

Lenguaje, conocimiento y tecnología educativa: avances recientes

Mireya Tovar Vidal
Claudia Zepeda Cortés
Darnes Vilariño Ayala
Juan Manuel González Calleros
Josefina Guerrero García

Editores



Lenguaje, conocimiento y
tecnología educativa: avances
recientes

Lenguaje, conocimiento y tecnología educativa: avances recientes

Mireya Tovar Vidal
Claudia Zepeda Cortés
Darnes Vilariño Ayala
Juan Manuel González Calleros
Josefina Guerrero García
Coordinadores



Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
2021

Primera Edición **2021**
ISBN BUAP: 978-607-525-761-7

DR © Benemérita Universidad Autónoma de Puebla
4 Sur 104, Col. Centro Histórico, Puebla, Pue. CP 72000
Teléfono: 01 (222) 229 55 00
www.buap.mx

Dirección General de Publicaciones
2 norte 1404, Col. Centro Histórico, Puebla, Pue. CP. 72000
Teléfono: 01 (222) 246 85 59 y 01 (222) 55 00 Ext. 5768
publicaciones.buap.mx

Facultad de Ciencias de la Computación
14 sur esq. Con Av. San Claudio
Ciudad Universitaria, Puebla, Pue.
Telfonos: 01 (222) 229 55 00 Ext. 7200 y 7204
www.cs.buap.mx

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA • *Rector:* José Alfonso Esparza Ortíz • *Secretario General:* Guadalupe Grajales Y Porras • *Vicerrector de Extensión y Difusión de la Cultura:* José Carlos Bernal Suárez • *Director General de Publicaciones:* Hugo Vargas Comsille • *Directora de la Facultad de Ciencias de la Computación:* María del Consuelo Molina García

Hecho en México
Made in Mexico

Prólogo

El presente libro surgió como una iniciativa para establecer un estudio sobre los distintos avances en investigación que se llevan a cabo en la Facultad de Ciencias de la Computación de la Benémerita Universidad Autónoma de Puebla, que permitan al lector conocer sobre los distintos tópicos de investigación que se desarrollan en la misma. Por lo tanto, en este libro titulado “Lenguaje, conocimiento y tecnología educativa: avances recientes” se presentan diferentes tópicos de investigación en el área de la Ingeniería del Lenguaje y del Conocimiento en las Ciencias de la Computación durante el primer semestre del 2021.

La obra incluye once capítulos de investigación, seleccionados de un total de veintidós, en las áreas de reconocimiento de patrones, visión computacional, redes neuronales, coloreo de árboles, procesamiento de lenguaje natural, inteligencia artificial aplicada a la educación, entre otras.

Los capítulos que forman parte de esta obra fueron revisados por el sistema de doble par ciego y aprobados para su publicación por expertos en el área, lo que permitió asegurar su calidad científica en las áreas de estudio. A continuación se menciona la aportación de cada uno de ellos.

En el Capítulo 1 se presenta un enfoque no supervisado para identificar y extraer relaciones entre dos entidades nombradas. El enfoque se conforma por casos, estableciendo un conjunto de patrones para identificar relaciones predefinidas. El proceso se realiza de forma automática sobre documentos no estructurados en el dominio de noticias políticas en idioma español. En el Capítulo 2 se describe una propuesta para el problema del reconocimiento automático de lengua de señas, al mismo tiempo se identifican las áreas de oportunidad, la metodología propuesta y algunos resultados iniciales. En el Capítulo 3 se muestra una revisión de las principales técnicas existentes para la detección automática de la Retinopatía Diabética en imágenes de la retina utilizando diferentes técnicas de análisis de imágenes, visión computacional e inteligencia artificial. En el Capítulo 4 se propone una metodología general que consta de la primera versión de un algoritmo iterativo que utiliza una red neuronal artificial que estima los parámetros dinámicos de un robot. En el Capítulo 5 se describe la aplicación de diferentes herramientas para clasificar una muestra de textos extraídos del corpus OneStopEnglish conforme al Marco Común de Referencia Europeo y aplica tres métricas populares de lecturabilidad. Se concluye que la granularidad de la información extraída es un referente para recomendar textos a los estudiantes. En el Capítulo 6 se propone una metodología de diseño para obtener una plataforma experimental capaz de asistir en la creación de redes neuronales para su simulación e implementación en un sistema embebido, con el fin de permitir reducir el tiempo y esfuerzo necesario para desarrollar redes neuronales. En el Capítulo 7 se presenta un método para modelar el 3 coloreo de grafos representados por árboles poligonales, la propuesta se basa en especificar lógicamente las restricciones generadas para que se dé un 3 coloreo válido sobre

grafos poligonales. Para que el proceso total sea de tiempo lineal, las restricciones lógicas son formadas de manera dinámica. En el Capítulo 8 se describe el desarrollo de un sistema para medir el nivel de atención de un estudiante en su entorno escolar en tiempo real y de forma cuantitativa. Se relaciona el porcentaje de atención que se obtiene de las ondas cerebrales, con los ángulos obtenidos de la postura corporal a través de un análisis matemático cuyo fin es encontrar un modelo de correlación entre estas dos variables. Como resultado se obtienen dos modelos matemáticos: el primero para una población de estudiantes con el Trastorno de Déficit de Atención e Hiperactividad (TDAH) y el segundo para una población de estudiantes sin TDAH. Por otro lado, en el Capítulo 9 se muestra el proceso de optimización del consumo eléctrico de una casa habitación implementando la técnica metaheurística de cúmulo de partículas. En el Capítulo 10, los autores describen una revisión metodológica del proceso para la creación automatizada de redes complejas a partir de secuencias de videovigilancia, con el fin de detectar anomalías en video empleando herramientas del procesamiento de texto. Por último, en el Capítulo 11 se presenta una metodología, basada en visión estéreo, para la obtención de mallas de superficies de calles y el coloreo de curvaturas para visualizar sus irregularidades.

Por último, queremos expresar nuestro agradecimiento a cada uno de los autores por su aportación, a nuestros revisores por su valiosa labor, a la Facultad de Ciencias de la Computación de la Benemérita Universidad Autónoma de Puebla y a todos aquellos cuya participación contribuyó para la publicación de esta obra.

Los editores,
Mireya Tovar Vidal
Claudia Zepeda Cortés
Darnes Vilariño Ayala
Juan Manuel González Calleros
Josefina Guerrero García

Índice general

Prólogo	IV
Capítulo 1. Extracción de relaciones entre entidades usando árboles de dependencia	1
<i>Orlando Ramos-Flores, David Pinto</i>	
Capítulo 2. Metodología y avances para el Reconocimiento Continuo de Lengua de Señas	11
<i>Daniel Sánchez-Ruiz, J. Arturo Olvera-López, Ivan Olmos Pineda</i>	
Capítulo 3. Análisis de técnicas para la detección y clasificación de la retinopatía diabética mediante imágenes digitales	21
<i>David Ferreira Piñeiro, Iván Olmos Pineda, Arturo Olvera López</i>	
Capítulo 4. Red Neuronal Artificial para Extraer Parámetros Dinámicos de Robots a Partir de Información Incompleta de su Movimiento	31
<i>Carlos Leopoldo Carreón Díaz de León, Sergio Vergara Limon, Juan Manuel González Calleros, María Aurora Diozcora Vargas Treviño</i>	
Capítulo 5. Construyendo el contenido de un Sistema Tutor Inteligente: un estudio piloto de métricas populares de lecturabilidad para enseñanza de idiomas	41
<i>Adelina Escobar Acevedo, Josefina Guerrero García</i>	
Capítulo 6. Generación de una plataforma que asista en la creación de redes neuronales para su simulación e implementación en un sistema embebido	51
<i>Daniel Marcelo González Arriaga, María Aurora Diozcora Vargas Treviño, Sergio Vergara Limon, Josefina Guerrero García</i>	
Capítulo 7. Modelado del 3 coloreo de árboles poligonales vía Satisfactibilidad Incremental	61
<i>Cristina Lopez, Guillermo De Ita</i>	
Capítulo 8. Influencia de la postura corporal sobre la atención	72
<i>Alfredo Garcia, Juan Manuel Gonzalez, Amparo Palomino</i>	
Capítulo 9. Optimización del Consumo Eléctrico mediante la Heurística Cúmulo de Partículas	82
<i>Blanca Nydia Perez Camacho, Juan Manuel Gonzalez Calleros, Gustavo Rodriguez Gomez</i>	

Capítulo 10. Creación automática de redes complejas a partir de secuencias de videovigilancia	92
<i>Diana Karina Guevara Flores, David Eduardo Pinto Avendaño, José Antonio Méndez Bermúdez</i>	
Capítulo 11. Análisis de Elevaciones en Superficies Viales mediante Visión Estéreo	98
<i>Ana L. Ballinas-Hernández, Ivan Olmos-Pineda y J. Arturo Olvera-López</i>	
Índice de autores	108
Compiladores	109
Revisores	110
Editores	111

Capítulo 1

Extracción de relaciones entre entidades usando árboles de dependencia

Orlando Ramos-Flores, David Pinto

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
orlandxrf@gmail.com, dpinto@cs.buap.mx

Resumen La extracción de relaciones es una tarea del Procesamiento de Lenguaje Natural que tiene como objetivo obtener una comprensión de textos no estructurados. En este trabajo se presenta un enfoque no supervisado para identificar y extraer relaciones entre dos entidades nombradas. El enfoque se conforma por casos, estableciendo un conjunto de patrones para identificar relaciones predefinidas. Para identificar la relación contenida de forma intrínseca en la oración para ambas entidades, se emplean dependencias universales como *appos* y *amod*, así como identificar elementos clave de la oración como son el *verbo* entre dos entidades nombradas, y el *sujeto* y *objeto* de la oración. Este proceso se realiza de forma automática sobre documentos no estructurados en el dominio de noticias políticas en idioma español.

Palabras Clave: Extracción de Relaciones, Árboles de Dependencia, Patrones

1. Introducción

La Extracción de Relaciones (ER) es una subtarea de la Extracción de la Información (EI) y consiste de identificar, extraer y estructurar los datos contenidos en documentos de texto. La EI generalmente se divide en tres subproblemas: Resolución de Correferencia, Reconocimiento de Entidades Nombradas y Extracción de Relaciones (Bunescu, 2005). En este trabajo se presenta un enfoque no supervisado para identificar y extraer relaciones entre dos entidades nombradas, las entidades nombradas han sido previamente definidas, como se describe en la Tabla 1. La ER es una tarea que no se ha resuelto del todo, aunque se han abordado diferentes métodos no supervisados principalmente empleando agrupamiento (Hasegawa-2004, Yan-2009), usando modelos probabilísticos sobre Extracción de Información Abierta (EIA) (Banko-2007) y recientemente sobre la misma línea (EIA) empleando frases verbales y basadas en cláusulas (Vo, 2018).

El aporte de este trabajo se centra en dos puntos, el primero consiste en un método con diferentes casos para extraer relaciones en idioma español empleando árboles de dependencia universal. El segundo contempla la extracción de relaciones en dos entidades nombradas sin conocer previamente la relación,

es decir, no se presupone un conjunto de relaciones, la relación se obtiene intrínsecamente de la oración, y es formada de una o más palabras y no necesariamente se tienen que encontrar entre ambas entidades. Además, se asume que la información de la relación en cada oración es independiente de las demás.

La organización se describe a continuación: en la Sección 2 se describen trabajos previos abordando esta tarea con árboles de dependencia. El conjunto de datos es descrito en la Sección 3. En la Sección 4 se presentan el método propuesto sobre cada caso para la identificación y extracción de relaciones, en la Sección 5 los resultados obtenidos se describen y en la Sección 6 se presentan las conclusiones.

2. Trabajos relacionados

En el trabajo de Bunescu y Mooney (2005) proponen un método para extraer entidades entre dos entidades nombradas, usando el análisis de dependencias al generar árboles de dependencia. Proponen obtener la ruta más corta entre ambas entidades, y ocurre cuando ambas entidades están conectadas al predicado. Emplean el corpus ACE (Automated Context Extraction) con relaciones y entidades anotadas, en sus experimentos emplearon un clasificador (SVM) Support Vector Machine obteniendo un F-score de 52.5 %.

El uso de árboles de análisis de dependencias es usado por Fundel et al (2007), en el cual usan un conjunto pequeño de reglas, un diccionario de sinónimos que contienen nombres de proteínas y genes, y una lista de términos de restricción que se utilizan para describir relaciones de interés. Las relaciones se crean extrayendo rutas que conectan pares de proteínas de árboles de análisis de dependencia. Se identifican y extraen tres tipos de relaciones. Previo a la evaluación realizaron una evaluación manual. Obtienen un F-score de 78 % en la evaluación.

Frases de análisis de dependencia son definidas en el trabajo de Wu et al (2009). Estas frases son compuestas por características de productos y expresiones de opinión. Extraen la relación que existe entre las mismas a través de un subárbol (frases de análisis de dependencia) y debe tener una distancia en el árbol menor a cinco hacia la relación. El corpus fue anotado de forma manual y usaron un clasificador SVM, con un F-score de 46.3 %.

Un enfoque no supervisado para extraer relaciones usando el análisis de dependencias es presentado por Afzal et al (2011). En primera instancia reconocen entidades nombradas con GENIA tagger, después reemplazan el texto de la entidad por su clase dentro de cada oración y aplican el análisis de dependencia sobre esta nueva oración. Del árbol generado obtienen los patrones candidatos, para luego jerarquizarlos en base a su relevancia en el corpus usando diferentes métricas para ello. En sus evaluaciones de las métricas usando Chi-Square y Normalised Mutual Information presentan los mejores resultados en base a su precisión en los patrones de árboles de dependencia.

Otro enfoque no supervisado para la extracción de relaciones es el descrito por Quan y Ren (2014). Usan agrupaciones de patrones basada en un kernel polinomial (vectores de características) para identificar la interacción de

palabras, esas palabras se combinan con el análisis sintáctico de la estructura de la frase y el análisis de la dependencia para la extracción de relaciones. Utilizan un clasificador KNN, en su evaluación usan el conjunto de datos AImed que es un benchmark para interacciones entre dos proteínas, además se comparan con otros trabajos del estado del arte. En sus resultados alcanzan un F-score de 82.05 % sobre el primer conjunto de datos y un F-score de 47.20 % en el segundo conjunto de datos.

En el trabajo de Miwa y Bansal (2016) proponen un modelo con arquitectura arquitectura LSTM-RNN (Li et al, 2015) que extrae relaciones entre entidades en secuencias de palabras y estructuras de árboles de dependencia. Su modelo es capaz de detectar entidades nombradas y relaciones. Hacen uso de la estructura con la ruta más corta, un subárbol y el árbol completo, además de un conjunto de características adicionales. En sus experimentos usan el corpus ACE 2004 y ACE 2005, obteniendo un F-score de 48.4 % y de 55.6 % respectivamente.

3. Conjunto de datos

El conjunto de datos utilizado consta de 32,147 documentos de noticias políticas. El conjunto de datos contiene quince clases de entidades nombradas como se describe en la Tabla 1. En la Tabla 1 la columna *FU* representa al número de entidades únicas y la columna *FT* enumera el total de entidades nombradas por clase.

Tabla 1. Entidades reconocidas en el corpus de noticias.

No.	Clase	Entidad	FU	FT
1	PER	Persona	15,132	110,455
2	ORG	Organización	8,485	75,066
3	TIT	Puesto trabajo	7,903	78,801
4	GPE	Geopolítica	2,152	26,707
5	MNY	Moneda	1,810	3,650
6	DAT	Fecha	1,785	13,027
7	FAC	Instalación	1,499	6,591
8	DOC	Documento	692	2,360
9	EVT	Evento	424	2,555
10	PRO	Producto	407	1,490
11	PEX	Partido Político	344	20,260
12	TIM	Tiempo	324	5,147
13	AGE	Edad	129	1,489
14	DEM	Gentilicio	99	3,793
15	LOC	Lugar	27	117

Para cada uno de los documentos se identificaron sus oraciones y se filtraron aquellas con al menos dos entidades nombradas. Cuando existen más de dos entidades se crea una ventana para visualizar únicamente dos entidades por

oración, una oración s con tres entidades e_1, e_2, e_3 se convierte en dos oraciones: la primera oración s_1 con las entidades e_1 y e_2 y la segunda oración s_2 con las entidades e_2 y e_3 .

4. Extracción de relaciones

Un enfoque no supervisado se usó para abordar esta tarea usando el análisis de dependencias para las oraciones. En primera instancia, se realizó un estudio sobre el análisis de dependencias en oraciones generado sus respectivos árboles, para ello se empleó el modelo en español de la biblioteca de Python Spacy¹ para el análisis, y las bibliotecas Networkx² y Graphviz³ para su visualización. Después de realizar el análisis se definieron los casos para obtener relaciones entre dos entidades nombradas.

4.1. Casos analizados

Los casos que se proponen consisten en identificar patrones para obtener un conjunto de relaciones predefinidas, casos basados en las relaciones de dependencia universal que ligan a dos entidades, y aquellos casos que toman elementos clave de la oración como son el *sujeto* y *objeto*.

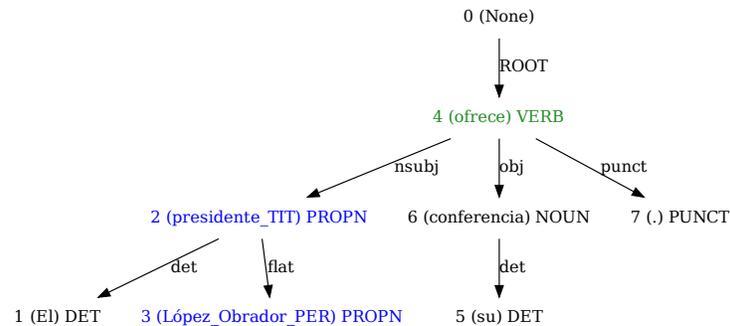


Figura 1. Árbol de dependencias generado

La oración "El presidente_TIT López_Obrador_PER ofrece su conferencia ." genera un árbol de dependencias como se muestra en la Figura 1, donde los tokens

¹ <https://spacy.io/models/es>

² <https://networkx.org/documentation/stable/>

³ <https://graphviz.readthedocs.io/en/stable/>

(palabra o símbolo) representa a los nodos y las aristas (líneas con dirección) son las relaciones de dependencia universal. Cada nodo contiene id (*token*) POS . El id es también el orden en que aparece el token en la oración, y POS es la categoría gramatical (Part-of-Speech) del token.

