seguridad en redes que rigen en la actualidad, así como poder llevarse a cabo y saber cuál de ellas es la mejor opción para cumplir la misión fundamental de proteger la información de vulnerabilidades existentes.

1.2 Vulnerabilidades actuales

En la actualidad, el mundo accede de algún modo a internet. Con tantos accesos concurrentes a la red de redes, la posible amenaza de seguridad a los sistemas informáticos crece y se complejiza, a pesar de las diversas y especializadas maneras de contrarrestarlas. La seguridad de la información en la red es más que un problema de protección de datos, y debe estar básicamente orientada a asegurar la propiedad intelectual y la información importante de las organizaciones y de las personas. Los riesgos de la información están presentes cuando confluyen dos elementos: las amenazas de los ataques, y las vulnerabilidades de la tecnología; conceptos íntimamente relacionados donde no es posible ninguna consecuencia sin la presencia conjunta de estos.[7][13][14]

Las amenazas, pueden venir de cualquier parte, agentes externos, personal interno, está muy relacionada con el ambiente de las organizaciones.[7] Las vulnerabilidades son zonas de riesgo en la tecnología o en los procesos asociados con la información. Existen personas o usuarios de internet que emplean las tecnologías para vulnerar los sistemas en la red y robar información, son comúnmente conocidos como hackers.[8][16]

Algunos de los principales ataques en la red son los malwares (software malicioso que daña un sistema). Comúnmente los tipos de malware son virus, gusanos, troyanos, spyware y de los más dañinos el ransomware (secuestra datos encriptándolos y pedir un rescate por ellos).[9][11][16]

Por mencionar algunos malwares más tenemos el Phishing (suplantación de identidad) los Botnets (Redes de robots) y la denegación de servicios(inhabilita el uso de un sistema o computadora) [10][11][16]

Para realizar alguno de estos ataques, los intrusos deben disponer de los medios técnicos, los conocimientos y las herramientas adecuadas y se tiene que dar además una

oportunidad que facilite el desarrollo del ataque, como podría ser un fallo en la seguridad del sistema informático elegido.[12][16]

2 Estado del Arte

2.1 Modelo TCP/IP

Para hablar sobre vulnerabilidades y medidas de seguridad en la red de los servidores es importante conocer la estructura de funcionamiento. Para ello el modelo TCP/IP detalla cómo se transmite la información a través de la red. Recordando que un modelo de red sirve para interconectar equipos computacionales que utilizan diferentes sistemas operativos, diferentes tarjetas de red. En particular, el modelo TCP /IP está dividido e 4 capas: capa de red (los equipos conectados a internet se implementan en esta capa), capa de internet (permite interconexión, la dirección y el encaminamiento son sus principales funciones), capa de transporte (control de flujo y de errores), capa de aplicación (aplicaciones que se utilizan en internet: clientes y servidores).[10][17][18]

2.1.1 Protocolos

Otro concepto relacionado al modelo TCP/IP y que da soporte y flexibilidad a la red es su familia de protocolos que hace posible la comunicación entre las capas.[10]

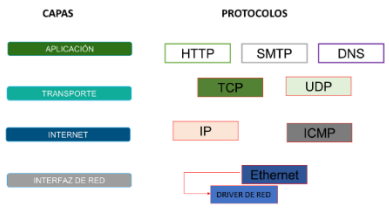


Fig. 1. Capas y sus respectivos protocolos del modelo TCP/I

Esta estructura es fundamental para efectuar el análisis ya que aquí es donde se producen vulnerabilidades. [10] en cada capa del modelo TCP/IP pueden existir distintas vulnerabilidades que puede ser aprovechadas por los atacantes para explotar los protocolos asociados a cada una de ellas.

Por ejemplo: *la capa de red*, aquí los ataques requieren un acceso físico a los equipos que se quieren atacar; *capa internet*, aquí puede realizar cualquier escucha de red (snnifing), la suplantación de mensajes, la modificación de datos y la denegación de mensajes; *capa de transporte*, aquí se pueden encontrar problemas de autenticación, de integridad y de confidencialidad (denegaciones de servicio), además de la interceptación de sesiones TCP establecidas; *capa de aplicación*, tiene varias deficiencias de seguridad asociadas a sus protocolos.[10][17][18]

La capa de aplicación de la pila TCP ofrece a las aplicaciones la posibilidad de acceder a los servicios de las demás capas y define los protocolos que utilizan las aplicaciones para intercambiar datos, como correo electrónico (POP y SMTP), gestores de bases de datos y protocolos de transferencia de archivos (FTP**, SSH), consulta de páginas web (HTTP /HTTPS), etc**. Esta capa contiene las aplicaciones visibles para el usuario **donde se tiene que verificar la s**eguridad **y el c**ifrado

2.2 Herramientas de seguridad

Antes de planear un posible ataque contra una o varias computadoras de una red es necesario saber que tenemos que atacar. Se necesita obtener toda la información de la víctima, y para eso se utilizan las técnicas de obtención y recolección de información.[10][19][20]

2.2.1 NMAP

Es una herramienta gratuita y de código abierto utilizada por los administradores del sistema para descubrir redes y auditar la seguridad. Es rápido en su funcionamiento, está bien documentado, cuenta con interfaz gráfica. Admite transferencia de datos, inventario de red, etc. [10][19][20][21]

2.2.2 Nessus

Es una herramienta de escaneo remoto que puedes usar para verificar las vulnerabilidades de seguridad de los ordenadores. No bloquea activamente las vulnerabilidades que tienen, pero podrás detectar ejecutando más de 1200 comprobaciones de vulnerabilidad, lanzando alertas cuando sea necesario realizar parches de seguridad. [19][20][21]

2.2.3 John The Ripper

Otra herramienta popular de cracking utilizada en la comunidad de pruebas de penetración (y hacking). Inicialmente fue desarrollado para sistemas Unix, pero ha crecido para estar disponible en más de 10 distros. Cuenta con un cracker personalizable, detección de hash de contraseña automática, ataque de fuerza bruta, y ataque de diccionario (Entro otros modos de cracking).[19][20][21]