El proceso consiste en transformar la oración obtenida del modelo de análisis de dependencias a un grafo, y se define como $G = (V, E, \theta)$ donde V son los vértices o nodos, E son las aristas y $\theta : E \rightarrow \{\{x, d, y\} : x, y \in V \wedge d \in D = \{\textit{dependencias universales}\}\}$, es decir, se asigna un arista d a un par de nodos. Los vértices x e y contienen la estructura $(id, POS, token)$. El conjunto de “*dependencias universales*” se define como $D = \{ROOT, acl, advcl, advmod, amod, appos, aux, case, cc, ccomp, compound, conj, cop, csubj, dep, det, expl:pass, fixed, flat, iobj, mark, nmod, nsubj, nummod, obj, obl, parataxis, punct, xcomp\}$. Y al conjunto que contiene a las etiquetas POS como $S = \{ADJ, ADP, ADV, AUX, CONJ, DET, INTJ, NOUN, NUM, PART, PRON, PROPN, PUNCT, SCONJ, SYM, VERB\}$, además se establece al conjunto $O = \{\textit{Tokens de la oración}\}$. Estos conjuntos son empleados en el grafo, el conjunto O es el único que cambia dependiendo la oración que se este procesando. En cada uno de los métodos propuestos como paso inicial se busca un camino simple entre la primer entidad (e_1) y la segunda entidad (e_2), donde $e_1 \in V$ y $e_2 \in V$, exceptuando los métodos donde se identifica al *sujeto* y *objeto* de la oración.

Relaciones basadas en patrones En este método se definieron cuatro tipos de relaciones de forma manual. La primera relación que se identifica es del tipo “*puestos de trabajo*”. Para definir los patrones se usan las clases de las entidades nombradas: ORG, PER, PEX y TIT. Los patrones se definen en pares usando la clase de ambas entidades (e_1, e_2) , el conjunto de patrones definido es $P = \{(ORG, PER), (ORG, TIT), (PER, ORG), (PER, PEX), (PER, TIT), (PEX, PER), (PEX, TIT), (TIT, ORG), (TIT, PER), (TIT, PEX)\}$. Para representar la relación de dependencia de dos nodos se establece como $nodo_1 \xrightarrow{r} nodo_2 \mid r \in D$. Las condiciones que se deben cumplir para identificar este tipo de relaciones se establece como: $R_1 = \{e_1 \xrightarrow{d} e_2 \wedge (e_2 \xrightarrow{punct} x \vee e_2 \xrightarrow{none} \emptyset) \mid d \in D, (e_1, e_2) \in P\}$. De forma gráfica la Figura 2 en (1) ilustra como e_2 debe ser descendiente directo de e_1 (no deben existir nodos entre ellas). Además se debe cumplir que el nodo $x = (id, PUNCT, t) \mid t \in T, T = \{“,”, “;”, “(”, “)”\}$. Además e_2 puede tener una dependencia *punct* hacia un nodo con una etiqueta POS de tipo *PUNCT*, o bien e_2 puede no tener descendientes.

La segunda relación a identificar es de tipo “*tiene el acrónimo de*” se redefine el conjunto P . El conjunto de patrones $P = \{(PER, PER), (ORG, ORG), (TIT, TIT), (GPE, GPE), (PEX, PEX), (FAC, FAC), (EVT, EVT), (DOC, DOC), (PRO, PRO), (LOC, LOC)\}$. Para obtener este tipo de relaciones se debe cumplir que $R_2 = \{e_1 \xrightarrow{d} e_2 \wedge (e_2 \xrightarrow{punct} s_a \wedge e_2 \xrightarrow{punct} s_b) \mid (e_1, e_2) \in P\}$. Donde el vértice $s_a = (id, PUNCT, “(”)$ y el vértice $s_b = (id, PUNCT, “)”)$. En la Figura 2 en (2) se observa que e_2 sea descendiente directo de e_1 , y que e_2 debe tener dos descendientes específicos (paréntesis de apertura y cierre).

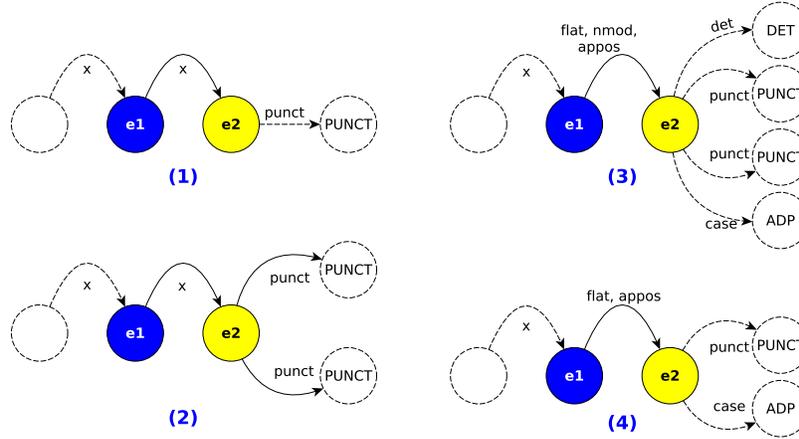


Figura 2. Árboles de dependencia para identificar relaciones predefinidas.

La tercera relación es del tipo *“que pertenece a”*, el conjunto de patrones se redefine como $P = \{(FAC, GPE), (ORG, GPE)\}$, el conjunto $D_a = \{flat, nmod, appos\}$, $D_b = \{case, det\}$, $S_a = \{ADP, DET\}$. Dados los conjuntos se tiene que $R_3 = \{e_1 \xrightarrow{r_a} e_2 \wedge ((e_2 \xrightarrow{punct} s_a \wedge e_2 \xrightarrow{punct} s_b) \vee e_2 \xrightarrow{r_b} s) \mid r_a \in D_a, r_b \in D_b, (e_1, e_2) \in P\}$. Donde el vértice $s_a = (id, PUNCT, “(”)$ y el vértice $s_b = (id, PUNCT, “)”)$, y se tiene que $s = (id, ADP, token) \vee (id, DET, token)$. En la Figura 2 en (3) se puede observar como e_1 debe ser ancestro directo de e_2 , a su vez e_2 puede tener descendientes con etiqueta PUNCT (signo de puntuación) paréntesis de apertura y cierre o tener descendientes con etiquetas ADP (Adposición) y DET (Determinante).

La cuarta relación definida de forma manual en base a sus patrones es del tipo *“representado por”*. El conjunto de patrones se redefine como $P = \{(GPE, PER)\}$ y $D_a = \{flat, appos\}$. Para identificar esta regla se debe cumplir que $R_4 = \{e_1 \xrightarrow{r_a} e_2 \wedge e_2 \xrightarrow{punct} s \mid r_a \in D_a, (e_1, e_2) \in P\}$, donde el vértice $s = (id, PUNCT, “,”)$. La Figura 2 en (4) representa las condiciones para identificar esta relación. e_1 debe ser ancestro de e_2 y tener alguna de las dependencias *flat* o *appos*. Así como e_2 debe tener una dependencia de tipo *punct* hacia un nodo con un token “,”. Cuando se cumplen las condiciones en R_1, R_2, R_3 y R_4 se define la tripleta $\langle e_1, relación\ predefinida, e_2 \rangle$ resultante y se almacena.

Relaciones intrínsecas en la oración Además de usar patrones de clases de entidades para identificar relaciones, se emplea las dependencias universales entre tokens (nodos), así como localizar el *verbo* entre ambas entidades, e identificar elementos clave como *sujeto*, y *objeto* de la oración enlazados al *predicado* (*verbo*). Todos los nodos tienen la estructura $(id, POS, token)$ aunque solo se

refiera a ellos por su etiqueta POS. En la Figura 3 se ilustran los métodos en su forma básica para identificar y extraer relaciones entre entidades nombradas.

Para identificar relaciones entre entidades usando la dependencia “*appos*” (modificador de sustantivo) como se observa en la Figura 3 en (5), se debe cumplir en su forma más básica que $R_5 = \{e_1 \xrightarrow{d} NOUN \wedge NOUN \xrightarrow{appos} e_2 \mid d \in D\}$, esto indica que entre los vértices e_1 y e_2 exista un nodo *NOUN*, y que contenga una dependencia *appos* hacia el nodo e_2 . Además de forma opcional, cuando existan más vértices entre los nodos mencionados, se deben cumplir las dependencias descritas en la Figura 3 en (5). Por ejemplo se se tiene $R_{ex} = \{(e_1 \xrightarrow{d} ADV) \wedge (ADV \xrightarrow{nm\text{od}} NOUN) \wedge (NOUN \xrightarrow{appos} X) \wedge (X \xrightarrow{d} e_2) \mid \exists d \in D\}$ donde *ADV* y *NOUN* son etiquetas POS específicas de nodos, y *X* es la etiqueta POS de cualesquiera vértices, así se establecen las condiciones que se deben cumplir para identificar la relación R_{ex} , y los nodos almacenados solo serán aquellos que sean específicamente definidos, es decir el nodo con etiqueta POS *X* es descartado. Para extraer la relación se obtiene el conjunto de los vértices obtenidos, del ejemplo anterior se tiene $R_{ex} = \{(id_1, ADV, token_1), (id_2, NOUN, token_2)\}$ y de cada vértice en el conjunto se obtiene una lista $r = [token_1, token_2]$ de *tokens* ordenados en forma ascendente en base a su *id*. Esta lista es la relación identificada por R_{ex} , finalmente se crea la tripleta correspondiente $\langle e_1, r, e_2 \rangle$ y se almacena en una base de datos.

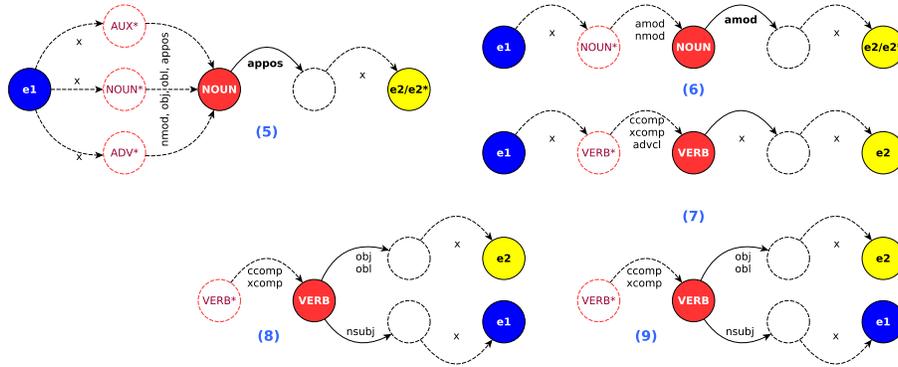


Figura 3. Árboles de dependencia para identificar relaciones no específicas.

El método para identificar relaciones usando la dependencia universal “*amod*” (modificador de adjetivo) se define como $R_6 = \{(e_1 \xrightarrow{d} NOUN) \wedge (NOUN \xrightarrow{amod} e_2) \mid d \in D\}$ sin embargo puede ocurrir que existan una serie de combinaciones entre otros vértices como se observa en la Figura 3 en (6). De igual manera, de los vértices obtenidos se genera una lista de tokens ordenados de los vértices, se construye la tripleta y se almacena.

Otro método presentado es usando el “*verbo entre ambas entidades*” para identificar relaciones, y se define como: $R_7 = \{(e_1 \xrightarrow{d} VERB) \wedge (VERB \xrightarrow{d} e_2) \mid d \in D\}$. Cuando la oración presenta más de un vértice *VERB* entre ambas entidades, se selecciona el primer vértice siguiendo la ruta de $e_2 \rightarrow e_1$. En caso de existir un vértice *VERB* como ancestro del vértice *VERB* seleccionado previamente, se debe cumplir que ambos vértices estén ligados por alguna de las dependencias mostradas en la Figura 3 en (7). El proceso de extracción de la relación es similar al anterior, se extraen y se ordenan los tokens, se crea la tripleta y se almacena.

Por último se definieron dos métodos más para identificar relaciones entre entidades. Consiste en identificar al vértice principal (*predicado*) que es un nodo *VERB*, de este nodo se identifican las dependencias *sujeto* y *objeto* que se vinculan hacia otros nodos. A partir de aquí se definieron dos métodos, el primero cuando e_1 y e_2 se encuentran vinculados hacia el nodo *predicado* por medio de una dependencia *sujeto* y *objeto* respectivamente (Figura 3 en (8)), y viceversa (Figura 3 en (9)). La definición cuando e_1 esta vinculada por medio del *sujeto* y e_2 del *objeto* es: $R_8 = \{VERB \xrightarrow{nsbj} e_1 \wedge VERB \xrightarrow{obj \vee obl} e_2\}$. De manera similar cuando e_2 proviene de una dependencia *sujeto* y e_1 de una *objeto*: $R_9 = \{VERB \xrightarrow{nsbj} e_2 \wedge VERB \xrightarrow{obj \vee obl} e_1\}$. La extracción de las relaciones tiene el mismo proceso, se extraen los nodos y de ellos se obtiene una lista ordenada de los tokens, para finalmente crear la tripleta correspondiente y se almacenarla.

5. Resultados

Las tripletas obtenidas por cada uno de los nueve casos presentados se almacenaron en una base de datos. La información de cada tripleta almacenada contiene un id, el conjunto de ids de las oraciones donde se extrajo la tripleta, las partes de la tripleta por separado (e_1 , *relación*, e_2), el caso usado y la evaluación individual de cada uno de los componentes de la tripleta.

La evaluación consistió en revisar al menos 5 oraciones por cada tripleta, evaluando de forma *positiva* o *negativa* a cada elemento de la tripleta: e_1 , *relación* y e_2 , y una segunda evaluación sobre la *relación* de la tripleta sin importar las entidades. La evaluación fue realizada por una persona y los resultados se muestran en la Tabla 2.

En la Tabla 2 la columna *Total* enumera la cantidad neta de relaciones extraídas por método. La columna *Eval* (Evaluadas) describe la cantidad original de tripletas obtenidas para la evaluación, siendo las primeras 200 tripletas por cada método, con excepción del método cuatro donde solo se obtuvieron 33 tripletas y solo se usaron 32 (se descarto una tripleta) en la evaluación. La columna *CR* (Conteo de Relaciones) hace referencia al número de relaciones de la tripleta evaluadas de forma correcta. En el mismo sentido la columna *CT* (Conteo Tripletas) enumera la cantidad de tripletas evaluadas correctamente, esto incluye una evaluación positiva de ambas entidades y la relación. Las columnas *PR* (Porcentaje Relación) y *PT* (Porcentaje Tripletas) describen los porcentajes de evaluación para cada uno de los métodos.

Tabla 2. Evaluación manual de las tripletas extraídas.

No.	Método	Total	Eval	CR	CT	PR	PT
1	Puestos de trabajo	17,073	200	193	193	96.5 %	96.5 %
2	Acrónimo	2,367	200	199	192	99.5 %	96.0 %
3	Localizado	343	200	184	172	92.0 %	86.0 %
4	Representado por	33	32	28	25	87.5 %	78.12 %
5	Dependencia appos	1,408	200	156	132	78.0 %	66.0 %
6	Dependencia amod	1,922	200	161	131	80.5 %	65.5 %
7	Verbo entre entidades	7,896	200	124	102	62.0 %	51.0 %
8	Verbo ancestro E1	44,479	200	126	111	63.0 %	55.5 %
9	Verbo ancestro E2	11,396	200	109	93	54.5 %	46.5 %
		86,917	1,632	1,280	1,151	78.43 %	70.52 %

Como se observa en la Tabla 2 las relaciones mejores evaluadas son las de *puestos de trabajo* y *acrónimo* esto se debe en gran medida a la alta frecuencia de oraciones con clases de entidades sobre *personas*, *título de trabajo* y *organizaciones*. La tripleta más frecuente en relaciones de *puestos de trabajo* es *<presidente, es el título de, Andrés Manuel López Obrador>* en cuanto a la relación *acrónimo* la más frecuente fue *<Instituto Nacional Electoral, tiene el acrónimo de, INE>*.

En cuanto a las relaciones obtenidas de forma intrínseca de la oración entre las frecuentes del caso *appos* se tienen *<Gobernadora de Puebla, y su esposo el, senador>* y *<Olga Sánchez Cordero, exministra y futura, secretaria de Gobernación>*, como se observa se identificaron relaciones que conectan el puesto de trabajo de dos entidades, así como relaciones familiares. En el caso *amod* se identificaron y extrajeron tripletas como *<conferencia de prensa, realizada en, Palacio Nacional>*, *<Pedro Carrizales, conocido como, El Mijis>* y *<Policía Federal, creada hace, 20 años>* este caso fue útil para identificar relaciones con frases verbales entre entidades nombradas al igual que los casos 8 y 9, obteniendo tripletas como *<AMLO, se reúne con, Meade>* y *<senador, asistirá en representación del, Japón>*.

Sin embargo cabe señalar que se identificaron relaciones que no aportan un significado o contexto claro entre ambas entidades, como por ejemplo las tripletas: *<secretario, en el ramo, José Narro Robles>*, *<Comisión de la Verdad, sobre el caso, Iguala>* y *<AMLO, previo a vista de, Mike Pompeo>*. Esto se debe en primera instancia a que se abordó este problema teniendo en consideración dos entidades por oración, descartando así la posibilidad de encontrar una mejor relación entre otras entidades. En segunda instancia algunas relaciones se sobre *“generalizaron”* y no se contemplo una estructura más profunda de ellas.

6. Conclusiones

En este trabajo se presentó un enfoque no supervisado para extraer relaciones entre dos entidades empleando árboles de dependencias. Para este fin se

definieron un conjunto de patrones, dependencias universales y verbos entre entidades, e identificando las dependencias de *sujeto*, *predicado* y *objeto*. En los casos donde se definieron patrones también se establecieron las relaciones a identificar de forma manual, en cambio en los otros métodos se identificaron y extrajeron relaciones de forma intrínseca. En los resultados se observa que las relaciones con alta frecuencia son las que se definieron de forma manual. Por otro lado se identificó un número mayor de relaciones con frases verbales, así como relaciones que indican parentesco.

Como trabajo a futuro, se planea abordar esta tarea con más de dos entidades nombradas dentro de una oración. Además se pretende adaptar este enfoque al idioma inglés y usar un conjunto de datos del estado del arte, con la intención de comparar la eficacia de este método con los de la literatura similar. Así mismo, hacer uso de métodos supervisados aplicando algoritmos de clasificación y/o arquitecturas de redes neuronales.

Referencias

1. Afzal, N., Mitkov, R., y Farzindar, A. (2011). “Unsupervised relation extraction using dependency trees for automatic generation of multiple-choice questions”. *Canadian Conference on Artificial Intelligence*, pp. 32–43.
2. Banko M., Cafarella M. J., Soderland S., Broadhead M., y Etzioni O. (2007). Open information extraction for the web. *IJCAI’07: Proceedings of the 20th international joint conference on Artificial intelligence*, vol. 7, pp 2670–2676.
3. Bunescu, R., y Mooney, R. (2005). “A shortest path dependency kernel for relation extraction”. *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pp. 724–731.
4. Fundel, K., Küffner, R., y Zimmer, R. (2007). “RelEx-Relation extraction using dependency parse trees”. *Bioinformatics*, vol. 23, pp. 365–371.
5. Hasegawa, T., Sekine, S., y Grishman, R. (2004). “Discovering relations among named entities from large corpora”. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pp. 415–422.
6. Li, J., Luong, M. T., Jurafsky, D., y Hovy, E. (2015). “When Are Tree Structures Necessary for Deep Learning of Representations?”. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2304–2314.
7. Miwa, M., y Bansal, M. (2016). “End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures”. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 1105–1116.
8. Quan, C., Wang, M., y Ren, F. (2014). “An unsupervised text mining method for relation extraction from biomedical literature”. *PloS one*, vol. 9, pp. e102039.
9. Vo, D. T., y Bagheri, E. (2018). Open information extraction. In *Semantic Computing*, pp. 3–8.
10. Wu, Y., Zhang, Q., Huang, X. J., y Wu, L. (2009). “Phrase dependency parsing for opinion mining”. *Proceedings of the 2009 conference on empirical methods in natural language processing*, pp. 1533–1541.
11. Yan, Y., Okazaki, N., Matsuo, Y., Yang, Z., y Ishizuka, M. (2009). Unsupervised relation extraction by mining wikipedia texts using information from the web. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pp. 1021–1029.

Capítulo 2

Metodología y avances para el Reconocimiento Continuo de Lengua de Señas

Daniel Sánchez-Ruiz, J. Arturo Olvera-López, Ivan Olmos Pineda

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación

daniel.sanchez.4712@mail.com, aolvera@cs.buap.mx, iolmos@cs.buap.mx

Resumen. La población que sufre de audición limitada o la pérdida total de la misma tiene como uno de sus grandes retos el de la comunicación con la población hablante. Los avances tecnológicos han permitido tener mejores interacciones; esto se ha logrado gracias a los avances en el área del procesamiento de lengua de señas, donde se definen tres grandes áreas: el reconocimiento, la generación y la traducción automática de las distintas lenguas de señas. En este trabajo se realiza una propuesta para el problema del reconocimiento automático de lengua de señas, en él se identifican las áreas de oportunidad, la metodología propuesta, algunos resultados iniciales y se discuten los siguientes pasos para el trabajo futuro.

Palabras Clave: Reconocimiento de lengua de señas, reconocimiento de patrones, visión por computadora.

1 Introducción

Actualmente se calcula que 430 millones de personas tienen alguna discapacidad de pérdida de audición en todo el mundo, lo cual representa alrededor del 5% de la población total, y se estima que para el año 2050 habrá 700 millones de personas con este tipo de discapacidad. Tener una discapacidad auditiva significa que una persona no es capaz de escuchar del mismo modo que lo hace una persona oyente (en umbrales de audición de 25 dB o más en ambas orejas) (WHO, 2021).

Existen dos categorías principales de discapacidad auditiva: problemas de audición, que se refiere a personas con pérdida de audición media a severa (que generalmente suelen comunicarse de forma hablada y que mejoran su capacidad auditiva a través de implantes o dispositivos auditivos), y la otra categoría es la de sordos, donde se tiene pérdida de audición de forma profunda, lo cual implica una audición extremadamente limitada o la supresión total de todo tipo de capacidad auditiva. Este último grupo de personas suele comunicarse a través de lengua de señas (WHO, 2021).

De acuerdo con la Federación Mundial de Sordos, existen alrededor de 300 lenguas de señas en el mundo y 70 millones de personas sordas que los usan (WDF, 2021). Las lenguas de señas hacen uso de señas gesticuladas a través de las manos que se complementan con expresiones faciales y lenguaje corporal. Este lenguaje es un medio de comunicación con el cual buscan poder transmitir sus sentimientos, ideas y necesidades. Al igual que los lenguajes hablados, las lenguas de señas se componen de distintas reglas gramaticales y vocabulario dependiendo de la región, inclusive hay lenguas que cuentan con distintos componentes subregionales, por lo cual no se puede hablar de una lengua de señas universal.

Actualmente se busca mejorar la calidad de la vida de las personas a través de los distintos avances tecnológicos que se han desarrollado, y que se tienen planeados a futuro. No obstante, en el caso de las tecnologías de la comunicación, la gran mayoría sólo tienen soporte para lenguaje hablado o escrito, excluyendo a las lenguas de señas. Aunado a esto, existen pocas personas que dominen lengua de señas, por lo cual la comunidad sorda sufre de una barrera de comunicación con la mayoría de las personas (Bragg et al., 2019).

El procesamiento de lengua de señas ayudaría a romper estas barreras de comunicación de la comunidad sorda, esta área de procesamiento aborda las tareas de reconocimiento, generación y traducción de lengua de señas (Bragg et al., 2019). El reconocimiento automático de lengua de señas es el primer paso en cualquier sistema de procesamiento de lengua de señas y es el tema de interés de este trabajo. El reconocimiento automático de lengua de señas es por sí misma un área de investigación, que involucra conceptos referentes a las áreas de reconocimiento de patrones, visión por computadora, procesamiento de lenguaje natural y lingüística. Su objetivo es construir métodos y algoritmos con el propósito de identificar señas pertenecientes a una lengua de señas y su significado correspondiente en lengua escrita (Wadhawan y Kumar, 2019).

Un sistema de reconocimiento automático de lengua de señas puede ser clasificado en tres categorías con base a las señas que se busca reconocer: Deletreo con dedos, palabras aisladas y oraciones con señas continuas (Mohandes et al., 2014). El deletreo de palabras es usado en situaciones donde palabras nuevas, nombres de personas, lugares o palabras no tienen una seña definida, por lo cual tienen que ser “deletreadas” por movimientos de manos. En la categoría del reconocimiento de palabras aisladas, se analiza un dato de entrada, el cual va a representar una única seña. Finalmente, en la última categoría de reconocimiento de oraciones se busca reconocer el contenido de una conversación, la cual está compuesta de múltiples señas (Kamal et al., 2019). Este último tipo se conoce también como reconocimiento continuo de lengua de señas y es en el que se enfoca este trabajo.

El trabajo está organizado de la siguiente manera: en la Sección 2 se aborda un análisis del trabajo relacionado para el reconocimiento continuo de lengua de señas; en la Sección 3 se describe la metodología que se propone para la resolución del problema; la Sección 4 muestra algunos resultados obtenidos; finalmente, en la Sección 5 se listan las conclusiones y el trabajo futuro a realizar en la investigación.

2 Trabajo Relacionado

Uno de los trabajos más relevantes en el reconocimiento continuo de lengua de señas presentó un enfoque que hace uso de características multimodales (Koller et al., 2015), en particular aquellas que se refieren a la forma, posición y movimiento de las manos, pero también empieza a explorar expresiones faciales como características relevantes. Además de explorar esos datos, se detalla un conjunto de datos para el reconocimiento continuo de lengua de señas que se ha convertido en una referencia. Posteriormente (Koller et al., 2018) introdujeron una nueva metodología donde hacen uso de técnicas de aprendizaje profundo, algo que en las últimas investigaciones es cada vez más ocupado, en particular, en su trabajo se introduce una red neuronal convolucional (CNN) en un modelo oculto de Márkov (HMM), esto se hace con el propósito de ocupar las habilidades discriminativas de las CNN pero también el modelado de secuencias de los HMM, el resultado de esa fase se procesa con un modelo bayesiano dando buenos resultados.

En otro trabajo (Cui et al., 2017) exploran un enfoque similar donde también se utiliza una CNN para extraer características espacio temporales, sin embargo, en su caso hacen uso de una red de largo y corto término (LSTM) para el modelado de secuencias, es decir, la clasificación de las señas; en el trabajo mencionan que obtienen mejores resultados a comparación de (Koller et al., 2018). Un acercamiento distinto al problema lo presentan en (Camgoz et al., 2017), donde se emplean pequeñas sub redes que modelan conocimiento específico, el cual está relacionado a características espaciales y temporales, en cada sub red se emplea un enfoque similar donde se hace uso de una CNN y una LSTM. Además, ellos agregaron en sus experimentos un enfoque de fusión donde consideraron datos de la región de la mano y del *frame* completo, ocupando información local y global.

En una investigación más reciente (Elakkiya y Selvamani, 2019) se ocupa un concepto similar al de las sub redes, solo que en este caso se definen como sub unidades. Aquí el modelado de cada sub unidad no hace uso de métodos de aprendizaje profundo, sino que realizan el modelado espacial a través de operaciones geométricas y de posición con respecto a la región de las manos. En particular la posición también sirve para modelar el comportamiento temporal, con estas características definen un léxico sobre los datos y este sirve para la etapa de clasificación.

Si bien los resultados obtenidos en todos estos trabajos son prometedores y cada vez mejores, aún no son capaces de funcionar con la precisión necesaria en la práctica, pues estos sistemas todavía no son capaces de funcionar bajo cualquier contexto, ante cualquier usuario y con un vocabulario amplio. Esto se debe a que en la actualidad siguen existiendo varias áreas de oportunidad que deben de ser abordadas para lograr mejores resultados. En un trabajo reciente (Ibrahim et al., 2019), se mencionan varios retos por resolver, entre ellos se listan los siguientes:

- **Conjunto de datos estándar.** Los conjuntos de datos son un problema para lograr el desarrollo de un sistema de reconocimiento de lengua de señas de alto rendimiento. Por el momento el conjunto de datos RWTH-PHOENIX-Weather

2014 (Koller et al., 2015) es el de referencia, sin embargo el vocabulario del mismo es muy limitado, debido a que se abordan únicamente temas de predicciones meteorológicas. Un conjunto de datos que sea estándar, que tenga un vocabulario amplio y que no sea capturado en un ambiente controlado aún es un problema abierto en estas investigaciones.

- **Extracción de características multimodales.** Existen dos tipos de características para modelar las señas, las que se basan en la región de la mano y las que no son manuales, es decir, las que se basan en las expresiones faciales, la postura corporal o partes específicas del cuerpo. En los trabajos previamente examinados se observó que las características no manuales no se extraen con tanta frecuencia como las características manuales. De acuerdo con (Ibrahim et al., 2019) no considerar más a fondo las primeras conlleva a pérdida de información relevante en la etapa del modelado.
- **Detección transiciones entre señas.** Otro hallazgo clave en (Ibrahim et al., 2019) es que la detección de movimientos de epéntesis (movimientos realizados de forma natural entre señas) puede ser útil en la estimación de la longitud de cada una de las señas presentes en una oración. Es decir, la identificación de las transiciones entre señas es un problema que de ser resuelto puede ayudar en la correcta identificación de las señas presentes en una oración.
- **Adaptación a señantes.** Al igual que en cualquier lengua, la correcta comunicación depende enteramente de las personas que la ocupan, en este caso, los señantes. Varias personas se comunican de distinta forma a pesar de ocupar el mismo lenguaje, esto puede deberse a que hablan más rápido, inventan palabras o modifican las existentes a través de modismos. En la lengua de señas sucede lo mismo, por lo que la adaptación a distintos señantes y a las variantes que cada uno presente en su forma de comunicarse, sigue siendo un problema abierto para el reconocimiento de lengua de señas.

El contenido de la propuesta presentada en este documento hace énfasis en los puntos referentes a la extracción de características multimodales y en desarrollar una propuesta para la detección de transiciones entre señas, esto con el propósito de obtener resultados competitivos con los reportados en el estado del arte.

3 Metodología Propuesta

Para la resolución del problema del reconocimiento continuo y tomando en cuenta las áreas de oportunidad que se buscan abordar, se propone la metodología presentada en la Figura 1, donde de manera inicial se requiere de un conjunto de datos, posteriormente se realiza la etapa de detección de transiciones entre señas en oraciones, paso seguido se identifican y segmentan las regiones de interés, con las cuales se hará una etapa de extracción de características, las cuales servirán para la última fase donde a través de un

método de reconocimiento se identificarán y evaluarán los significados de las distintas señas. Dentro de esta metodología los puntos que contendrán el aporte, son el de la detección de transiciones entre señas, dado que esta tarea no se ha estudiado a profundidad y el enfoque que se busca ocupar (a través de modelos bayesianos) no ha sido empleado; además de esto, en la etapa de extracción de características, también se propone hacer la extracción de descriptores que no han sido considerados para los componentes no manuales, con lo cual se busca tener descriptores más robustos y generar un modelo discriminativo más completo.

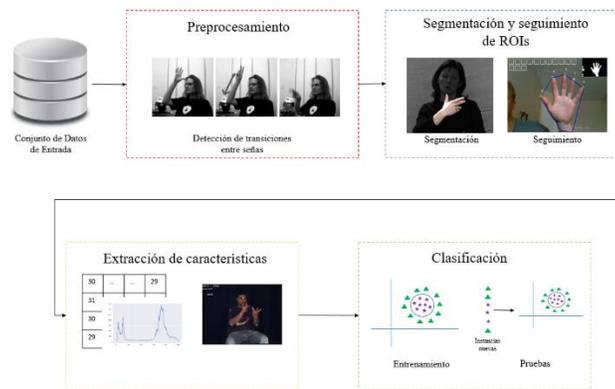


Figura 1. Metodología general propuesta para el reconocimiento continuo de lengua de señas

Con base en la metodología descrita, se detallan las actividades realizadas para la detección y segmentación de las regiones de interés, apartado en el cual se tienen avances hasta el momento. Cabe hacer mención que las regiones de interés que se consideran son las manos, el rostro y también se consideran los puntos clave relacionados con la estimación de la postura del cuerpo, todo ello se justifica en el hecho de tomar en cuenta características manuales y no manuales.

Para los experimentos realizados para la detección de las regiones de interés y la estimación de los puntos clave referentes a la postura del cuerpo, se ocupó el conjunto de datos LIBRAS (Quadros et al., 2021), en particular el de la región de Florianópolis, el cual contiene 639 registros, cada uno compuesto de videos con una resolución de 640x414 píxeles, capturados con una tasa de refresco de 30 fotogramas por segundo y un archivo eaf de anotación de las señas. En este conjunto de datos hay señantes en los intervalos de edad de los 30 a más de 60 años, además de esto, el vocabulario presente es variado, pues se tratan como fechas, indicaciones, historia, narrativa, objetos y hasta conversaciones en formato libre. Cada registro consta de la conversación entre dos personas y cuenta con 4 videos, una toma frontal a ambas personas, una frontal a cada persona y una tomada desde un ángulo aéreo.

Posteriormente se definió el conjunto de datos que serán parte de las siguientes etapas en la metodología empleada. De forma inicial se descartaron 40 registros, esto porque no

contenían videos, sólo tenían los archivos de anotación. De los 599 registros restantes sólo se tomaron en cuenta los videos referentes a las tomas frontales a cada uno de los participantes en las conversaciones, es decir, se descartaron los videos que corresponden a la toma frontal de los dos participantes y la toma aérea. Estos videos se descartaron ya que se busca hacer la extracción de descriptores con base en la región de la cara y en estos videos no se aprecia correctamente esta región. Una vez descartados los registros y videos denotados, el número total de videos con los que se cuenta es de 1198, estos videos representan la muestra total de datos con los que se va a trabajar.

Para la tarea de detectar la región de las manos y del rostro se ocupó el sistema YOLOv5 (Jocher et al., 2021), el cual es el estado del arte en la tarea de detección de objetos. Sin embargo, dado que las características que están presentes en los datos son muy específicas, es decir, las deformaciones y oclusiones que son comunes en la región de las manos y del rostro durante la ejecución de las señas. Como primera tarea se realizó la extracción de una muestra sistemática de imágenes con base a los datos del conjunto y posteriormente se realizó la anotación manual de las mismas en las regiones de interés, todo esto con el propósito de ocupar este conjunto de datos para la etapa de entrenamiento de YOLOv5. Este proceso se resume en la Figura 2.



Figura 2. Resumen de muestreo y anotación manual de datos

A través de un muestreo sistemático se consideró una submuestra de 50 videos, de los cuales fueron extraídas imágenes cada 15 segundos, lo que dio un total de 614 imágenes. Estas imágenes fueron anotadas manualmente con la herramienta gratuita CVAT (Sekachev et al., 2021) tanto para las clases de las manos y la del rostro. Una vez obtenido este conjunto de datos, se realizó el entrenamiento de YOLOv5 para obtener un modelo de inferencia para cada una de las regiones.

En el apartado de la estimación de la postura del cuerpo se ocupó el sistema OpenPose (Cao et al., 2019), el cual además de estimar los puntos clave referentes a las extremidades del cuerpo, también tiene la posibilidad de estimar puntos clave referentes a la región de la mano y del rostro.

4 Experimentos y Resultados

Los experimentos realizados para la detección de las regiones de interés y la estimación de los puntos clave referentes a la postura del cuerpo ocuparon tanto el conjunto de imágenes que se anotaron manualmente como el conjunto de datos LIBRAS (Quadros et al., 2021).

Para el proceso del entrenamiento se decidió ocupar la herramienta en línea de Google Colab, la cual permite hacer uso de equipos que cuentan con recursos considerables. El recurso al que se pudo tener acceso fue a una unidad de procesamiento gráfico (GPU) Tesla T4 que cuenta con una memoria RAM de 16 GB, 2,560 núcleos NVIDIA CUDA y un rendimiento de precisión única de 8.1 TFLOPS.

Tanto para el proceso de entrenamiento como en el de evaluación, el total de los datos del conjunto de imágenes se dividió en tres grupos: uno de entrenamiento, uno de validación y uno de prueba; para esto se siguió la práctica en las investigaciones de detección de objetos de ocupar 70% para entrenamiento, 20% para validación y 10% para pruebas; además de esto, los datos se redimensionaron a una resolución de 416x416 píxeles, esto último porque YOLOv5 obtiene mejores resultados en imágenes con resoluciones que sean múltiplos de 32 (Cao et al., 2019).

Los parámetros que deben de definirse para el proceso del entrenamiento son el número de épocas, el cual se estableció en 400 y que se definió de forma empírica a través de varios experimentos. Además de esto se define el tamaño del *batch*, el cual se dejó en el que venía predeterminado que es 16.