3 Desarrollo

3.1 Hacking

Se llevo a cabo prácticas de hacking ético para el desarrollo y representación de este trabajo de investigación. Pero, ¿Qué es él hacking? se puede definir como “la búsqueda y explotación de vulnerabilidades de seguridad en sistemas o redes”. consiste en la detección de vulnerabilidades de seguridad, y explotación de las mismas. [10][11][13]

3.2 Selección de herramientas

Existen diversas metodologías open source, o libres las cuales tratan de dirigir o guiar los requerimientos de las evaluaciones en seguridad. La idea principal de utilizar una metodología durante una evaluación es ejecutar diferentes tipos de pruebas paso a paso, para poder juzgar con una alta precisión la seguridad de los sistemas. Por lo tanto, se procedió a usar herramientas libres que ya algunas vienen instaladas en el sistema Kali Linux y ofrecen una gran manejabilidad, documentación suficiente y destacada para poner en obra las pruebas de seguridad. [10][19][20][21]

La principal diferencia entre una evaluación de vulnerabilidades y una prueba de penetración radica en el hecho de las pruebas de penetración van más allá del nivel donde únicamente se identifican las vulnerabilidades, mientras una evaluación de vulnerabilidades proporciona una amplia visión sobre las fallas existentes en los sistemas, pero sin medir el impacto real de estas vulnerabilidades para los sistemas objetivos de la evaluación. [10][19][20][21]

3.3 Pruebas

Las pruebas fueron implementadas, con previa autorización, en el entorno de desarrollo y virtualización del centro de datos de la Dirección General de Cómputo y Tecnologías de la Información y Comunicaciones (DCyTIC) de la Benemérita Universidad Autónoma de Puebla (BUAP).

La selección de las herramientas se hizo de menor a mayor prioridad, así mismo su complejidad comenzando por la herramienta NMAP seguida de la herramienta Nesuss y terminando la fase de pruebas con John de Ripper

3.3.1 Network Mapper (NMAP)

Utiliza paquetes IP en bruto de maneras novedosas para determinar cuáles hosts están disponibles en la red, cuales servicios (nombre y versión) estos hosts ofrecen, cuales sistemas operativos (y versión de SO) están ejecutando. Ha sido diseñado para escanear velozmente grandes redes, consecuentemente funciona también con host únicos.

Para realizar el escaneo en la línea de comandos de la máquina de Kali Linux se ejecuta el siguiente comando.

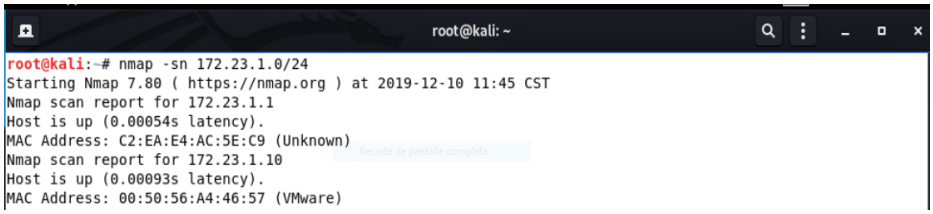


Fig. 2. Escaneo simple con NMAP a toda la subred

La opción -sn le indica a la herramienta de nmap no realizar el escaneo de puertos después del descubrimiento de host en la subred, solo se imprime los hosts disponibles que responden al escaneo.

3.3.2 Nessus

Una más de las herramientas que se consideran importantes en el escaneo de vulnerabilidades, esta herramienta no viene instalada en Kali por lo que se tiene que instalar.

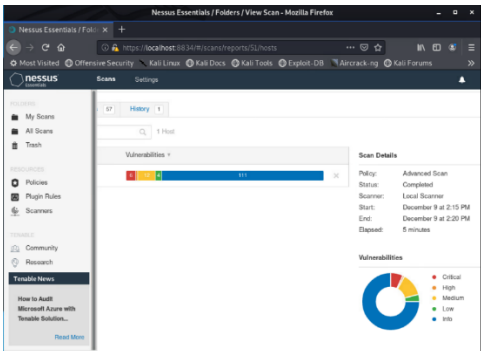


Fig. 3. Interfaz de la herramienta Nessus

Una posible desventaja es que es de licenciamiento por lo que se tiene que pagar, pero cuenta con una versión de prueba gratuita de 30 días la cual permite probarla y comprobar sí es de gran utilidad como se espera para posteriormente adquirirla.[10][21]
Se utiliza mediante web, la instalación es sencilla y la manipulación lo es aún más, su funcionalidad principal permite realizar un escáner del servidor a detalle el cual al finalizar detalla cuantas vulnerabilidades encontró, de que gravedad, ya que son categorizadas por niveles mediante colores y para más detalles al ingresar a cada tipo de vulnerabilidad indica cómo se puede resolver y a la vez como y mediante que es que puede ser explotada.[21]

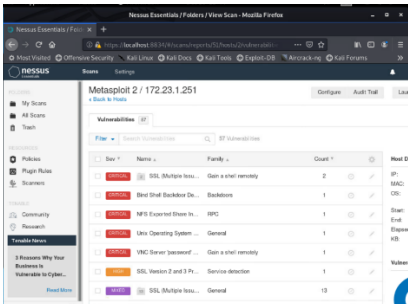


Fig. 4. Listado de vulnerabilidades halladas con Nessus

Si se da clic en las vulnerabilidades de color rojo que son las más críticas despliega un menú de todas las vulnerabilidades encontradas cada una de ellas con su descripción general que al ingresar la detalla un poco más.

En este caso particular la herramienta es capaz de indicar con que es posible explotar dicha vulnerabilidad.

El siguiente ataque se llevó a cabo con el comando rpcinfo

Comando: rpcinfo -p 172.23.1.251genera información sobre el servicio de llamada de procedimiento remoto RPC que se ejecuta en el sistema por consiguiente al ser visible NFS se procede a crear un nuevo directorio en la maquina atacante.

Comando: mkdir target para después ser montado el directorio NFS y tener acceso a la maquina victima la cual toda información compartida a ese directorio podrá ser vista por el atacante.

Comando: mount -t nfs 172.23.1.251:/ target

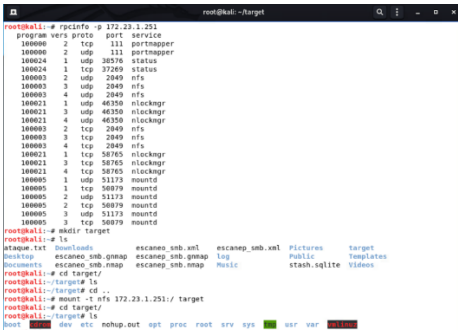


Fig. 5. Explotación de la vulnerabilidad NFS

3.3.3 John The Ripper

Otro de los ataques que se logra dar es con la herramienta John The Ripper una de las herramientas más peligrosas porque es capaz de visualizar las contraseñas de los usuarios aun estando cifradas. Para realizar la explotación de esta vulnerabilidad se hace uso nuevamente del ataque al directorio NFS montado desde la maquina Kali, obtenido el acceso se dirige a la directiva del archivo shadow y passwd los cuales son los archivos críticos que contienen los usuarios del sistema, así como su contraseña con la particularidad de que se encuentra encriptada por lo que si se lista el contenido del archivo no es posible apreciar a simple vista la verdadera contraseña.[21]

Para mostrar el contenido del archivo se ejecuta el comando que muestra los usuarios y contraseñas encriptadas.

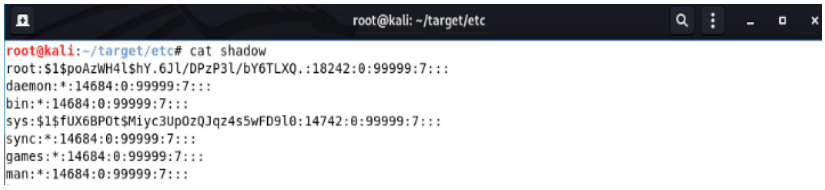


Fig. 6. Contenido del archivo shadow de la maquina vulnerable

También se lista el contenido del archivo que contiene a todos los usuarios del sistema de la maquina vulnerada, así como su respectivo directorio que tienen asignado.

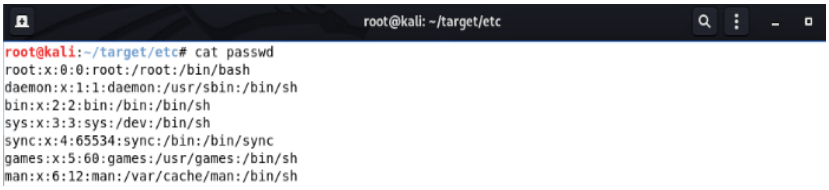


Fig. 7. Contenido del archivo passwd de la maquina vulnerable

Entonces, con el control total sobre los archivos shadow y passwd se hace una copia de estos mismos a una ruta de la maquina atacante para que desde ahí se realice el ataque por parte de John The Ripper

cp shadow /root/ataque-shadow

cp passwd /root/ataque-passwd

Ahora se tienen que unir los dos archivos en uno mismo para que la herramienta de John The Ripper efectue la desencriptación mediante fuerza bruta con la información de un solo archivo. Para ello se ejecuta

Comando: ***unshadow ataque-passwd ataque-shadow > ataque-fusion.txt***

Como punto importante el comando no da ninguna salida de ser ejecutado correctamente y solo se corrobora listando para comprobar que el nuevo archivo fue creado.

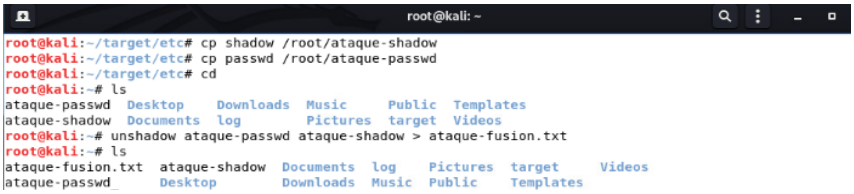


Fig. 8. Copia y fusión de archivos shadow y passwd

Con el archivo fusionado y la ayuda de un diccionario predeterminado que contiene cientos de contraseñas posibles con las cuales se van a comparar las encriptadas, se ejecuta el ataque de John The Ripper con el comando

Comando***: john --wordlist=password.lst ataque-fusion.txt*** pasando como argumento la clave --wordlist=password.lst que es el diccionario que ocupara para llevar a cabo la fuerza bruta y el archivo fusionado con el contenido de los usuarios y contraseñas ataque-fusion.txt

4 Resultados

Los resultados obtenidos son sorprendentes, para hacer uso de cada herramienta se siguió una secuencia de menor a mayor complejidad. Al llevar a la practica NMAP a toda la subred se visualizaron todos los hosts conectados que emitieron respuesta y efectivamente se encontraban conectados, sin pasar ningún argumento se desplego a detalle la dirección IP de cada host conectado, así como el listado de sus puertos, el estado en que se encontraron y el servicio que corría por cada uno de estos, su dirección MAC. Obteniendo un total de 236 equipos conectados a la subred escaneada.

Al utilizar Nessus en un ámbito web con una interfaz muy bien presentable porque fácilmente pudo informar que tan protegido o vulnerable estaba el sistema que se analizó y lo mejor de todo es que al visualizar cada resultado obtenido explico a detalle a que se debe cada vulnerabilidad. Por ejemplo, se usó “NFS Exported Share” dio la información de lo que se refiere la vulnerabilidad y en el apartado inferior derecho indico con que se puede explotar esa vulnerabilidad, es fácil llevar a cabo este tipo de ataques, básicamente lo que nos dice esa vulnerabilidad es que el servidor está expuesto a que el atacante pueda montar un directorio NFS con lo cual tendrá acceso y manipulación. El resultado fue exitoso ya que se pudo montar el directorio NFS y al acceder a él se pudo ver toda la información de la maquina atacada del usuario root.