Una vez entrenado el modelo se evaluaron los resultados obtenidos, para poder llevar a cabo esta tarea se ocuparon dos métricas, la primera de ellas es *Intersection over Union* (IoU) (Redmon y Farhadi, 2018) que mide la superposición entre 2 regiones, una región viene delimitada por el cuadro envolvente de la anotación que se hizo manualmente y la otra región está dada por el que se predijo a través del modelo entrenado. Esta métrica nos da un intervalo de confianza sobre la predicción realizada, en la Figura 3 se aprecia cómo se calcula esta métrica.

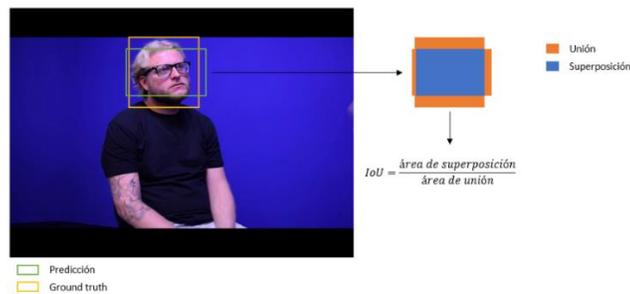


Figura 3. Representación visual del cálculo de la métrica IoU.

La segunda métrica es la del promedio de la precisión obtenida (*mAP*) (Redmon y Farhadi, 2018), para ello se hace uso de la primera métrica descrita del IoU; lo que se tiene que realizar es definir un intervalo de confianza y cualquier predicción realizada que sea inferior al intervalo definido, se considera como una predicción errónea y en caso contrario se toma como una predicción correcta. Al final se realiza la relación de las predicciones correctas sobre el número total de los datos de prueba y esa es la precisión obtenida para el modelo. En los trabajos de detección de objetos se suelen calcular la precisión con un intervalo de confianza de 0.5 (*mAP@0.5*) y el promedio de las precisiones obtenidas en el intervalo [0.5-0.95] (*mAP@0.5-0.95*) con un tamaño de incremento de 0.05. En la Tabla 1 se muestran los resultados de *mAP* y en la Figura 4 se muestran algunos resultados obtenidos en el conjunto de datos de prueba.

Tabla 1. Resultados obtenidos para la detección de las regiones del rostro y las manos con la métrica *mAP*.

Métrica	Manos	Rostro
<i>mAP@0.5</i>	96.22%	99.68%
<i>mAP@0.5-0.95</i>	62.22%	84.05%



Figura 4. Resultados obtenidos con el modelo de inferencia generado con YOLOv5.

Finalmente, para los experimentos de la estimación del cuerpo, se ocuparon directamente los datos de videos de LIBRAS. Se habilitó la opción de OpenPose de generar los puntos clave referentes la región de las manos y de los referentes a las expresiones faciales. Dado que para estas pruebas no se cuentan con datos anotados con los cuales realizar comparaciones y así obtener resultados cuantitativos, sólo se muestran de forma cualitativa en la Figura 5.



Figura 5. Resultados obtenidos de la estimación de puntos clave relacionados a la postura del cuerpo, de las manos y de expresiones faciales.

Se puede observar que los resultados obtenidos para todas las regiones de interés son bastante fiables. Con la correcta detección de estas regiones de interés se realizará una etapa de extracción de características, tanto para las tareas de detección de transiciones entre señas como para la de clasificación.

5 Conclusiones y Trabajo Futuro

En el presente trabajo se destacó la utilidad y el estado actual del reconocimiento de lengua de señas. A través de un breve análisis del trabajo relacionado, se identificaron áreas de oportunidad donde a forma de lista se destacaron algunos de los principales problemas que aún no han sido resueltos. Con base en estos puntos, se propone una metodología que aborda la detección de la transición entre señas y la extracción de características basadas en componentes manuales y no manuales, esto con el propósito de obtener resultados competitivos en la tarea del reconocimiento continuo de lengua de señas.

Posteriormente se presentó el conjunto de datos a emplear, se definieron sus características, el número de datos y los videos que serán considerados. Con dichos datos se realizaron los primeros experimentos que correspondieron a la detección y segmentación de las regiones de interés. Los resultados obtenidos se mostraron de forma cuantitativa y cualitativa, dichos resultados mostraron ser prometedores.

Finalmente, como trabajo futuro es necesario atacar las otras etapas de la metodología propuesta, el siguiente paso a abordar es el de la detección de transiciones entre señas. Para resolver dicha tarea, se propone de forma tentativa realizar la extracción de características con base a las regiones segmentadas en este trabajo y ocuparlas en una red dinámica bayesiana.

Referencias

- Bragg, D., Koller, O., Bellard, M., Berke, L., Boudrealt, P., Braffort, A., Caselli, N., Huenerfauth, M., Kacorri, H., Verhoef, T., Vogler, C., y Morris, M. R. (2019). "Sign Language Recognition, Generation, and Translation: An Interdisciplinary Perspective". *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*.
- Camgoz, N. C., Hadfield, S., Koller, O., y Bowden, R. (2017). "SubUNets: End-to-End Hand Shape and Continuous Sign Language Recognition". *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3075-3084.
- Cao, Z., Hidalgo, G., Simon, T., Wei, S. E., y Sheikh, Y. (2019). "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields". *IEEE transactions on pattern analysis and machine intelligence*, 43(1), 172-186.
- Cui, R., Liu, H., y Zhang, C. (2017). "Recurrent Convolutional Neural Networks for Continuous Sign Language Recognition by Staged Optimization". *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1610-1618.
- Elakkiya, R., y Selvamani, K. (2019). "Subunit sign modeling framework for continuous sign language recognition". *Computers and Electrical Engineering*, vol. 74, pp. 379-390.
- Ibrahim, N. B., Zayed, H. H., y Selim, M. M. (2019). "Advances, Challenges and Opportunities in Continuous Sign Language Recognition". *Journal of Engineering and Applied Sciences*, vol. 15, no. 5, pp. 1205-1227.
- Jocher, G., Stoken, A., Borovec, J., NanoCode012, ChristopherSTAN, Changyu, L., ... y Yu, L. (2021). "ultralytics/yolov5: v4.0 - nn.SiLU() activations, Weights & Biases logging, PyTorch Hub integration". *Zenodo*. doi:10.5281/zenodo.4418161.
- Kamal, S. M., Chen, Y., Li, S., Shi, X., y Zheng, J. (2019). "Technical Approaches to Chinese Sign Language Processing: A Review". *IEEE Access*, vol. 7, pp. 96926-96935.
- Koller, O., Forster, J., y Ney, H. (2015). "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers". *Computer Vision and Image Understanding*, vol. 141, pp. 108-125.
- Koller, O., Zargaran, S., Ney, H., y Bowden, R. (2018). "Deep Sign: Enabling Robust Statistical Continuous Sign Language Recognition via Hybrid CNN-HMMs". *International Journal of Computer Vision*, vol. 126, no. 12, pp. 1311-1325.
- Mohandes, M., Deriche, M., y Liu, J. (2014). "Image-Based and Sensor-Based Approaches to Arabic Sign Language Recognition". *IEEE Transactions on Human-Machine Systems*, vol. 44, no. 4, pp. 551-557.
- Quadros, R. M., Schmitt, D., Lohn, J., de Arantes Leite, T. (2021). Corpus de Libras.
- Redmon, J., y Farhadi, A. (2018). "Yolov3: An incremental improvement." *arXiv preprint arXiv:1804.02767*.
- Sekachev, B., Manovich, N., Zhavoronkov, A., Kalinin, D., Hoff, B., TOSmanov, ..., y Truong, T. (2021). "opencv/cvat: v1.1.0 (Version v1.1.0)". *Zenodo*. <http://doi.org/10.5281/zenodo.4009388>
- Wadhawan, A., y Kumar, P. (2019). "Sign Language Recognition Systems: A Decade Systematic Literature Review". *Archives of Computational Methods in Engineering*, 0123456789
- WDF, World Federation of the Deaf. (2021). *Our Work*. Recuperado de <https://wfdeaf.org/our-work/>.
- WHO, World Health Organization. (2021). *Deafness and hearing loss*. Recuperado de <https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss>.

Capítulo 3

Análisis de técnicas para la detección y clasificación de la retinopatía diabética mediante imágenes digitales

David Ferreiro Piñeiro, Iván Olmos Pineda, Arturo Olvera López

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación

davidfp9210@gmail.com, ivanoprkl@gmail.com, aolvera@cs.buap.mx

Resumen. Se realiza una revisión de las principales técnicas existentes para la detección automática de la Retinopatía Diabética en imágenes de la retina utilizando diferentes técnicas de análisis de imágenes, visión computacional e inteligencia artificial. A pesar de observarse algunos resultados en el área, estos pueden ser mejorados con vistas a lograr desarrollos más sensibles y específicos en el diagnóstico de la enfermedad. A partir de este análisis, se señala un área de oportunidad para permitir el desarrollo de nuevos mecanismos de detección y clasificación de interés para la comunidad científica.

Palabras Clave: Retinopatía Diabética, visión computacional, inteligencia artificial.

1 Introducción

El descontrol metabólico generado por la diabetes provoca el desarrollo de una microangiopatía retiniana progresiva, conocida como Retinopatía Diabética (RD), que se manifiesta con el desarrollo de anomalías clínicas de difícil detección (Rudas y Sánchez Torres, 2013) afectando diferentes estructuras vasculares en la retina. La enfermedad se manifiesta con el desarrollo de abultamientos en las paredes de los vasos sanguíneos, hemorragias, depósitos de lipoproteínas, así como algunos infartos retinianos causados por la falta de irrigación sanguínea. Estas lesiones caracterizan el estado inicial de la enfermedad denominado No Proliferativo (RDNP), pero puede evolucionar hasta un estado severo denominado Proliferativo (RDP), causante de ceguera a los pacientes, que se caracteriza por el desarrollo de nuevos vasos sanguíneos defectuosos, hemorragias extensas e incluso, desprendimientos de la retina (Eguía Martínez et al., 2009; Río Torres et al., 2018).

En la figura 1 se puede observar las diferentes manifestaciones clínicas de la enfermedad.



Figura 1: Principales manifestaciones clínicas de la RD (Rudas y Sánchez Torres, 2013)

Los métodos clínicos de diagnóstico de la enfermedad son altamente especializados, costosos, subjetivos, requieren de personal altamente calificado y se realizan de forma manual. Estas características limitan su alcance para un amplio sector de la población con menos recursos. Una alternativa a estos procesos se ha desarrollado a partir del empleo de técnicas de visión computacional, con las ventajas de homogeneizar los procesos, abaratar los costos y por tanto democratizar la atención. Sin embargo, los resultados obtenidos en ese sentido resultan limitados porque la sensibilidad y la especificidad reportada es inferior, en muchos casos, a estas mismas métricas evaluadas en los métodos clínicos.

Estas razones incentivan los estudios para lograr algoritmos o sistemas capaces de diagnosticar a la enfermedad con resultados superiores a los ya obtenidos. Los desarrollos en el área son continuos y en muchas direcciones, ello hace necesario que se establezcan revisiones del estado del arte que permitan establecer los alcances actuales y a partir de ahí definir posibles áreas de investigación. En el presente trabajo se realiza una revisión bibliográfica de los trabajos publicados, en este sentido, entre los años 2017-2020, las revisiones anteriores solo analizan periodos anteriores a 2019; por tanto, se encuentran limitadas y en ello reside la relevancia del estudio que se presenta. Resulta importante aclarar que, dada la rapidez del desarrollo investigativo, los resultados presentados tienen una incidencia temporal, permitiendo identificar las principales áreas de trabajo actuales.

2 Estado del arte

La evolución de la RD permite identificar diferentes estados caracterizados por las lesiones presentes. En este sentido, se desarrollan algoritmos enfocados a la detección de la enfermedad (paciente sano-paciente enfermo) y otros más específicos que detallan el estado correspondiente (sano, leve, moderado, severo o proliferativo). Estos algoritmos de clasificación pueden dividirse en dos enfoques diferentes: un primer enfoque orientado a la segmentación manual de las lesiones características para realizar, con posterioridad, las labores de clasificación y un segundo enfoque en el que los procesos de extracción de características y la clasificación se realizan automáticamente. Con el desarrollo tecnológico,

resulta posible observar un predominio del segundo enfoque, aunque ambos resultan válidos y vigentes.

En los párrafos siguientes se detallarán algunos aspectos de interés de los artículos analizados, así como de los principales repositorios públicos de imágenes que sustentan las investigaciones. Los artículos analizados manejan diferentes métricas para evaluar su desempeño, dentro de esas métricas se encuentra: la exactitud (Exa), especificidad (Esp), sensibilidad (Sen), el área bajo la curva de operación (AUC), entre otras.

2.1 Bases de datos de imágenes de la retina

La disponibilidad de datos resulta esencial en el desarrollo de las investigaciones asociadas a aspectos médicos (Khan et al., 2021; Mateen et al., 2020); con este objetivo, se han desarrollado colecciones de imágenes destinados a la detección y clasificación de enfermedades visuales. Estos repositorios fueron conformados bajo diferentes condiciones y las imágenes capturadas con equipos de distintas características, pero brindan la oportunidad de evaluar los desarrollos en el área y comparar los resultados obtenidos. Seguidamente se mencionan los principales repositorios públicos disponibles utilizados con frecuencia en los procesos de evaluación.

La base de datos Messidor: “*Methods to Evaluate Segmentation and Indexing Techniques in the field of Retinal Ophthalmology*” (Decencière et al., 2014) está compuesta por 1200 imágenes a color clasificadas de acuerdo con su grado de retinopatía y al riesgo de desarrollar edema macular, estas anotaciones fueron realizadas por especialistas de los centros médicos participantes. Otra base de datos disponible es IDRiD: “*Indian Diabetic Retinopathy Image Dataset*” (Porwal et al., 2018, 2020) y proporciona información sobre el grado de RD y del edema macular, además señala las lesiones típicas que describe a cada una de las enfermedades. El conjunto destinado a la clasificación está conformado por 516 imágenes y está considerada como uno de los repositorios más completos porque permite explorar diferentes aspectos para la detección y la clasificación.

Otras bases de datos son DIARETDB0 y DIARETDB1: “*Standard Diabetic Retinopathy Database Calibration level 0 o 1*”, respectivamente (T. Kauppi et al., 2007; Tomi Kauppi et al., 2006). Estas bases brindan las imágenes para desarrollar la evaluación de procesos de detección de RD. La base de nivel de calibración 1, cuenta con 89 imágenes (84 contienen signos no proliferativos leves), mientras que la base de nivel de calibración 0 está formada por 130 imágenes (20 normales y 110 con signos de RD). Estos repositorios fueron utilizados durante los procesos de experimentación de los diferentes algoritmos analizados, de ahí la importancia de conocer la composición de cada uno.

2.2 Trabajos enfocados a la detección de la RD

En la tabla 1 se presenta un resumen de los artículos que se enfocan en la detección de la RD. Resulta posible observar que existe una tendencia al empleo de las Redes Neuronales Convolucionales (CNN) para realizar los procesos de extracción de características y la

posterior clasificación. En este sentido se observan modificaciones al número de capas convolucionales orientadas a la detección de lesiones específicas en las imágenes de entrada (Zago et al., 2020); otras variantes se enfocan a la modificación de redes conocidas (Hajabdollahi et al., 2019) utilizando los paradigmas de transferencia de aprendizaje sobre la base de datos de ImageNet, o modificando las funciones de activación (Kamble y Kokate, 2020; Kumar et al., 2020), para de este modo reducir la ocurrencia de los procesos de sobreajuste que repercuten durante el proceso de clasificación.

Tabla 1: Resumen de artículos enfocados a la detección de la RD.

Artículo	Técnica utilizada	Resultados obtenidos
(Zago et al., 2020)	CNN	Sen = 94% AUC = 0.912
(Kumar et al., 2020)	CNN- Radial	Sen = 87% Esp = 93%
(Kamble y Kokate, 2020)	CNN-Radial	Exa = 71.2% (DIARETDB0) Exa = 89.4% (DIARETDB1)
(Hajabdollahi et al., 2019)	CNN (VGG-16)	Exa = 92% Sen = 98.33% Esp = 83.7%
(Qomariah et al., 2019)	SVM	Exa = 95.83% (Messidor-12) Exa = 95.24% (Messidor-13)
(de la Torre et al., 2020)	CNN	Exa = 85.7% (Kaggle) Exa = 91% (Messidor-2)
(Washburn et al., 2020)	Impulso Adaptativo	Exa = 98.4%
(Saman et al., 2020)	SVM	No reportado

De manera similar, se analizaron algunas aproximaciones híbridas, donde la extracción de característica se realiza utilizando enfoques basados en aprendizaje profundo y la clasificación se basa en la implementación de máquinas de soporte de vectores (SVM) o clasificadores de impulso adaptativos. Estos enfoques resuelven la problemática de la extracción de características manuales y permiten la relativamente sencilla implementación de otras estructuras de clasificación.

Una de las desventajas prácticas identificadas durante el desarrollo de modelos basados en CNN se encuentra en la capacidad para interpretar sus resultados, por estas razones en (de la Torre et al., 2020) se propone un modelo que brinda la puntuación de importancia de cada pixel para determinar los puntos relevantes en la toma de decisiones y de esta manera facilitar los procesos de verificación por parte de los expertos humanos.

2.3 Trabajos enfocados a la clasificación de la RD

En la tabla 2 se presenta un resumen de los artículos enfocados a las labores de clasificación de la RD en las categorías detalladas con anterioridad. En este sentido es posible identificar diferentes enfoques.

La clasificación basada en técnicas estadísticas realiza la extracción de las características de acuerdo con aspectos relacionados a la morfología de la red vascular, determinando el área de la región capilar e incluyendo descriptores de la región avascular de la fóvea (Abdelsalam, 2020) o discriminando la ubicación de las diferentes lesiones, esta discriminación permite la reducción de los tiempos de procesamiento necesario en cada caso (M. K. Dutta et al., 2017).

Otro enfoque explota las potencialidades de la máquina de vectores de soporte (SVM), en este caso se realiza una extracción de las características de textura utilizando una técnica de histograma de gradiente orientado (Sarwinda et al., 2018) para de esta manera detectar las deformaciones presentes en las imágenes en forma de lesiones. En (AbdelMaksoud et al., 2020; Issac et al., 2020) se explora otras variantes al emplear técnicas de análisis de imágenes para resaltar lesiones específicas y utilizar la umbralización para la segmentación de cada una de forma tal que se detecten los cambios patológicos producto a la enfermedad y estas características le permitan al clasificador categorizar cada una de las imágenes.