Teniendo los privilegios de root se pudo visualizar el archivo de usuarios y contraseñas del sistema, Para este ejemplo en particular se creó un nuevo usuario en la maquina atacada llamado alberto el cual se le asigno una contraseña fácilmente descifrable ya que muchos cometen ese error, la contraseña en este caso es la ‘qwer1234’ que para muchos a veces se les hace fácil recordar porque son teclas que están consecutivas en el teclado al igual que la sucesión de números y consideran una contraseña fiable por contener letras y números. Pero lamentablemente no es así, ya que fácilmente con el ataque de John The Ripper la contraseña puede ser descubierta debido a su popularidad a pesar de su encriptación.

5 Conclusiones

Entender el estado actual de la seguridad en la red es importante para poder proteger los equipos, los datos, los negocios de las amenazas. Las estadísticas encontradas, a pesar de todas las medidas que se toman, los profesionales y las empresas no logran contener la expansión de los ataques. Entonces para proteger la información el primer paso es ser ético y mantenerse actualizado acerca de las últimas amenazas. El siguiente paso es invertir en seguridad y estar un paso adelante de los atacantes. También es importante el uso de estas y todas las herramientas que tiene como objetivo realizar las pruebas de seguridad y penetración. Sin embargo, haciendo énfasis en las 3 herramientas que se utilizaron se llegó a la conclusión de que todas y cada una son muy útiles y que van de la mano una con la otra. Que ellas mismas se pueden unir para generar un gran plan de detección de vulnerabilidades en los sistemas y con ello reforzar la seguridad para cuidar la información. Sin duda alguna la herramienta más destacable es Nessus, el trabajar con el entorno web, la interfaz gráfica sencilla pero muy completa, muy minuciosa en el escaneo de los sistemas la han llevado a que sea catalogada como la mejor en su rama, recordando que es de licenciamiento y será una buena inversión.

6 Trabajo Futuro

- Uso de otras herramientas que tienen por objetivo evaluar la seguridad y contrarrestar las vulnerabilidades instaladas en el sistema operativo Kali Linux.
- Tener presente la creación de entornos seguros, con reglas establecidas en las políticas de seguridad de la información, así mismo fomentar la ética en el personal dando una mayor capacitación y establecer mecanismos que contrarresten malas acciones.

References

1. Significado de vulnerabilidad (12 de noviembre de 2018). Significado de vulnerabilidad. Obtenido de Significados: [en línea] https://www.significados.com/vulnerabilidad/
2. Blogthinkbig.com (22 febrero 2019). Mayores filtraciones del 2018. Obtenido de: [en línea] https://blogthinkbig.com/filtraciones-datos-ciberseguridad-2018
3. Raúl Álvarez. (7 Septiembre 2017Actualizado 11 Enero 2018). Los datos de 143 millones de personas filtrados ante el hackeo a Equifax, una de las mayores agencias crediticias. 25 /agosto / 2020, de XATAKA Sitio web: https://www.xataka.com/seguridad/hackean-equifax-una-de-las-mayores-agencias-de-informes-crediticios-afectando-a-143-millones-de-usuarios
4. La Información. (15/09/2014). Snowden: "Si Resides En Nueva Zelanda, Te Están Vigilando". 25/08/2020, De La Información Sitio web: https://www.lainformacion.com/espana/snowden-si-resides-en-nueva-zelanda-te-estan-vigilando_3yizFYpJA5Patlf3yvEyT1/
5. Clifford Stoll. (1988). Stalking The Wily Hacker. Agosto 2020, De Communication Of The Acm Sitio web: http://pdf.textfiles.com/academics/wilyhacker.pdf
6. David Jiménez Domínguez. (2018). Privacidad de la Información. Ataques Dirigidos. Agosto 2020, De Revista .Seguridad | 1 251 478, 1 251 477 | Revista Bimestral Sitio Web: Https://Revista.Seguridad.Unam.Mx/Numero-05/Privacidad-De-La-Informaci%C3%B3n-Ataques-Dirigidos
7. b2b Consultores (enero 10, 2019). Amenazas y vulnerabilidades de los sistemas informáticos. Obtenido de: [en línea] http://btob.com.mx/ciberseguridad/amenazas-y-vulnerabilidades-de-los-sistemas-informaticos/
8. Josep Jorba Esteve, Remo Suppi Boldrito. (2004). Administración avanzada de GNU/Linux. Fundació per a la Universitat Oberta de Catalunya: U F o r m a c i ó n d e P o s g r a d o.

9. OpenWebinars (19 agosto 2019) Que es el hacking. Obtenido de: [en línea] https://openwebinars.net/blog/que-es-el-hacking/
10. Jordi Herrera Joancomartí (coord.) Joaquín García Alfaro, Xavier Perramón Torni. (2004). Aspectos avanzados de seguridad en redes. Catalunya: UOC F o r m a c i ó n d e P o s g r a d o.
11. López Gómez, Julio. (2010). Guía de campo de hackers aprender a atacar y defenderte, México, Alfaomega, 180 p.
12. Daniel Bechimol.(2011).Hacking desde cero. Argentina: Fox Andina S.A. Recuperado de: https://www.freelibros.me/ Consultado el 20 de julio de 2020
13. Gómez Vieites, Álvaro. (2011). "Enciclopedia de la seguridad informática", 2da edición, RA-MA, 830 p.
14. Daltabuit, Enrique. (2007). "La seguridad de la información", México, Limusa, 774 p
15. Buch, Jordi; Jordan, Francisco. (2006). "Seguridad en las transacciones en internet"
16. Picouto Ramos, Fernando; Lorente Pérez, Iñaki; Ramos Varón, Antonio. (2008). "Hacking y seguridad en internet", 1ra Edición; México; Editorial Alfaomega.
17. Siles, Raul. (2002). "Análisis de seguridad de la familia de protocolos TCP/IP y sus servicios asociados", edición 1.
18. Gont, F. (2008). "Security Assessment of the Transmission Control Protocol (TCP)", Centre for the protection of national infrastructure (CPNI), Estados Unidos.
19. Garfinkel, Simson; Spafford, Gene (2000). "Seguridad practica en Unix e internet", 2da edición, McGraw Hill Interamericana, 835 p.
20. Jakub, Zemanek. (2005). "Cracking sin secretos. Ataque y defensa de software", España, RA-MA, 400 p.
21. MasGNULinuX (2019) Las mejores 20 herramientas de hacking y penetración para Kali Linux. Obtenido de: [en línea] https://maslinux.es/las-mejores-20-herramientas-de-hacking-y-penetracion-para-kali-linux/