Tabla 2: Resumen de artículos enfocados a la clasificación de la RD.

Artículo	Técnica utilizada	Resultados obtenidos
(Abdelsalam, 2020)	Estadístico	Exa = 97%
(M. K. Dutta et al., 2017)	Estadístico	Exa = 90%
(Birajdar et al., 2020)	CNN	Exa = 88%
(Saranya y Prabakaran, 2020)	CNN	Exa = 90.89%
(Galdran et al., 2020)	CNN	AUC = 0.9158
(Martinez-Murcia et al., 2020)	CNN (ResNet50)	AUC = 0.93 (leve) AUC = 0.81 (moderada) AUC = 0.92 (severa)
(Samanta et al., 2020)	CNN (DenseNet)	Cohens Kappa = 0.8836
(Shanthi y Sabeenian, 2019)	CNN (AlexNet)	Exa = 96.6% (sana) Exa = 96.2% (leve) Exa = 96.6% (moderada)
(Bourouis et al., 2018)	SVM	Exa = 90.7%
(Sarwinda et al., 2018)	SVM	Exa = 85%
(Issac et al., 2020)	SVM	Exa = 84% (Messidor-1) Exa = 92.13% (DIARETDB0)
(AbdelMaksoud et al., 2020)	SVM	Exa = 89.2% Sen = 85.1% Esp = 85.2%
(Frazao et al., 2019)	Multiclasificador	Exa = 74.6%

		Sen = 90.1% Esp = 56.1%
(Wu et al., 2020)	CNN	Exa = 56.16% (IDRiD) Exa = 83.10 (Kaggle)
(Momeni Pour et al., 2020)	CNN (EfficientNet-B5)	AUC = 0.945
(Saxena et al., 2020)	CNN (Inception-V3; InceptionResNet-V2)	AUC = 0.958 (Messidor-1) AUC = 0.92 (Messidor-2)
(Katada et al., 2020)	Híbrida (Inception-V3, SVM)	Sen = 81.5% Esp = 71.9%
(Nguyen et al., 2020)	CNN (VGG-16/19)	Sen = 80% Esp = 82%
(S. Dutta et al., 2018)	CNN	Exa = 72.5%
(R C et al., 2020)	CNN (ResNet-101)	Exa = 82%
(Zhou et al., 2018)	CNN (AlexNet)	AUC = 0.925 (Kaggle) AUC = 0.96 (Messidor)
(Sangeethaa y Uma Maheswari, 2018)	CNN	Exa = 96.9% Sen = 98% Esp = 93%
(Li et al., 2019)	CNN (Inception-V3)	Exa = 93.49% Sen = 96.93% Esp = 95%
(Mobeen-ur-Rehman et al., 2019)	CNN	Exa = 98.15%
(Butt et al., 2019)	CNN multicanal	Exa = 97.08%
(Afrin y Shill, 2019)	Lógica difusa	Exa = 95.63% Sen = 95.77% Esp = 94.44%

Los enfoques basados en CNN se orientan a dos aspectos: el diseño de nuevas redes convolucionales o la modificación de redes conocidas para adaptarlas a los requerimientos del problema a tratar. Ambos enfoques resultan afectados por procesos de sobreajuste de parámetros, estos procesos son mitigados en las redes de diseño propio modificando la estructura propuesta, en el resto de los casos se utilizan enfoques basados en la transferencia de aprendizaje para intentar mitigar los efectos adversos. Sin embargo, resulta evidente la tendencia al empleo de este tipo de aproximación en las investigaciones analizadas. Otro enfoque interesante, se discute en (Frazao et al., 2019), en este caso se realiza una extracción de características locales utilizando diferentes técnicas de procesamiento de imágenes para con posterioridad realizar la clasificación dentro de un marco multclasificador conformado por SVM, K-NN y una red neuronal artificial. La clasificación final se realiza ponderando

la salida de cada uno de los clasificadores individuales, resultando interesante este enfoque por las potencialidades que presenta considerando los diferentes algoritmos que pueden ser considerados para realizar la clasificación o la extracción adecuada de las características.

3 Conclusiones

El diagnóstico de la RD es un problema de investigación activo dentro del área de visión computacional e inteligencia artificial, donde se busca el desarrollo de un algoritmo que permita realizar los procesos de detección y clasificación de manera automática, eliminando de este modo los problemas de los métodos clínicos actuales.

El trabajo presentado evidenció que existen diferentes enfoques para abordar esta problemática que abarcan desde la extracción de características utilizando técnicas de procesamiento de imágenes hasta el diseño de métodos de aprendizaje profundo que permiten automatizar estos procesos. En este sentido, existe una tendencia al desarrollo de nuevas redes neuronales o a la modificación de las ya existentes para resolver las problemáticas derivadas, sin embargo, debido a la limitada cantidad de datos disponibles, estos enfoques sufren de procesos de sobreajustes que requieren el empleo de métodos de transferencia de aprendizaje. Como desventaja, además, se puede identificar que estos modelos resultan poco interpretables para los especialistas y por tanto no pueden ser evaluados de forma simple por estos.

En este sentido, se considera que existe un área de oportunidad en el desarrollo de un algoritmo basado en un enfoque híbrido-multclasificador, donde la extracción de las características se realice utilizando un enfoque de aprendizaje profundo, pero la clasificación considere un marco de votación ponderada de diferentes algoritmos de clasificación. Además, resultaría interesante considerar incluir mecanismos que permitan interpretar de manera simple los resultados para garantizar la evaluación de los resultados directamente por los especialistas humanos.

Referencias

- AbdelMaksoud, E., Barakat, S., y Elmogy, M. (2020). "A comprehensive diagnosis system for early signs and different diabetic retinopathy grades using fundus retinal images based on pathological changes detection". *Computers in Biology and Medicine*, vol. 126, pp. 104039.
- Abdelsalam, M. M. (2020). "Effective blood vessels reconstruction methodology for early detection and classification of diabetic retinopathy using OCTA images by artificial neural network". *Informatics in Medicine Unlocked*, vol. 20, pp. 100390.
- Afrin, R., y Shill, P. C. (2019). "Automatic Lesions Detection and Classification of Diabetic Retinopathy Using Fuzzy Logic". *2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, pp. 527-532.
- Birajdar, U., Gadhawe, S., Chikodkar, S., Dadhich, S., y Chiwhane, S. (2020). "Detection and

- Classification of Diabetic Retinopathy Using AlexNet Architecture of Convolutional Neural Networks”. En *Proceeding of International Conference on Computational Science and Applications* (pp. 245-253).
- Bourouis, S., Zaguia, A., y Bouguila, N. (2018). “Hybrid Statistical Framework for Diabetic Retinopathy Detection”. En A. Campilho, F. Karray, & B. ter Haar Romeny (Eds.), *International Conference Image Analysis and Recognition* (pp. 687-694). Springer International Publishing.
- Butt, M. M., Latif, G., Iskandar, D. N. F. A., Alghazo, J., y Khan, A. H. (2019). “Multi-channel Convolutions Neural Network Based Diabetic Retinopathy Detection from Fundus Images”. *Procedia Computer Science*, vol. 163, pp. 283-291.
- de la Torre, J., Valls, A., y Puig, D. (2020). “A deep learning interpretable classifier for diabetic retinopathy disease grading”. *Neurocomputing*, vol. 396, pp. 465-476.
- Decencière, E., Zhang, X., Cazuguel, G., Lay, B., Cochener, B., Trone, C., Gain, P., Ordóñez-Varela, J. R., Massin, P., Erginay, A., Charton, B., y Klein, J. C. (2014). “Feedback on a publicly distributed image database: The Messidor database”. *Image Analysis and Stereology*.
- Dutta, M. K., Parthasarathi, M., Ganguly, S., Ganguly, S., y Srivastava, K. (2017). “An efficient image processing based technique for comprehensive detection and grading of nonproliferative diabetic retinopathy from fundus images”. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 5, pp. 195-207.
- Dutta, S., Manideep, B. C. S., Basha, S. M., Caytiles, R. D., y Iyengar, N. C. S. N. (2018). “Classification of Diabetic Retinopathy Images by Using Deep Learning Models”. *International Journal of Grid and Distributed Computing*, vol. 11, pp. 99-106.
- Eguía Martínez, F., Rió Torres, M., Capote Cabrera, A., Ríos Caso, R., Hernández Silva, J. R., Gómez Cabrera, C., Ramírez García, L. K., Díaz Azze, M., Pacheco Ledesma, P., Agramonte Centelles, I., Gómez Castillo, Z., Fernández García, K., Jareño Ochoa, M., Cuevas Ruiz, J., Castillo Pérez, A., López Hernández, S., Rodríguez de la Paz, U., Vega Quroga, B., Esvallona Leyva, E., ... Chiang Rodríguez, C. (2009). *Manual de Diagnóstico y Tratamiento en Oftalmología*. Ciencias Médicas.
- Frazao, L. B., Theera-Umpon, N., y Auephanwiriyakul, S. (2019). “Diagnosis of diabetic retinopathy based on holistic texture and local retinal features”. *Information Sciences*, vol. 475, pp. 44-66.
- Galdran, A., Chelbi, J., Kobi, R., Dolz, J., Lombaert, H., ben Ayed, I., y Chakor, H. (2020). “Non-uniform Label Smoothing for Diabetic Retinopathy Grading from Retinal Fundus Images with Deep Neural Networks”. *Translational Vision Science & Technology*, vol. 9, pp. 34.
- Hajabdollahi, M., Esfandiarpour, R., Najarian, K., Karimi, N., Samavi, S., y Reza Soroushmehr, S. M. (2019). “Hierarchical Pruning for Simplification of Convolutional Neural Networks in Diabetic Retinopathy Classification”. *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 970-973.
- Issac, A., Dutta, M. K., y Travieso, C. M. (2020). “Automatic computer vision-based detection and quantitative analysis of indicative parameters for grading of diabetic retinopathy”. *Neural Computing and Applications*, vol. 32, pp. 15687-15697.
- Kamble, V. V., y Kokate, R. D. (2020). “Automated diabetic retinopathy detection using radial basis function”. *Procedia Computer Science*, vol. 167, pp. 799-808.
- Katada, Y., Ozawa, N., Masayoshi, K., Ofuji, Y., Tsubota, K., y Kurihara, T. (2020). “Automatic screening for diabetic retinopathy in interracial fundus images using artificial intelligence”. *Intelligence-Based Medicine*, vol. 3-4, pp. 100024.
- Kauppi, T., Kalesnykiene, V., Kamarainen, J. K., Lensu, L., Sorri, I., Raninen, A., Voutilainen, R., Pietilä, J., Kälviäinen, H., y Uusitalo, H. (2007). “The DIARETDB1 diabetic retinopathy database and evaluation protocol”. *BMVC 2007 - Proceedings of the British Machine Vision*

- Conference 2007*, pp. 1-18.
- Kauppi, Tomi, Kalesnykiene, V., Kamarainen, J., Lensu, L., y Sorri, I. (2006). "DIARETDB0 : Evaluation Database and Methodology for Diabetic Retinopathy Algorithms". *Machine Vision and Pattern Recognition Research Group, Lappeenranta University of Technology, Finland.*, pp. 1-17.
- Khan, S. M., Liu, X., Nath, S., Korot, E., Faes, L., Wagner, S. K., Keane, P. A., Sebire, N. J., Burton, M. J., y Denniston, A. K. (2021). "A global review of publicly available datasets for ophthalmological imaging: barriers to access, usability, and generalisability". *The Lancet Digital Health*, vol. 3, pp. e51-e66.
- Kumar, S., Adarsh, A., Kumar, B., y Singh, A. K. (2020). "An automated early diabetic retinopathy detection through improved blood vessel and optic disc segmentation". *Optics & Laser Technology*, vol. 121, pp. 105815.
- Li, F., Liu, Z., Chen, H., Jiang, M., Zhang, X., y Wu, Z. (2019). "Automatic Detection of Diabetic Retinopathy in Retinal Fundus Photographs Based on Deep Learning Algorithm". *Translational Vision Science & Technology*, vol. 8, pp. 4.
- Martinez-Murcia, F. J., Ortiz, A., Ramírez, J., Górriz, J. M., y Cruz, R. (2020). "Deep residual transfer learning for automatic diagnosis and grading of diabetic retinopathy". *Neurocomputing*.
- Mateen, M., Wen, J., Hassan, M., Nasrullah, N., Sun, S., y Hayat, S. (2020). "Automatic Detection of Diabetic Retinopathy: A Review on Datasets, Methods and Evaluation Metrics". *IEEE Access*, vol. 8, pp. 48784-48811.
- Mobeen-ur-Rehman, Khan, S. H., Abbas, Z., y Rizvi, S. D. (2019). "Classification of Diabetic Retinopathy Images Based on Customised CNN Architecture". *2019 Amity International Conference on Artificial Intelligence (AICAI)*, pp. 244-248.
- Momeni Pour, A., Seyedarabi, H., Abbasi Jahromi, S. H., y Javadzadeh, A. (2020). "Automatic Detection and Monitoring of Diabetic Retinopathy Using Efficient Convolutional Neural Networks and Contrast Limited Adaptive Histogram Equalization". *IEEE Access*, vol. 8, pp. 136668-136673.
- Nguyen, Q. H., Muthuraman, R., Singh, L., Sen, G., Tran, A. C., Nguyen, B. P., y Chua, M. (2020). "Diabetic Retinopathy Detection using Deep Learning". *Proceedings of the 4th International Conference on Machine Learning and Soft Computing*, vol. 16, pp. 103-107.
- Porwal, P., Pachade, S., Kamble, R., Kokare, M., Deshmukh, G., Sahasrabudde, V., y Meriaudeau, F. (2018). "Indian Diabetic Retinopathy Image Dataset (IDRiD)". *IEEE Dataport*.
- Porwal, P., Pachade, S., Kokare, M., Deshmukh, G., Son, J., Bae, W., Liu, L., Wang, J., Liu, X., Gao, L., Wu, T., Xiao, J., Wang, F., Yin, B., Wang, Y., Danala, G., He, L., Choi, Y. H., Lee, Y. C., ... Mériaudeau, F. (2020). "IDRiD: Diabetic Retinopathy – Segmentation and Grading Challenge". *Medical Image Analysis*, vol. 59, pp. 101561.
- Qomariah, D. U. N., Tjandrasa, H., y Fatichah, C. (2019). "Classification of Diabetic Retinopathy and Normal Retinal Images using CNN and SVM". *2019 12th International Conference on Information & Communication Technology and System (ICTS)*, pp. 152-157.
- R C, K., Ankalaki, S., y Majumdar, J. (2020). "A Study of Performance Evaluation of Convolution Neural Network for Diabetic Retinopathy". *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, pp. 5009-5014.
- Rio Torres, M., Fernández Argones, L., Hernández Silva, J. R., y Ramos López, M. (2018). *Oftalmología: Diagnóstico y tratamiento* (2.^a ed.). Ciencias Médicas.
- Rudas, J., y Sánchez Torres, G. (2013). "Detección de patologías derivadas de las afecciones diabéticas: una revisión del análisis digital de imágenes de retina". *Ingeniería y Desarrollo*, vol. 31, pp. 316-338.
- Saman, G., Gohar, N., Noor, S., Shahnaz, A., Idress, S., Jehan, N., Rashid, R., y Khattak, S. S. (2020).

- “Automatic detection and severity classification of diabetic retinopathy”. *Multimedia Tools and Applications*, vol. 79, pp. 31803-31817.
- Samanta, A., Saha, A., Satapathy, S. C., Fernandes, S. L., y Zhang, Y.-D. (2020). “Automated detection of diabetic retinopathy using convolutional neural networks on a small dataset”. *Pattern Recognition Letters*, vol. 135, pp. 293-298.
- Sangeethaa, S. N., y Uma Maheswari, P. (2018). “An Intelligent Model for Blood Vessel Segmentation in Diagnosing DR Using CNN”. *Journal of Medical Systems*, vol. 42, pp. 175.
- Saranya, P., y Prabakaran, S. (2020). “Automatic detection of non-proliferative diabetic retinopathy in retinal fundus images using convolution neural network”. *Journal of Ambient Intelligence and Humanized Computing*, 0123456789.
- Sarwinda, D., Siswantining, T., y Bustamam, A. (2018). “Classification of Diabetic Retinopathy Stages using Histogram of Oriented Gradients and Shallow Learning”. *2018 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, pp. 83-87.
- Saxena, G., Verma, D. K., Paraye, A., Rajan, A., y Rawat, A. (2020). “Improved and robust deep learning agent for preliminary detection of diabetic retinopathy using public datasets”. *Intelligence-Based Medicine*, vol. 3-4, pp. 100022.
- Shanthi, T., y Sabeenian, R. S. (2019). “Modified Alexnet architecture for classification of diabetic retinopathy images”. *Computers & Electrical Engineering*, vol. 76, pp. 56-64.
- Washburn, P. S., Mahendran, Dhanasekharan, Periyasamy, y Murugeswari. (2020). “Investigation of severity level of diabetic retinopathy using adaboost classifier algorithm”. *Materials Today: Proceedings*, vol. 33, pp. 3037-3042.
- Wu, Z., Shi, G., Chen, Y., Shi, F., Chen, X., Coatrieux, G., Yang, J., Luo, L., y Li, S. (2020). “Coarse-to-fine classification for diabetic retinopathy grading using convolutional neural network”. *Artificial Intelligence in Medicine*, vol. 108, pp. 101936.
- Zago, G. T., Andreão, R. V., Dorizzi, B., y Teatini Salles, E. O. (2020). “Diabetic retinopathy detection using red lesion localization and convolutional neural networks”. *Computers in Biology and Medicine*, vol. 116, pp. 103537.
- Zhou, L., Zhao, Y., Yang, J., Yu, Q., y Xu, X. (2018). “Deep multiple instance learning for automatic detection of diabetic retinopathy in retinal images”. *IET Image Processing*, vol. 12, pp. 563-571.

Capítulo 4

Red Neuronal Artificial para Extraer Parámetros Dinámicos de Robots a Partir de Información Incompleta de su Movimiento

Carlos Leopoldo Carreón Díaz de León¹, Sergio Vergara Limon², Juan Manuel González Calleros¹, María Aurora Diozcora Vargas Treviño²

¹ Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación

² Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Electrónica

carlos.carreond@alumno.buap.mx, sergio.vergara@correo.buap.mx,
juan.gonzalez@cs.buap.mx, aurora.vargas@correo.buap.mx

Resumen. Las redes neuronales artificiales son una herramienta útil para caracterizar un sistema robótico que dispone de tecnología de medición limitada. Con el procesamiento adecuado de la información disponible de un robot se puede implementar un sistema de aprendizaje supervisado que indique los parámetros dinámicos del sistema. Con la caracterización de robots por medio de la identificación de parámetros dinámicos, las aplicaciones de control de movimiento se verán beneficiadas y, en consecuencia, la producción de bienes y servicios también lo estará. En el estado del arte de identificación paramétrica se abordan técnicas de optimización de la trayectoria que deben seguir los robots para poder identificar los parámetros del robot en el menor tiempo posible, sin embargo, ninguno de estos trabajos considera las mediciones como un conjunto de información incompleta. La metodología propuesta consta de la primera versión de un algoritmo iterativo que utiliza una red neuronal artificial que estima los parámetros dinámicos de un robot. Se presenta la metodología general lograda hasta el momento. El trabajo futuro de esta investigación es la aplicación del algoritmo propuesto a dos casos de estudio para lograr extraer los parámetros dinámicos y compararlos con datos de simulación para determinar la exactitud del algoritmo.

Palabras Clave: Red Neuronal, Robots, Parámetros dinámicos.

1 Introducción

La identificación paramétrica de un robot es el procedimiento que permite medir los coeficientes de la ecuación diferencial o modelo dinámico del robot con el objetivo de ser utilizados en la predicción de su movimiento.

En sistemas de control de robots es importante conocer la cantidad de energía mecánica necesaria para producir una trayectoria deseada, sin embargo, sin los parámetros dinámicos es complicado determinar la energía con precisión.

La implementación de sistemas robóticos es un proceso que se beneficia de conocer los parámetros dinámicos debido a que los experimentos de simulación con la caracterización del robot implican una emulación de cómo se debe comportar el robot.

Esto se ve reflejado en el tiempo de diseño que toma un sistema robótico ya que una computadora puede determinar el movimiento del robot con la simulación varios millones de veces más rápido que el sistema real.

Las limitaciones físicas y computacionales de los sistemas de medición de movimiento evitan identificar parámetros dinámicos de un robot fácilmente. Además, algunos robots tienen un comportamiento determinista hipersensible a las condiciones iniciales que impide identificar sus parámetros con algoritmos basados en ecuaciones diferenciales continuas en el tiempo.

Para la medición de movimiento se utiliza convencionalmente estimaciones basadas en mediciones parciales de parte de este, sin embargo, no se ha encontrado un estudio aún que aborde la fiabilidad de estas estimaciones en identificación paramétrica.

2 Trabajos relacionados

En trabajos de identificación paramétrica se ha encontrado que los sensores de movimiento son considerados como muestreadores discretos analógicos (Ogata, 1996). Del espectro de frecuencia del modelo de sensor considerado se recupera la señal original debajo del ancho de banda del muestreo. Sin embargo, no se ha encontrado aún un trabajo de identificación paramétrica que considere los sensores como muestreadores asíncronos digitales. En la figura 2.1 está la distribución de los actuadores que utilizan los robots.

En (Xiao et. al., 2007) y en (Swevers et. al., 2007) se muestran los métodos de diferenciación numérica y por espectro respectivamente utilizados en la mayoría de los trabajos consultados para estimación de velocidad y aceleración. Los actuadores son considerados como sistemas de transferencia de energía muestreados analógicos, sin embargo, al igual que los sensores no son modelados como muestreadores asíncronos digitales.

Para el análisis de ruido de las señales de los robots se ha encontrado que se asume un modelo de ruido aditivo en las señales de posición y estimaciones de velocidad, aceleración y torque. En (Chávez Olivares et. al., 2017) se estudia la identificación de un robot manipulador y se considera que las estimaciones de velocidad y aceleración contienen ruido

numérico. En la figura 2.2 está la distribución del análisis de ruido de los trabajos consultados.

En (Faa Jeng et. al., 2008) se implementa un filtro digital para un encoder de posición. Por otra parte, en (Huynh et. al., 2020) consideran que el encoder de posición utilizado en su robot contiene ruido aditivo de distribución normal, aunque las gráficas de posición no lo muestran.

El filtro de Kalman (Kalman, 1960), (Ding, Zhao, & Wu, 2014), (Soh, & Wu, 2017) contiene un estudio detallado para sistemas dinámicos con ruido aditivo de distribución normal en sus señales, sin embargo, las señales de las variables de estado se consideran continuas discretas.

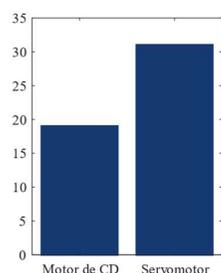


Fig. 2.1: Distribución de los actuadores utilizados en el trabajo relacionado de identificación paramétrica.

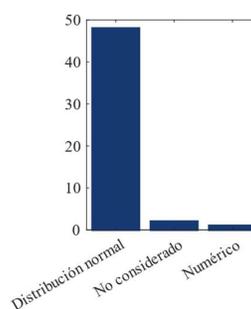


Fig. 2.2: Distribución de modelos de ruido considerados en el trabajo relacionado.

El modelo dinámico de los robots estudiados en los trabajos de identificación paramétrica es desarrollado por la metodología de Euler-Lagrange, Newton-Euler y por aproximaciones lineales (Taylor, 2018). Los modelos presentados son resumidos la representación vectorial mostrada en la sección 3. En la figura 2.5 se aprecia la distribución de metodologías para los modelos dinámicos.

En los trabajos consultados, no se analiza previamente el comportamiento del modelo dinámico. En (Qin, Baron, & Birglen, 2009) se estudia un robot de dos grados de libertad para identificación paramétrica el cual exhibe un comportamiento caótico en función de las condiciones iniciales. Los modelos de fricción utilizados en la mayoría de trabajos son la fricción viscosa, de coulomb y el modelo de la curva de Stribeck (You et. al., 2012) (ver figura 2.6).

Los algoritmos de identificación paramétrica más utilizados son mínimos cuadrados (Pohlman & Leitner, 2003), máxima verosimilitud (Swevers J et. al., 1997), (Devore, 2016) y redes neuronales (Berzal, 2018). En (Hu & Xiong, 2018) se utiliza una red neuronal multicapa para estimar el torque de un robot sin identificar los parámetros dinámicos (ver figura 2.3).

En (Wu & Jahanshahi, 2019) se utiliza una red neuronal convolucional para estimar el movimiento de tres sistemas dinámicos. En (Agand, Shoorehdeli, & Sedigh, 2017) se utiliza una red neuronal recurrente para compensar el torque de un controlador PID en un

helicóptero colocado en una base firme, sus resultados de más de 100 segundos muestran que el helicóptero se estabiliza en la posición deseada. En (Wang et. al., 2020) se utiliza una red neuronal LSTM en conjunto con el algoritmo de mínimos cuadrados para la identificación de parámetros dinámicos de un robot, sin embargo, se necesita de la optimización de una trayectoria de movimiento para que el robot permita identificar parámetros dinámicos. En (Liu, et al., 2020) se utiliza una red neuronal artificial en conjunto con el modelo dinámico de un robot para estimar el torque. En (Wu & Jahanshahi, 2019) y (Su, et al., 2020) se utilizan redes neuronales convolucionales para estimar el movimiento de robots manipuladores y dirigirlos con controles externos respectivamente.

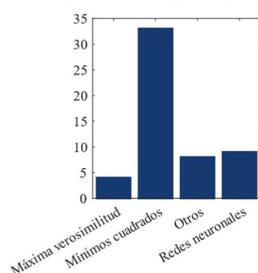


Fig. 2.3: Distribución de algoritmos de identificación paramétrica utilizados.

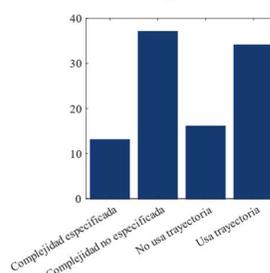


Fig. 2.4: Distribución de uso y características de trayectoria parametrizada para identificación.

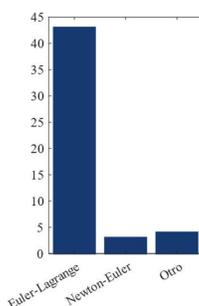


Fig. 2.5: Distribución de metodologías para determinar el modelo dinámico de robots.

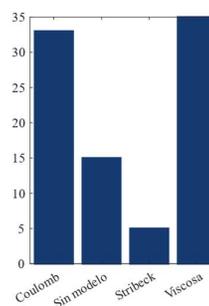


Fig. 2.6: Distribución de modelos de fricción utilizados.

En la gran mayoría de trabajos consultados se utiliza la optimización de una trayectoria de movimiento con el fin de observar los parámetros dinámicos. En la sección dos se muestra las ecuaciones de la trayectoria comúnmente utilizada. La trayectoria se optimiza con algoritmos de inteligencia artificial y las justificaciones halladas son relacionadas a reducir el ruido de las señales de posición, excitar los parámetros dinámicos, evitar la frecuencia natural del robot y evitar diferenciación numérica (ver figura 2.4). La validación

de los parámetros dinámicos de los trabajos consultados nunca se realiza a bajas, medias y altas velocidades todavía. Tampoco se compara la aportación de cada parámetro dinámico al torque y su relación con los demás todavía. En la figura 2.7 se observa como se validan los parámetros dinámicos encontrados de los robots.

Las conclusiones del estado del arte son que no se ha hallado un algoritmo de estimación paramétrica con una red neuronal artificial que estime parámetros dinámicos sin la optimización de una trayectoria. Se ha encontrado que los modelos de señales considerados evitan mediciones completas del movimiento. Se da como aceptado y adecuado las estimaciones de velocidad y aceleración sin un análisis previo, además, son consideradas como mediciones directas de velocidad sin errores y continuas en los algoritmos de identificación. No se analiza el comportamiento de los robots descritos y las limitaciones de los sistemas de medición utilizados. En las figuras 2.1 a 2.7 se aprecia el análisis de los cincuenta trabajos consultados en el estado del arte.

Este trabajo está organizado en la siguiente forma: la sección 3 muestra los preliminares matemáticos del modelo dinámico de un robot de N grados y la teoría de las redes neuronales artificiales tipo *FeedForward*, la sección 4 describe la metodología propuesta, la sección 5 muestra los primeros avances logrados y finalmente la sección 6 son las conclusiones de este trabajo.



Fig. 2.7: Validación de algoritmos de identificación paramétrica en el trabajo relacionado.

3 Preliminares

El modelo dinámico de un robot está determinado por un conjunto de ecuaciones diferenciales continuas. En la ecuación 3.1 el vector x contiene las variables de estado, la función f es el modelo dinámico, β son los parámetros dinámicos, t es el tiempo y el vector u es la entrada de energía. En este modelo se considera que el vector de estados que contiene

la posición y la velocidad de un robot pertenecen al conjunto de los números reales, además, el vector de entrada pertenece al mismo conjunto.

$$\frac{d}{dt}\mathbf{x} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \boldsymbol{\beta}, t) \in \mathbb{R}^{2N} \quad (3.1)$$

Los parámetros dinámicos $\boldsymbol{\beta}$ están asociadas a los parámetros de inercia, de fricción viscosa, de coulomb y a la gravedad. La ecuación (3.1) tiene una única solución para cada condición inicial \mathbf{x}_0 por lo que es posible solucionar la ecuación por medio de integración:

$$\mathbf{x} = \mathbf{x}_0 + \int_{t_0}^t \mathbf{f}(\mathbf{x}, \mathbf{u}, \boldsymbol{\beta}, s) ds$$

Para identificar los parámetros dinámicos $\boldsymbol{\beta}$ se modifica la ecuación (3.1) para que las funciones involucradas en la función \mathbf{f} sean proporcionales a los parámetros; el modelo se reescribe como $\frac{d}{dt}\mathbf{x} = A(\mathbf{x}, \mathbf{u}, t)\boldsymbol{\beta}$ donde A es la matriz de observaciones. En implementaciones reales la medición del vector \mathbf{x} es parcial para algunos componentes y el resto es estimado con la serie de Taylor expandida hasta el primer término:

$$y = ax, \quad \dot{x} = y \quad (3.2)$$

$$y_{k+1} = y_k + hax_k + \frac{h^2}{2!}a^2x_k + \dots + \frac{h^n}{n!}a^nx_k + \dots$$

Considerando la cuantificación de la señal y es descrita por $r\text{floor}(\frac{y}{r})$ donde r es la resolución, se observa que el estimador de la ecuación (3.2) contiene un error mayor al presentado al ignorar los términos $\frac{h^2}{2!}a^2x_k + \dots + \frac{h^n}{n!}a^nx_k + \dots$ de la serie. Si se deriva directamente la señal cuantificada de y , el resultado es un tren de impulsos; $\frac{dy}{dt} = \lim_{\Delta t \rightarrow 0} [\frac{r}{\Delta t} \sum_{m=0}^{\infty} \delta(t_m)]$ lo que ya no representa la información del movimiento esperada. Para el caso de los robots, la señal y es la posición y sus dos derivadas posteriores son la velocidad y aceleración. Debido a las limitaciones de medir las derivadas de la posición con la precisión suficiente para ser usadas en algoritmos de identificación deterministas, se propone utilizar un algoritmo con una red neuronal artificial que estime los parámetros dinámicos sin necesidad de estimaciones precisas de velocidad y aceleración. En la sección 4 se aborda la metodología propuesta para identificar los parámetros dinámicos de la ecuación (3.1).

4 Metodología propuesta

Se propone utilizar un algoritmo de identificación con una red neuronal artificial que procese la información del movimiento del robot (solo la que proviene de mediciones de

sensores y de los actuadores) y que devuelva los parámetros dinámicos del robot. En la figura 4.1 se observa la metodología propuesta dividida en tres fases; diseño preliminar, desarrollo del algoritmo e implementación.

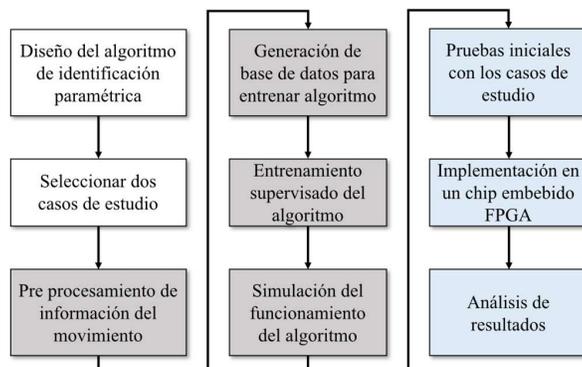


Fig. 4.1: Diagrama de la metodología propuesta.

En la primera fase se crea el diseño preliminar del algoritmo de identificación que incluye la arquitectura de la red neuronal artificial y los pasos preliminares para identificar los parámetros dinámicos. Se seleccionan dos robots manipuladores y sus modelos dinámicos que servirán para crear las bases de datos de simulación para entrenar el algoritmo propuesto. En la segunda fase se diseña el preprocesamiento de la información del movimiento del robot que incluye la forma en que es introducida al algoritmo de identificación.

Para la generación de la base de datos se utiliza la simulación del modelo dinámico de los dos robots manipuladores seleccionados. Debido a la cantidad de parámetros y variables involucradas en un modelo de un robot, la simulación se implementa en c++ donde la ejecución es rápida lo que permite generar miles de datos en poco tiempo. Para entrenar al algoritmo se utiliza el gradiente descendente del error donde la función de costo es la función cuadrática. La base de datos es dividida en dos partes; una para entrenar la red neuronal artificial del algoritmo y otra para validar el entrenamiento de la red. Para corroborar el funcionamiento del algoritmo se simula con datos generados de una simulación con parámetros dinámicos aleatorios.

La tercera fase consiste en realizar pruebas con los dos casos de estudio para observar el funcionamiento del algoritmo y retroalimentar el diseño de este. En esta fase si las pruebas son satisfactorias se implementará el algoritmo en un sistema embebido FGPA con un procesador tipo soft-core. En la sección cuatro se muestra la primera versión del algoritmo propuesto junto con algunas funciones necesarias para identificar los parámetros dinámicos.

5 Avances logrados

En el algoritmo 1 se muestra la primera versión del algoritmo de identificación paramétrica. Los datos de entrada son el modelo dinámico, la posición del robot y el torque aplicado donde k indica que son mediciones obtenidas por el muestreo de los sensores y actuadores. En la línea 2 se unen las mediciones con las estimaciones para conformar el conjunto Y_m . Con la función **submuestreo** de la línea 3 se reduce el número de muestras a una longitud n . En esta función se pretende que la longitud de entrada y de salida no sean múltiplos; se debe submuestrear las señales a una cantidad $n < n_k$ donde n_k es la longitud de los datos obtenidos.

En la línea 4 se inicializan los parámetros dinámicos p de longitud n_p en forma aleatoria con la función **inicialización**. Dentro de esta función se debe normalizar la inicialización para que no se obtengan parámetros iniciales negativos. En la línea 5 se procesa la información con la función **preprocesamiento** la cual devuelve en la variable M los datos preprocesados para ser introducidos a la red neuronal.

El ciclo **while** de la línea 7 ejecuta la red neuronal para hallar la cantidad que es necesaria para hallar los parámetros dinámicos, ejecuta la métrica de evaluación y genera un conjunto de datos nuevo en caso de que la métrica almacenada en v no satisfaga el criterio de $v \geq 0.9$. Finalmente, el algoritmo cuando se satisface la métrica de evaluación devuelve los parámetros dinámicos.

Algoritmo 1 Identificación Paramétrica de un robot manipulador

Entrada: Señales de posición $q(k)$ y torque $\tau(k)$ del robot, Modelo dinámico $A(q, \tau)\beta$

Salida: Parámetros dinámicos del robot.

```
1:  $Y \leftarrow q(k)$ 
2:  $Y_m \leftarrow Y \cup \tau(k) \cup Y_s$ 
3:  $y_m \leftarrow \text{submuestreo}(Y_m, n)$ 
4:  $p \leftarrow \text{inicialización}(n_p)$ 
5:  $M \leftarrow \text{preprocesamiento}(y_m, p)$ 
6:  $v \leftarrow 0$ 
7: while  $v < 0.9$  do
8:      $p \leftarrow \text{red neuronal}(M, p) + p$ 
9:      $v \leftarrow \text{métrica}(p, q, \tau, f)$ 
10:     $M \leftarrow \text{preprocesamiento}(y_m, p)$ 
11: devolver  $p$ 
```

La red neuronal utilizada continua en fase de desarrollo con el fin de seleccionar la estructura que mejor se adapte para identificar los parámetros dinámicos. En la figura 5.1 se encuentra la metodología para entrenar la red neuronal artificial.