Integración de Subsistemas de Modelo Satelital Tipo CanSat

Hermes Moreno[1], Fernando Martínez[1], Ma. del Carmen Santiago[2], Gustavo Rubín[2], Antonio Álvarez[2], Joel Contreras[3], Enríque García[3]
[1]Facultad de Ingeniería, Universidad Autónoma de Chihuahua
Circuito Universitario, Campus II Chihuahua, Chih. México
[2]Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla
Avenida San Claudio y 14 sur, CU. Col San Manuel, Puebla, México
[3]Universidad Popular Autónoma del Estado de Puebla
{hmoreno, fmartine}@uach.mx, {marycarmen.santiago, gustavo.rubin}@correo.buap.mx,
antonio.alvarez@alumno.buap.mx, ing.joelcontreras@gmail.com,
enriquerafael.garcia@upaep.edu.mx

Resumen. En este trabajo se describe la implementación de un prototipo CanSat, cuya misión es mostrar el desempeño de la capa de telemetría del sistema CanSat. El sistema reporta información ambiental en tiempo real. Para la comunicación entre el CanSat y la estación en tierra, se ha utilizado la tecnología Xbee, la computadora de vuelo lo simula un microcontrolador Atmel ATMEGA. Para el sistema de video se consideró un equipo SkyZone que cuenta con 8 canales, cada uno con un ancho de banda de 8 MHz. Por otro lado, la estación en tierra se integra por tres componentes: repositorio de información, servicio de conexión a la nube y monitor de eventos.

Palabras Clave: Telemetría, CanSat, Computadora de Vuelo, Sistema de Video.

1 Introducción

El dearrollo e implementación de los CanSat ha sido un excelente programa de educación práctica sobre ingeniería espacial para estudiantes universitarios, ganando impulso año tras año y extendiéndose por todo el mundo. Un CanSat se puede considerar una plataforma o modelo satelital de bajo costo, en dónde se pueden simular diferentes subsistemas de un satélite profesional. Uno de los objetivos del CanSat es reportar la información ambiental en tiempo real, así como la información de la trayectoria que toma durante el ascenso y descenso del mismo. El diseño de un CanSat implica un reto de ingeniería ya que éste, por lo general, se debe ajustar a las dimensiones de una lata de refresco, además de no sobrepasar algunas especificaciones sobre el peso. Lo anterior implica tener ciertas limitantes en el diseño mecánico, electrónico, y en sus componentes estructurales.

Un CanSat involucra varias disciplinas, por nombrar algunas de ellas se encuentra el control automático, aerodinámica, física atmosférica, comunicación, programación, etc. Generalmente, un CanSat básico consiste en una serie de sistemas fundamentales que hacen al aparato funcionar. Dentro de los sistemas existen varios elementos que son parte de estas secciones como lo son por ejemplo: microcontrolador, acelerómetro, sensores de presión y temperatura, cámara, estructura, paracaídas, entre otros.

El propósito para el diseño del CanSat abarca dos fuertes motivaciones, la primera es la contribución académica que se logra con respecto a la preparación conceptual de los estudiantes al acercarlos a un sistema que no está lejos de lo profesional, donde ellos pueden experimentar aspectos de diseño y construcción de satélites artificiales. Segundo, a través del CanSat se quiere monitorear y visualizar el comportamiento del "pico satélite", así como su entorno durante el periodo de ascenso y descenso en espacio abierto.

El CanSat cuenta con algunos subsistemas, entre los cuales se encuentran el subsistema de telemetría, el subsistema de comunicación con estación base, la computadora de vuelo y el subsistema de video. La estación en tierra además ofrece un sistema de visualización en la nube. Cada uno de los subsistemas se describe en las secciones siguientes.

2 Subsistemas del CanSat

Como se muestra en la Figura 1, el CanSat cuenta con una capa de sensado y su canal de comunicación con la estación en tierra. De forma independiente se tiene el subsistema encargado de la transmisión de video hacia la estación en tierra.

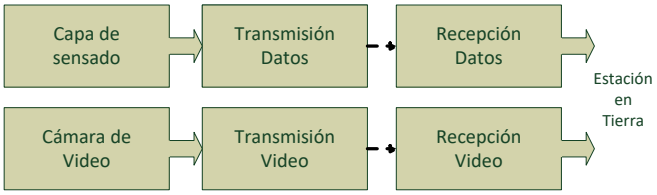


Fig. 1. Subsistemas que integran el CanSat

2.1 Capa de Sensado

Esta capa se encarga de recibir los datos de cada una de las tecnologías de sensores que se utilizan para monitorear el desempeño del CanSat en su ascenso y descenso. La Tabla 1, describe cada una de las tecnologías de sensores utilizadas en el CanSat.

Tabla 1. Sensores utilizados en el CanSat

Sensor	Descripción	Fabricante	Rango nominal
LM35	Sensor de temperatura (externa)	Texas Instruments	−55 °C a 150 °C
DTH11	Sensor de temperatura (interna)	Mouser Electronics	0-50 °C
SW-18020P	Sensor de vibración	Bailing Electronics	PPS
DTH11	Sensor de humedad	Mouser Electronics	20-90%RH
MPU-6050	Sensor de aceleración	Motion Tracking	±2g, ±4g, ±8g,
BMP180	Sensor de altitud	Bosch Sensortec	9000m a -500 nivel del mar

Cada uno de los sensores presentados en la Tabla 1, fueron evaluados, e integrados paulatinamente a la computadora de control, como lo muestra la Figura 2. Con estas tecnologías de sensores se recolecta la información de la temperatura interna y externa del CanSat, así como la humedad relativa en el interior. Adicionalmente se monitorea el nivel de vibración del CanSat durante el ascenso y descenso en campo libre. Durante este proceso también se recolecta información de la altitud y de la presión atmosférica.

2.2 Comunicación

La capa de comunicación entre el cansat y la estación en tierra utiliza la tecnología XBEE. Esta tecnología es ampliamente utilizada para la transmisión de datos debido a su bajo consumo de energía. En la figura 2, se observa el módulo en donde se aloja el dispositivo XBEE.

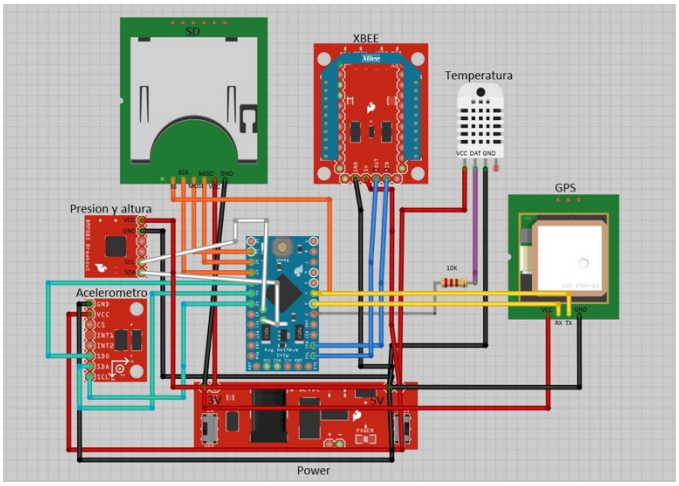


Fig. 2. Diagrama de conexión de los diferentes subsistemas del CanSat.

2.3 Computadora de vuelo

La computadora central del sistema CanSat está basada en un microcontrolador ATMEGA328 de la familia Atmel. El Atmega328 AVR es un microcontrolador de 8-bit de alto rendimiento el cual tiene arquitectura RISC, combinado con 32KB de memoria flash, 1KB de memoria EEPROM, 2KB de SRAM, 23 líneas de E/S de propósito general, 32 registros de propósito general, tres temporizadores/contadores con modo de comparación y captura, interrupciones internas y externas, programador de modo USART, una interfaz serial orientada a byte de 2 cables, puerto serial SPI, un Convertidor A/D de 6 canales y 10-bits de resolución, watchdog timer programable con oscilador interno, y cinco modos de ahorro de energía seleccionables por software. El dispositivo opera entre 1.8 y 5.5 voltios. Por medio de la ejecución de poderosas instrucciones en un solo ciclo de reloj, el dispositivo alcanza una respuesta de 1MIPS, balanceando consumo de energía y velocidad de proceso.

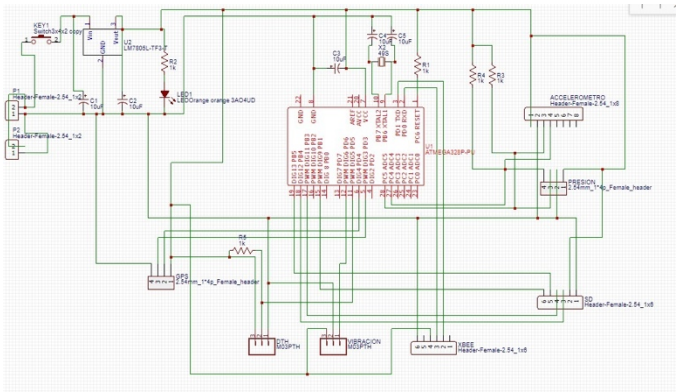


Fig. 3. Microcontrolador ATMEGA como elemento de procesamiento central del sistema CanSat.

La Figura 3 muestra el diagrama esquemático de las conexiones entre los pines del microcontrolador ATMEGA (al centro color rojo) y los diferentes sockets donde estarán conectados las diferentes tecnologías de sensado.

2.4 Sistema de video

Como se mencionó, el sistema de video opera por separado del sistema principal. El sistema de video es un equipo SkyZone que cuenta con 8 canales, cada uno con un ancho de banda de 8 MHz, lo cual le permite realizar transmisión de video en distancias de hasta 500m. En la Figura 4 se muestra el equipo SkyZone el cual consta de una cámara 700TVL CMOS de FatShark, un módulo de transmisión y receptor de video y cables de conexión. Su consumo de voltaje está en el rango de 3.5 a 5 volts, y el consumo de corriente es del orden de los 60mA operando a 5V. La resolución de la imagen es de 726(H) por 582(V).

Fig. 4. Sistema de video para el prototipo CanSat

3 Estación en Tierra

La estación en tierra representa el medio a través del cual es posible monitorear el comportamiento del CanSat. La estación de tierra está formado por dos módulos: el canal de video junto con el canal de datos provenientes de los sensores y el canal de visualización. El canal de visualización se integra por tres componentes: repositorio de información, servicio de conexión a la nube y monitor de eventos.

El despliegue de datos se realiza mediante un conjunto de aplicación con una arquitectura Web Cliente/Servidor dividido en componentes (Figura 5). Los componentes son los encargados de la transmisión de datos a la nube y componentes de despliegue para mostrar los datos en dispositivos móviles, PC y laptops.