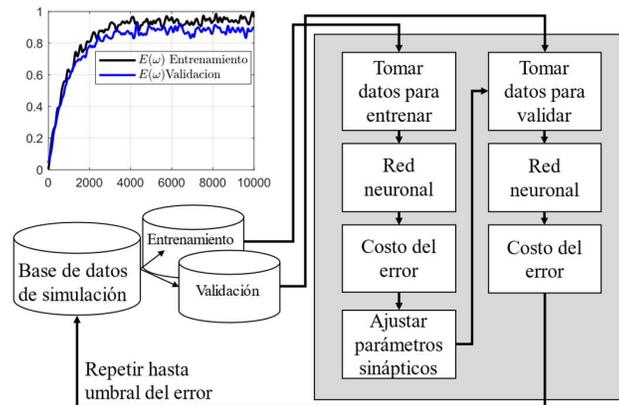


Fig. 5.1: Entrenamiento de la red neuronal artificial.

6 Conclusiones

La revisión sistemática del estado del arte ha mostrado las áreas de oportunidad encontradas para la identificación paramétrica de robots manipuladores. El área de oportunidad más importante es que los algoritmos convencionales son deterministas, pero trabajan con datos incompletos que producen demoras en el proceso. En los avances logrados está el diseño del algoritmo de identificación que no requiere de la optimización de una trayectoria de movimiento para el robot manipulador; solo es necesario utilizar una señal senoidal con un controlador proporcional derivativo $\tau_i = k_i f(q_{d_i} - q_i) - k_v f(\hat{q}_i)$.

El trabajo futuro es implementar el algoritmo propuesto y entrenar la red neuronal para los dos casos de estudio seleccionados. Para lograr este paso, es necesario que el modelo dinámico de los robots modele la respuesta de los robots reales con una precisión elevada.

Referencias

- Agand, P., Shoorehdeli, A. M., & Sedigh, K. A. (2017). "Adaptive recurrent neural network with Lyapunov stability learning rules for robot dynamic terms identification", *Engineering Applications of Artificial Intelligence*, vol. 11, pp. 1-11.
- Berzal, F. (2018). "Redes Neuronales y Deep Learning", *Granada, España: Ikor*.
- Chávez Olivares, C., Reyes Cortés, F., Galván Gonzáles, E., Mendoza Gutiérrez, M., & Bonilla Guitierrez, I. (2017). "Experimental Evaluation of Parameter Identification Schemes on an

- Anthropomorphic Direct Drive Robot”, *International Journal of Advanced Robotic Systems*, vol. 9, num. 5.
- Devore, J. L. (2016). “Probability and Statistics for Engineering and The Sciences”, Bosto, MA USA: *Cengage Learning*.
- Ding, Y., Zhao, B. Y., & Wu, B. (2014). “Structural System Identification with Extended Kalman Filter and Orthogonal Decomposition of Excitation”, *Mathematical Problems in Engineering*.
- Faa Jeng, L., Li Tao, T., Cheng Yan, C., & Chinh Kai, C. (2008). “Robust RBFN Control for Linear InductionMotor Drive Using FPGA”, *IEEE Transactions on Power Electronics*, vol. 23, num. 4, pp. 2170-2180.
- Hu, J., & Xiong, R. (2018). “Contact Force Estimation for Robot Manipulator Using Semiparametric Model and Disturbance Kalman Filter”, *IEEE Transactions on Industrial Electronics*, vol. 65, num. 4, pp. 3365 - 3375.
- Huynh, H. N., Assadi, H., Rivière Lorphèvre, E., Verlinden, O., & Ahmadi, K. (2020). “Modelling the dynamics of industrial robots for milling operations”, *Robotics and Computer-Integrated Manufacturing*, vol. 61.
- Kalman, R. (1960). “A New Approach to Linear Filtering and Prediction Problems”, *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, num. D, pp. 35-45.
- Liu, N., Li, L., Hao, B., Yang, L., Hu, T., Xue, T., . . . Shao, X. (2020). “Semiparametric Deep Learning Manipulator Inverse Dynamics Modeling Method for Smart City and Industrial Applications”, *Complexity*, pp. 1-11.
- Pohlman, J. T., & Leitner, D. W. (2003). “A Comparison of Ordinary Least Squares and Logistic Regression”, *The Ohio Journal of Science*, vol. 103, num. 5, pp. 118-125.
- Qin, Z., Baron, L., & Birglen, L. (2009). “A new approach to the dynamic parameter identification of robotic manipulators”, *Robotica*, vol. 28, num. 4, pp. 539-547.
- Soh, J., & Wu, X. (2017). “An FPGA-Based Unscented Kalman Filter for System-On-Chip Applications”, *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, num. 4, pp. 447 - 451.
- Su, H., Qi, W., Yang, C., Sandoval, J., Ferrigno, G., & De Momi, E. (2020). “Deep Neural Network Approach in Robot Tool Dynamics Identification for Bilateral Teleoperation”, *IEEE Robotics and Automation Letters*, vol. 5, num. 2, pp. 2943 - 2949.
- Swevers, J., Ganseman, C., Tükel, B. D., De Schutter, J., & Brussel, H. V. (1997). “Optimal robot excitation and identification”, *IEEE Transactions on Robotics and Automation*, vol. 13, num. 5, pp. 730 - 740.
- Swevers, J., Verdonck, W., & De Schutter, J. (2007). “Dynamic Model Identification for Industrial Robots”, *IEEE Control Systems Magazine*, vol. 27, num. 5, pp. 58 - 71.
- Taylor, J. R. (2018). “Mecánica clásica”, *Reverte*.
- Wang, S., Shao, X., Yang, L., & Liu, N. (2020). “Deep Learning Aided Dynamic Parameter Identification of 6-DOF Robot Manipulators”, *IEEE Access*, vol. 8, pp. 138102-138116.
- Wu, R. T., & Jahanshahi, M. (2019). “Deep Convolutional Neural Network for Structural Dynamic Response Estimation and System Identification”, *Journal of Engineering Mechanics*, vol. 145, num. 1.
- Xiao, W., Dunford, W. G., Palmer, P. R., & Capel, A. (2007). “Application of Centered Differentiation and Steepest Descent to Maximum Power Point Tracking”, *IEEE Transactions on Industrial Electronics*, vol. 54, num. 5, pp. 2539 - 2549.
- You, B., Qiao, M., Xu, J., & Liu, D. (2012). “Low-Speed Control of Heavy-Load Transfer Robot with Long Telescopic Boom Based on Stribeck Friction Model”, *Mathematical Problems in Engineering*.

Capítulo 5

Construyendo el contenido de un Sistema Tutor Inteligente: un estudio piloto de métricas populares de lecturabilidad para enseñanza de idiomas

Adelina Escobar Acevedo, Josefina Guerrero García

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación

adelina.escobar@alumno.buap.mx, josefina.guerrero@correo.buap.mx

Resumen. La lectura en idioma extranjero representa una oportunidad de exposición al lenguaje. Sin embargo, para un estudiante, si los textos son inadecuados no sólo se omiten los beneficios, sino que la experiencia puede ser contraproducente. La selección de los materiales es una tarea ardua comúnmente sustentada en la experiencia de docentes o casas editoriales. Hacer uso de un sistema tutor inteligente para proporcionar los textos y ejercicios de forma automática acorde al perfil del estudiante puede ser una estrategia de enseñanza para reforzar la comprensión lectora. No basta conocer el nivel de idioma del texto, hace falta hacer análisis más profundos incluyendo métricas de lecturabilidad y otras características textuales. El presente trabajo aplica herramientas para clasificar una muestra de textos extraídos del corpus OneStopEnglish conforme al Marco Común de Referencia Europeo y aplica tres métricas populares de lecturabilidad. Se concluye que la granularidad de la información extraída es un referente para recomendar textos a los estudiantes.

Palabras Clave: Métricas de lecturabilidad, Comprensión de Textos, Sistemas Tutores Inteligentes.

1 Introducción

La lectura es una actividad compleja que involucra variables cognitivas, lingüísticas, textuales, socioculturales, entre otras. Se puede suponer que la facilidad de leer un texto depende de las habilidades del lector; sin embargo, es el autor el que dirige el esfuerzo de redacción para que su obra sea comprendida. La ley del mínimo esfuerzo establece que, en una comunicación, tanto el receptor como el emisor trataran de mantener el mensaje lo menos ambiguo posible para evitar interacciones prolongadas (Zipf, 1949).

Leer en un idioma extranjero implica canalizar los esfuerzos en el idioma meta usando como base el idioma materno (Zarobe y Zarobe, 2019). El Marco Común Europeo de Referencia (CEFR) define competencias para seis niveles de idioma, de A1 a C2 (Instituto

Cervantes, 2002). El número de niveles varía acorde al referente e incluso se usan rangos de puntaje como en el TOEFL e IELTS para el caso del idioma inglés.

Las investigaciones demuestran que existe una relación entre el conocimiento de vocabulario en el idioma extranjero y la comprensión lectora, el estudiante debe conocer alrededor del 90% del vocabulario en el texto (Zarobe y Zarobe, 2019). Esta relación ha propiciado el etiquetado de términos por nivel de idioma considerando el CEFR (Tejada et al., 2015). Por ejemplo, la Universidad de Cambridge tiene publicada una lista de términos en inglés británico y americano clasificados por nivel (Cambridge University Press, 2015).

Hay herramientas que permiten determinar el nivel de los textos, entre ellos Text Inspector (Bax, 2020) y Text Analyzer (roadtogrammar.com, 2021); esta información es útil para los estudiantes y docentes de inglés. Sin embargo, dependiendo de los objetivos lingüísticos es conveniente hacer análisis más profundos que considerar sólo el nivel de idioma.

En el área de lenguaje, la mayoría de los sistemas tutores inteligentes todavía siguen una secuencia rígida en las lecciones. Se busca que en el futuro sean más flexibles y adaptables al usuario, para esto, es relevante extraer la mayor cantidad de información de los textos que sea útil para identificar los más adecuados para cada alumno. El objetivo de este artículo es presentar los resultados de métricas de lecturabilidad populares sobre una muestra del corpus OneStopEnglish. El resto del trabajo está organizado de la siguiente manera: en la sección 2 se aborda la lecturabilidad y simplificación de textos. En la sección 3 se presentan algunas métricas para el análisis de textos. La sección 4 presenta una revisión del trabajo de otros autores. La sección 5 contiene la evaluación realizada en este trabajo sobre una muestra de textos del corpus OneStopEnglish y finalmente se presenta una discusión.

2 Lecturabilidad y simplificación de textos

Una de las estrategias usadas por los docentes para motivar la lectura es proveer material de acuerdo con la edad e interés de los estudiantes. Sin bien parece simple, en ciertas poblaciones la variedad de temas de interés y niveles de idioma de los alumnos es vasta y difusa, como se observa en los grados universitarios. Adicionalmente, la mayoría de los repositorios están integrados con textos ajenos a otros niveles, de forma que cambiar el nivel implica cambiar totalmente de temas y materiales. La importancia de proveer versiones simplificadas del mismo texto depende del objetivo del docente.

La simplificación de textos es importante en aplicaciones dirigidas a personas con bajo nivel de lectura como pueden ser niños, estudiantes iniciados en un idioma extranjero, personas con poco dominio de los temas o con alguna condición que dificulta su nivel lector; por ejemplo, dislexia (Al-Thanyyan y Azmi, 2021). Por esto, las adaptaciones comunes de los Sistemas Tutores Inteligentes consisten en proporcionar textos más sencillos (otro texto de nivel inferior) o versiones simplificadas (el mismo texto de nivel inferior) a estudiantes que no han cumplido con el desempeño esperado.

La simplificación de textos se aborda principalmente mediante simplificación léxica o sintáctica, la primera consiste en reemplazar palabras complejas por sinónimos más simples haciendo uso de recursos como diccionarios o las listas de términos etiquetados por nivel. La segunda consiste en simplificar las estructuras complejas preservando el significado original, hasta hace poco las soluciones todavía estaban basadas en reglas utilizando etiquetadores y analizadores sintácticos (Al-Thanyyan y Azmi, 2021).

Un estudiante de idioma extranjero debe exponerse lo más rápido posible a textos originales: periódicos, revistas e incluso obras literarias en el idioma meta. Es usual obtener textos de forma dinámica de la web. Sin embargo, la extracción debe cumplir varios requisitos. Wikipedia es una fuente común y existen trabajos para la creación de corpus automáticos con la versión simplificada de la misma (Wilkens et al., 2018). No obstante, debido a la libertad con la que ha sido creada, Xu et al. (2015) determinan que sólo alrededor del 50% de los textos son verdaderas simplificaciones.

La lecturabilidad, por su parte, trata de describir la relación entre el texto y el esfuerzo del lector para entenderlo, para esto se incluyen factores como la simplicidad del léxico, estilo, cohesión, entorno, entre otros. En la actualidad las investigaciones frecuentemente identifican el nivel de educación necesaria para la comprensión (Liu, 2020). las aplicaciones son diversas: determinar, qué tan legible es un documento médico para el paciente, los documentos legales para los clientes, un texto para un no nativo, entre otras.

Entre las herramientas disponibles en la web se encuentran Coh-Metrix (Graesser et al., 2017) y Textcompare.org (2021), que nos permiten determinar qué tan fácil es leer un texto específico. Particularmente, Coh-Metrix y ha evolucionado a lo largo de los años para proveer 104 métricas de análisis (Graesser et al., 2004). Varios autores han utilizado métricas de Coh-Metrix como apoyo en sus investigaciones, en la siguiente sección se citan algunos.

3 Métricas de lecturabilidad y características textuales

Existe una gran variedad de características que pueden extraerse de los textos, entre ellas, las métricas de lecturabilidad se utilizan como predictores de comprensión a nivel oración y para determinar el tiempo necesario para leer el texto. Las métricas más populares, Flesch Kincaid Grade Level y Flesch Reading Ease, consideran características superficiales de los textos como el número de sílabas y la longitud de palabras y oraciones (McNamara et al., 2014).

Flesch Kincaid Grade Level

Es una métrica que predice el grado de estudios que debe tener el lector para comprender el texto, para ser aplicada adecuadamente se requieren documentos con al menos 200 palabras. La fórmula está definida por la ecuación 1.

$$RDFKGL = (.39 \times ASL) + (11.8 \times ASW) - 15.59 \quad (1)$$

Donde:

$$ASL = \left(\frac{\text{número de palabras}}{\text{número de oraciones}} \right)$$

$$ASW = \left(\frac{\text{número de sílabas}}{\text{número de palabras}} \right)$$

Los grados corresponden a la escolaridad americana, si bien comúnmente van de 0 a 12, el número máximo es 18. Un resumen de la interpretación de puntajes se aprecia en la Tabla 1.

Tabla 1. Interpretación de puntajes

Puntaje	Grado escolar	Descripción
5-8	5to a 7mo grado	Fácil de leer
8-9	8vo y 9no grado	Inglés conversacional
10-12	10mo-12vo grado	Ligeramente difícil
13-15	Universitario	Difícil
16-18	Graduado universitario	Muy difícil
18+	Profesional	Extremadamente difícil de leer

Flesch Reading Ease

Es una métrica con rango de 0 a 100 donde los puntajes mayores están directamente relacionados con la facilidad de lectura. La fórmula está representada en la ecuación 2.

$$RDFRE = 206.835 - (1.015 \times ASL) - (84.6 \times ASW) \quad (2)$$

La Tabla 2 muestra la interpretación de los puntajes considerando también el promedio de sílabas por cada 100 palabras y la longitud de las oraciones.

Tabla 2. Interpretación de puntajes

Puntaje	Grado escolar	Descripción	Sílabas en 100 palabras	Longitud promedio de la oración.
90-100	5	Muy fácil	123	8
80-90	6	Fácil	131	11
70-80	7	Relativamente fácil	139	14
60-70	8-9	Estándar	147	17
50-60	10-12	Relativamente difícil	155	21
30-50	Universitario	Difícil	167	25
0-30	Graduado universitario	Muy difícil	192	29

Dada la popularidad de estas métricas, se siguen usando como referencia; sin embargo, actualmente existen métricas más sofisticadas que involucran cohesión, complejidad sintáctica, entre otras características para determinar el nivel de lecturabilidad, es el caso de RDL2.

RDL2

La fórmula, mostrada en la ecuación 3, intenta predecir la legibilidad de los textos para lectores de segundo idioma. Considera la superposición y la frecuencia de vocabulario, así como la similitud sintáctica. La diferencia principal con otras fórmulas es que no sólo considera la dificultad del vocabulario y oraciones sino la cohesión del texto (McNamara et al., 2014).

$$RDL2 = -45.032 + (52.230 \times CRFCWO1) + (61.306 \times SYNSTRUT) + (22.205 \times WRDFRQmc) \quad (3)$$

Donde CRFCWO1 es la proporción media de la superposición de vocabulario de contenido en oraciones adyacentes. SYNSTRUT es la proporción media de similitud sintáctica entre oraciones adyacentes y WRDFRQmc es el valor medio de la frecuencia mínima de palabras de contenido (en logaritmo), usando la base de datos CELEX. Otras métricas relevantes obtenidas por Coh-Metrix se observan en la Tabla 3, la lista completa puede consultarse en la documentación de la herramienta.

Tabla 3. Muestra de índices en Coh-Metrix

Índice	Descripción
LDTTra	Diversidad lingüística, calculada por la tasa de tipos de tokens. Proporción de palabras únicas.
WRDPRO	Índice de pronombres.
DESSL	Número de palabras promedio por oración.
SMTEMP	Cohesión temporal del texto.
SYSNTRUTa	Simplicidad sintáctica basada en el promedio de oraciones adyacentes con estructuras sintácticas similares.
WRDIMGC	Promedio de palabras visualmente descriptivas.

Dado que el tutor que se pretende crear tiene la intención de cubrir niveles A2, B1 y B2 conforme al marco europeo CEFR (de elemental a intermedio), es recomendable trabajar con textos etiquetados por nivel. Recordando que un texto puede ser categorizado en diferente número de niveles dependiendo del referente. Incluso en algunos repositorios las clases pueden ser más difusas al estar relacionadas con el plan de estudios de casas editoriales y la libertad en el orden de los temas a enseñar. Por ello, además de buscar un corpus etiquetado, se hizo un análisis con métricas de lecturabilidad.

4 Revisión del estado del arte

Zhang (2016) utilizó Coh-Metrix para determinar cuál de dos libros de texto populares en China es más adecuado para estudiantes universitarios. Para esto, seleccionó veintiún medidas, incluyendo RDL2, para determinar que uno de los libros es más básico en los textos que presenta y por lo tanto servirá sólo para estudiantes de menor grado.