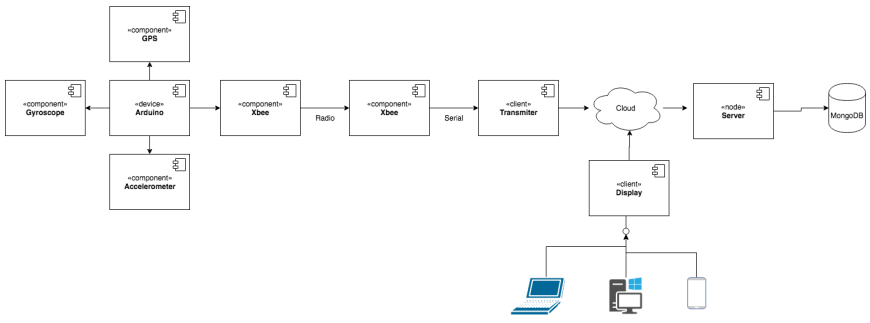


Fig. 5. Diagrama a bloques del sistema completo CanSat.

En las Figuras 6 y 7 se muestra un ejemplo de la interface de visualización de los datos provenientes del magnetómetro, acelerómetro y giroscopio, junto con la localización en Google Maps del dispositivo Cansat utilizando las coordenadas del módulo GPS. Cabe mencionar que para la localización del CanSat se está utilizando un módulo GYNEO6MV2 el cual opera en el rango de 3.5 a 5 volts con un consumo de corriente de 45mA.

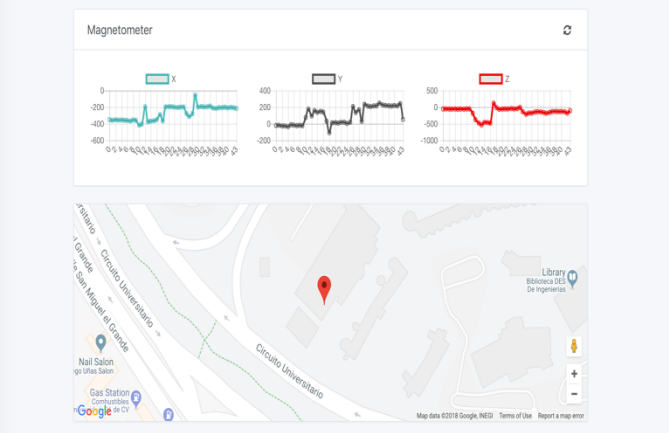


Fig. 6. Ejemplo de la vista de eventos del magnetómetro en sus tres ejes y visualización de las coordenadas GPS utilizando la API de Google Maps.

Fig. 7. Vista de eventos del acelerómetro y del giróscopo en el servicio Cloud de la estación en tierra.

4 Integración

El prototipo actual del CanSat se muestra en la Figura 8, al centro se observa el microcontrolador ATMEGA, y a los costados el módulo GPS junto con los sensores de telemetría. Este prototipo está montado en una baquelita perforada donde se ha añadido un circuito regulador de voltaje para reducir el voltaje proveniente de las pilas a un potencial de 5V. Cabe mencionar que las baterías utilizadas para alimentar el sistema CanSat son del tipo Zippy Compact 850mAh Li-PO con 2 celdas de 7.4V. La Figura 9 muestra el diseño del PCB que esta por integrarse en una etapa futura. La intención es

minimizar ruido en la interconexión entre el microcontrolador y módulos de telemetría, además de reducir tamaño y peso.

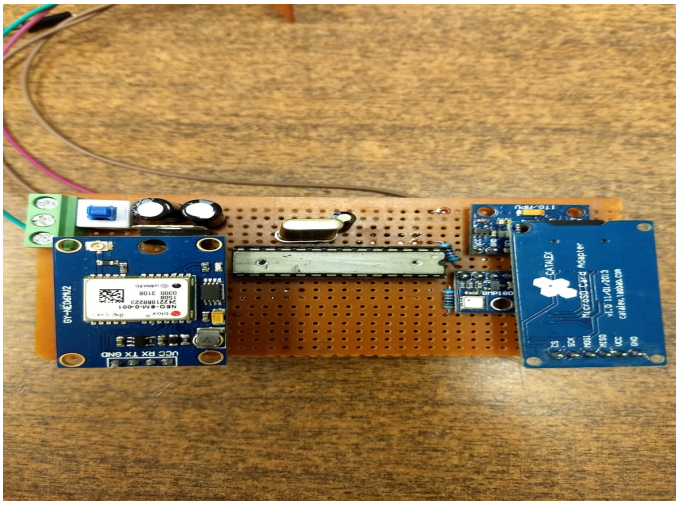


Fig. 8. Prototipo CanSat usado para pruebas.

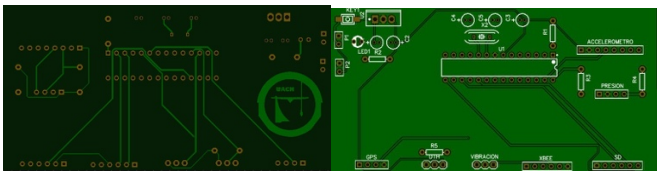


Fig. 9. Diagrama del PCB para el prototipo CanSat.

5 Diseño del paracaídas

El paracaídas es otro elemento importante dentro del sistema en virtud. El paracaídas permite controlar la velocidad de descenso permitiendo atender las restricciones establecidas en el concurso (5-9 Km/s), además de reducir el riesgo de daños para el CanSat a la hora del impacto con el suelo. El diseño del paracaídas considera tres aspectos físicos relevantes: caída libre, resistencia, y velocidad de descenso, como se muestra en la Figura 10.

Donde:

F_r [=] Resistencia
m [=] masa
g [=] gravedad (9.81 m/s^2)
X [=] altura

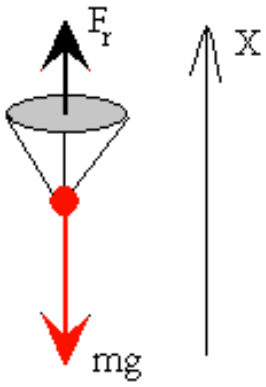


Fig. 10. Sistemas de fuerzas que actúan en un paracaídas.

Considerando el máximo peso permitido para el CanSat (1.05 kg), y una velocidad de descenso de 6 m/s, se puede calcular la resistencia/arrastre del paracaídas F_r a partir de (1),

$$Fr = \frac{mg}{V^2} \tag{1}$$

donde F_r representa el arrastre generado por el paracaídas para contrarrestar la fuerza de gravedad.

El siguiente paso es calcular las dimensiones que deberá guardar el paracaídas para lograr cumplir con la fuerza F_r, esto se determina mediante (2),

$$K = \frac{rAd}{2} \tag{2}$$

donde r es la densidad del aire; A es el área del paracaídas, y d es el coeficiente de arrastre. El coeficiente de arrastre depende directamente de la aerodinámica de un paracaídas y está determinada por la relación entre la fuerza de arrastre y la presión dinámica, esta última en función de la velocidad, la densidad del aire, la altitud y la temperatura. Si se considera un diseño circular del paracaidas, tenemos un valor aproximado para el coeficiente de arrastre de 1.2, y si se asume, por ejemplo, un valor de densidad del aire, r, a nivel del mar (1.29 kg/m^3), se puede calcular el diámetro del paracaídas a través de (3).

$$D^2 = \frac{8k}{Pd\pi} = \sqrt{\frac{8k}{Pd\pi}} \tag{3}$$

A partir del diámetro calculado para el paracaídas obtenemos su perímetro y utilizamos éste para obtener el número de gajos que compondrán al mismo paracaídas. Si se tiene un perímetro de 2.2m, se puede diseñar el paracaídas con 10 triángulos de 22cm de base por 55cm de altura.

6 Diseño de la estructura en impresión

Considerando la telemetría, se diseñó una plataforma con las medidas del PCB y una parte trasera para colocar las baterías que alimentan el cansat, también se diseño analizó e imprimió la estructura general donde estarían soportados todos los subsitemas involucrados, como se observa en la figura 11.

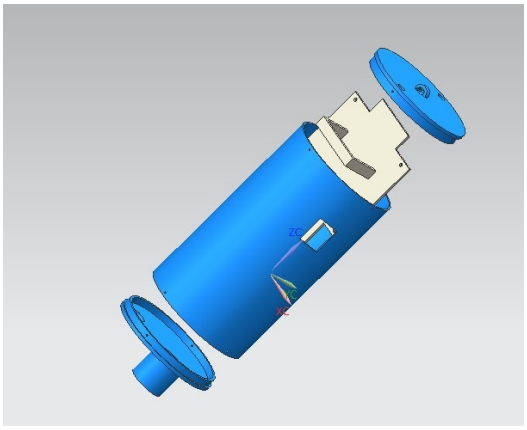


Fig. 11. Estructura diseñada para albergar el CanSat.

En la parte inferior del cansat, se trabajó con una base con salidas para las antenas, agregando un espacio para la cámara junto con una estructura diseñada para protegerla del impacto en el descenso, como puede observarse en las figuras 12 y 13, respectivamente.

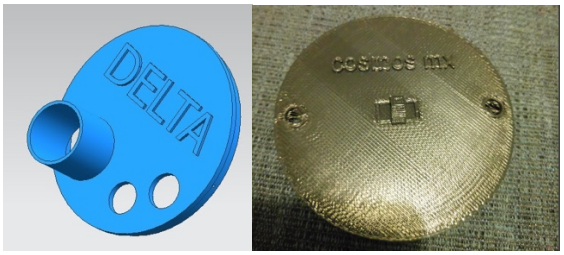


Fig. 12. Diseño y acabado final de las cuiertas superior e inferior de la estructura del CanSat.

Fig. 13. El CanSat en la estructura final diseñada.

7 Conclusiones

En este documento se presenta el diseño de un sistema CanSat el cual está basado en el microcontrolador ATMEGA.

Se diseño el PCB para realizar el acomodo las componentes, considerando el diesño de pistas para lograr la mejor distribución y eficiencia, tales como la implementación del protocolo de comunicación i^2c y la distribución de las tierras comunes totalmente en la capa baja de la tablilla, para minimizar el ruido.

La visualización se desarrolló utilizando MEAN Stack, que consiste en la conjugación de una serie de elementos entorno a Node.js, que van desde un sistema gestor de Bases de Datos (MongoDB), hasta frameworks de apoyo (Express y Angular). Logrando la visualización en tiempo real de los datos sensados y creando una base de datos

Dentro de los éxitos logrados se tiene el aprendizaje sobre el uso de diferentes módulos para telemetría, tomar decisiones, trabajar en un equipo multidisciplinario, estudiar diferentes tipos de estructuras y materiales y diseño electrónico y mecánico.

A la fecha de la elaboración de este documento no se tienen pruebas de campo ni un estudio sobre los datos medidos para llegar a una conclusión objetiva. Sin embargo, la fase en la que se encuentra el prototipo nos alienta a seguir para cumplir con los requerimientos marcados en la convocatoria del concurso.

Referencias

1. Curtis, HD (2013). Mecánica orbital para estudiantes de ingeniería . Butterworth-Heinemann.
2. **6º Programa de capacitación para líderes de CanSat**(2015) Disponible en: http://cltp.info/cltp6.html Accedido en Agosto 2018
3. .S. Yamaura, H. Akiyama and R. Kawashima, "Report of CanSat Leader Training Program," *Proceedings of 5th International Conference on Recent Advances in Space Technologies - RAST2011*, Istanbal, 2011.
4. M. E. Aydemir, R. C. Dursun and M. Pehlevan, "Ground station design procedures for CANSAT," *2013 6th International Conference on Recent Advances in Space Technologies (RAST)*, Istanbul, 2013.

Desarrollos Científicos y Tecnológicos en las Ciencias Computacionales
se terminó de editar en Diciembre de 2020 en la
Facultad de Ciencias de la Computación
Av. San Claudio y 14 Sur Jardines de San Manuel
Ciudad Universitaria
C.P. 72570

Desarrollos Científicos y Tecnológicos en las Ciencias Computacionales
Coordinado por Gustavo Trinidad Rubín Linares