De forma similar, un estudio en Chile hace un análisis de tres libros de texto de décimo, décimo primer y décimo segundo grados utilizados para la enseñanza del inglés. Utiliza las tres métricas de lecturabilidad que posee Coh-Metrix pero también hace un análisis de las preguntas que aparecen relacionadas a los textos. Concluye que debido a que los libros son de diferentes autores y editoriales, sustentado con las métricas aplicadas, el programa no es progresivo en realidad (Cárcamo Morales, 2020).

Nahatame (2020) hace un estudio de seguimiento de movimientos oculares para determinar cuál de las métricas de lecturabilidad es mejor predictor para determinar la dificultad del texto ante los estudiantes. Los experimentos se realizaron con cuarenta estudiantes universitarios japoneses concluyendo que tanto las métricas tradicionales (RDFKGL) como las más sofisticadas (RDL2) son confiables para predecir las fijaciones; sin embargo, la métrica RDL2 es mejor predictor. Los estudios de movimientos oculares indican que, mientras más complejo es el texto, se incrementa el número de fijaciones y su duración.

No con lectura, sino con los textos escritos por los alumnos y utilizando 19 métricas de Coh-Metrix, Li et al. (2018) hace un análisis de producciones de 293 estudiantes de secundaria. Entre las características incluye un par de medidas de lecturabilidad, incluyendo la de RDFKGL y RDL2. En este caso, la primera resulta útil como discriminante, la segunda no. El trabajo está enfocado en determinar la complejidad de las explicaciones de los alumnos.

Como indican los autores, no es conveniente considerar una única métrica para caracterizar un texto, dependiendo del objetivo deberían considerarse varias. En el caso de la facilidad de lectura además se debe tener en cuenta que el grado de narratividad, cohesión, variedad de vocabulario estén alineados con los objetivos perseguidos.

5 Evaluación

En este trabajo se evaluó una pequeña muestra de textos extraídos del corpus OneStopEnglish (Vajjala y Lucic, 2018). El corpus original consta de 189 textos en inglés, cada uno con tres versiones por nivel: avanzado, intermedio y elemental. Los textos fueron simplificados manualmente por docentes de inglés, tratando de conservar número de palabras y oraciones en lo posible. A la fecha es el único repositorio etiquetado que se ha encontrado libre para experimentación. El corpus fue creado con un doble objetivo, por una parte, apoyar la creación de modelos de simplificación y por otra la evaluación de lecturabilidad.

La página de donde fueron extraídos originalmente, está dedicada a la enseñanza de idiomas y los textos eran acompañados por otros materiales incluyendo ejercicios, desafortunadamente fueron eliminados probablemente por condiciones de distribución y derechos de autor, ya que la fuente tiene políticas claras con respecto al uso de materiales.

Si bien el corpus tiene los textos clasificados en tres clases: elemental, intermedio y avanzado. Se utilizó Text Analyzer para determinar el nivel de idioma con respecto al CEFR e IELTS (Figura 1).

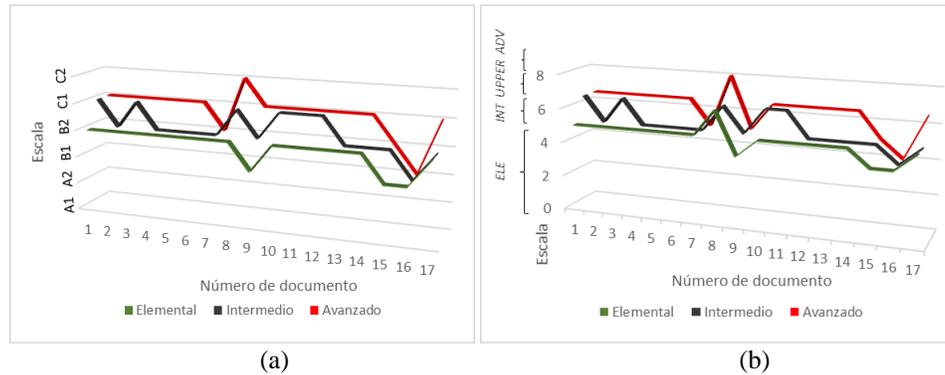


Figura 1. Niveles a) CEFR b) IELTS de los primeros 17 documentos del corpus OneStopEnglish

Se aprecia que las etiquetas originales no están alineadas con respecto a CEFR e IELTS. Hay que recordar que los textos originales fueron simplificados manualmente y la segunda simplificación fue etiquetada como elemental, aunque la herramienta los ubica en nivel intermedio. Comparando ambas gráficas se aprecia que, aunque la clasificación es similar, hay documentos que se asignaron a otro nivel, por ejemplo, el documento 8 es *intermediate* en CEFR, pero es *upper intermediate* en IELTS, esto es por tratarse de distintos referentes.

El segundo paso implica determinar las métricas de lecturabilidad, se usaron las tres que proporciona Coh-matrix con las tres versiones de los textos, Figura 2. Otros autores mencionan que las 3 métricas son buenos predictores estableciendo una ligera preferencia por RDL2.

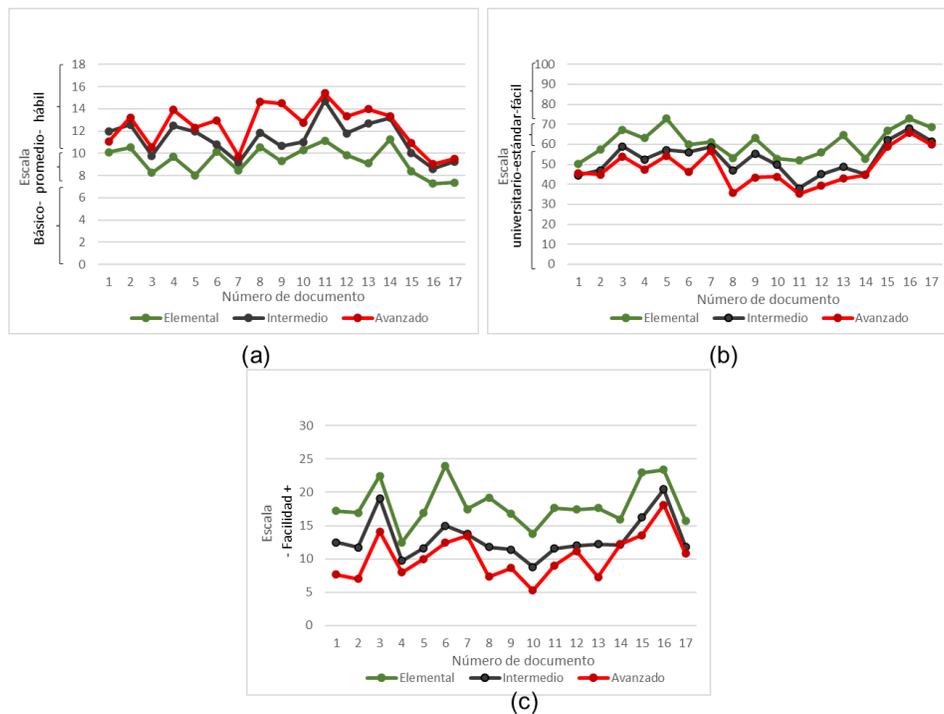


Figura 2. Métricas de lecturabilidad a) RDFKGL, b) RDFRE, c) RDL2

Se seleccionaron 3 documentos para ver el detalle. El documento 9 es una muestra del comportamiento esperado en donde se observa la simplificación progresiva de forma homogénea en todas las columnas de la tabla 4.

Tabla 4 Detalle de resultados de métricas

Número de documento	Etiquetado original	CEFR	IELTS	RDFKGL (5-18)	RDFRE (0-100)	RDL2 (0-30)*
9	Avanzado	C1	5-6	14.48	43.37	8.64
	Intermedio	B2	5-6	10.66	55.30	11.39
	Elemental	B1	4-5	9.27	63.05	16.77
7	Avanzado	B2	5-6	9.70	56.73	13.48
	Intermedio	B2	5-6	9.17	58.59	13.75
	Elemental	B2	5-6	8.44	61.08	17.46
1	Avanzado	C1	6.5-8	11.05	45.55	7.65
	Intermedio	C1	6.5-8	11.95	44.51	12.46
	Elemental	B2	5-6	10.11	50.31	17.21

* Rango en el corpus completo

En el documento 7 sólo se pueden observar las diferencias al aplicar métricas porque las variaciones son pequeñas, por ejemplo, en RDFKGL es de 0.53 puntos entre la versión avanzada y la intermedia. Por otra parte, observe el documento 1, en las simplificaciones los puntajes reflejados no son consecutivos para RDFKGL y RDFRE. Recordar estas dos métricas usan información superficial del texto como la longitud de las palabras. RDL2 utiliza características adicionales, se observa que si existe relación conforme al etiquetado original y es la única consistente en todos los documentos.

6 Discusión

Con este trabajo se identifica que el etiquetado inicial por niveles de lectura es insuficiente, se requiere un análisis del texto más profundo que al menos incluya métricas de lecturabilidad. Otros datos importantes son: narratividad, uso de pronombres, variedad léxica entre otras características para indicar si el texto se acopla a los objetivos lingüísticos perseguidos, antes de presentarlo a los estudiantes.

Otros autores ya han probado experimentalmente con alumnos la efectividad de hacer análisis más profundos de los textos para predecir la dificultad y el tiempo de la lectura. Este proceso es útil para que un sistema de recomendación sea capaz de tomar decisiones informadas sobre el texto más orientados para el usuario. Lo que es particularmente importante en la formación del módulo dominio para los Sistemas Tutores Inteligentes en el área de lenguaje.

Como trabajo futuro se considera la clasificación de los ejercicios y preguntas relacionadas con los textos. Si bien identificar las características de los textos es importante, plantear los ejercicios adecuados para que el estudiante alcance los objetivos lingüísticos esperados es indispensable.

Referencias

- Al-Thanyyan, S. S., y Azmi, A. M. (2021). "Automated Text Simplification: A survey". *ACM Computing Surveys*, 54(2), 1–36.
- Bax, S. (2020). *Text Inspector*. Recuperado de <https://textinspector.com/>
- Cambridge University Press. (2015). *English Profile, The CEFR for English*. Recuperado de <https://www.englishprofile.org/wordlists/evp>
- Cárcamo Morales, B. (2020). "Readability and types of questions in Chilean EFL high school textbooks". *TESOL Journal*, 11(2), 1–15.
- Graesser, A. C., McNamara, D. S., Louwerse, M. M., y Cai, Z. (2004). "Coh-Metrix : Analysis of text on cohesion and language". *Behavior Research Methods, Instruments, & Computers*, 36(2), 193–202.
- Graesser, A. C., McNamara, D. S., and Louwerse, M. M. (2017). *Coh-Metrix*. Recuperado de <http://cohmetrix.com/>
- Instituto Cervantes (2002). "Marco Común Europeo de Referencia para las Lenguas: aprendizaje,

- enseñanza, evaluación”. *Instituto Cervantes*.
- Li, H., Gobert, J., Dickler, R., y Morad, N. (2018). “Students’ Academic Language Use When Constructing Scientific Explanations in an Intelligent Tutoring System”. *Conference on Artificial Intelligence in Education*, 267–281. https://doi.org/10.1007/978-3-319-93843-1_20
- McNamara, D. S., Graesser, A. C., McCarthy, P. M., y Cai, Z. (2014). “Coh-Metrix Measures of Text Readability and Easability”. En *Automated Evaluation of Text and Discourse with Coh-Metrix* (pp. 78–95). Cambridge University Press.
- Nahatame, S. (2020). “Text readability and comprehension processes during L2 reading: A computational and eye-tracking investigation”. *Conference of the American Association for Applied Linguistics (AAAL)*.
- Roadtogrammar.com. (2021). *Text Analyzer*. Recuperado de <http://www.roadtogrammar.com/textanalysis/>
- Tejada, M. Á. Z., Gallardo, C. N., Ferradá, M. C. M., y López, M. I. C. (2015). “Building a Corpus of 2L English for Automatic Assessment: The CLEC Corpus”. *Procedia - Social and Behavioral Sciences*, 198(Cilc), 515–525. <https://doi.org/10.1016/j.sbspro.2015.07.474>
- Textcompare.org. (2021). *Textcompare.org*. Recuperado de <https://www.textcompare.org/readability/>
- Vajjala, S., y Lucic, I. (2018). “OneStopEnglish corpus : A new corpus for automatic readability assessment and text simplification”. *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 297–304. <https://www.aclweb.org/anthology/W18-0535/>
- Wilkens, R., Zilio, L., y Fairon, C. (2018). “SW4ALL : a CEFR-Classified and Aligned Corpus for Language Learning”. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 365–370.
- Xu, W., Callison-Burch, C., y Napoles, C. (2015). “Problems in Current Text Simplification Research: New Data Can Help”. *Transactions of the Association for Computational Linguistics*, 3, 283–297.
- Zarobe, Y. R. De, y Zarobe, L. R. De (Eds.). (2019). “La lectura en lengua e xtranjera”. Ediciones Octaedro.
- Zhang, R. (2016). “A Coh-Metrix Analysis of Two Textbooks: Successful English for Vocational Colleges and Vocational College English (An Integrated Skills Course)”. *US-China Foreign Language*, 14(5), 351–356.
- Zipf, G. K. (1949). “Introduction and Orientation”, *Human behavior and the principle of least effort: an introduction to human ecology*. Addison-Wesley Press.

Capítulo 6

Generación de una plataforma que asista en la creación de redes neuronales para su simulación e implementación en un sistema embebido

Daniel Marcelo González Arriaga¹, María Aurora Diozcora Vargas Treviño², Sergio Vergara Limon², Josefina Guerrero García¹

¹ Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación

² Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Electrónica

dm.gar93@gmail.com, auroravargast@hotmail.com,
svergara2@hotmail.com, josefina.guerrero@correo.buap.mx

Resumen. En el presente trabajo se propone una metodología de diseño para obtener una plataforma experimental capaz de asistir en la creación de redes neuronales para su simulación e implementación en un sistema embebido. Esto permitirá reducir el tiempo y esfuerzo necesario para desarrollar redes neuronales.

Palabras Clave: Asistente, redes neuronales, sistema embebido.

1 Introducción

La inteligencia artificial se está volviendo omnipresente muy rápidamente debido a su robusta aplicabilidad en los problemas, particularmente aquellos que no pueden ser resueltos bien por humanos, por ejemplo en la medicina donde los algoritmos se utilizan para identificar sujetos con antecedentes familiares de una enfermedad hereditaria o un riesgo aumentado de una enfermedad crónica o en la evaluación de cambios en el desempeño humano en tales situaciones como rehabilitación (Hamet & Tremblay, 2017).

Otra área de vital importancia donde se utiliza la inteligencia artificial es la agricultura, se han propuesto sistemas expertos para identificar plagas activas y sugerir medidas de control (Mahaman et al., 2003), las enfermedades de los cultivos son otro problema ya que se requiere conocimientos y experiencia para detectar una planta enferma y tomar las medidas necesarias, se propuso un modelo de procesamiento de imágenes junto con un modelo de red neuronal artificial para clasificar las enfermedades (Huang, 2007).

La aplicación de la inteligencia artificial en el campo del transporte está dirigida para superar los desafíos de una creciente demanda de viajes, emisión de contaminantes, problemas de seguridad y degradación ambiental (Abduljabbar et al., 2019). Los problemas de transporte se convierten en un desafío cuando el sistema y el comportamiento de los usuarios es demasiado difícil de modelar y predecir los patrones de viaje. Por lo tanto, la inteligencia artificial se considera una buena opción para el transporte. Un ejemplo de eso incluye transformar los sensores de tráfico en la carretera en un agente inteligente que detecta accidentes automáticamente y predice las condiciones futuras del tráfico (Klügl et al., 2010). Además, hay muchos métodos de IA (inteligencia artificial) que se utilizan en el transporte, como las ANN (Artificial Network Neuronal). Las ANN se pueden utilizar para la planificación de carreteras (Doğan & Akgüngör, 2013), transporte público (Akgüngör & Doğan, 2009; Budalakoti et al., 2009) Detección de incidentes de tráfico (Dia, 2001; Dia & Rose, 1997; Wang et al., 2016); y predecir las condiciones del tráfico (Crutchfield, 2017; W. Huang et al., 2014; Jiang et al., 2016; Król, 2016; Ledoux, 1997; Theofilatos et al., 2016; Wu et al., 2018).

Existen diversas herramientas para la implementación de redes neuronales, en la tabla 1 se muestran algunas de ellas.

Tabla 1 : Descripción de herramientas para redes neuronales.

TensorFlow	Empezó en el 2011 como un proyecto interno de Google llamado “Google Brain” y que se hizo público en el 2017 como un sistema de código abierto de aprendizaje profundo, es decir, de una red neuronal, la cual puede correr en múltiples CPUs y GPUs. Se usa para entrenar redes neuronales que puedan detectar y descifrar patrones y correlaciones análogas a las que vemos en el aprendizaje y razonamiento humano.
Caffe	Esta herramienta fue creada por BAIR (Berkeley Artificial Intelligence Research), en el 2014, y se hizo popular en la investigación académica. En un marco de trabajo de aprendizaje profundo usando redes convolutivas.
ONNX	Esta herramienta significa Open Neural Network exchanged y se anunció apenas en septiembre del 2017. Es un esfuerzo conjunto de Microsoft y Facebook. ONNX es un formato pensado para hacer fácil el intercambio de modelos de aprendizaje profundo entre entornos de esta naturaleza. La iniciativa busca hacer más fácil para los desarrolladores usar múltiples entornos de programación de redes neuronales.
Matlab	Cuenta con toolboxes especializadas para trabajar con aprendizaje automático (machine learning), redes neuronales, aprendizaje profundo, visión artificial y conducción autónoma. Puede utilizar MATLAB Coder a fin de generar código C y C++

	para su red entrenada, lo cual permite simular una red entrenada en hardware de PCs y, posteriormente, desplegar la red en sistemas embebidos.
Weka	Es una plataforma de software para el aprendizaje automático y la minería de datos escrito en Java y desarrollado en la Universidad de Waikato. Es usada principalmente para hacer clasificadores y solo pueden utilizar redes neuronales de tipo perceptrón multicapa

Las herramientas mencionadas sirven para intercambiar modelos de aprendizaje como ONNX, o para realizar el entrenamiento de alguna red neuronal como TensorFlow. Matlab es una herramienta interesante ya que cuenta con diferentes redes neuronales ya implementadas, el principal inconveniente es que no es de código abierto, esto además de generar un costo elevado, no se tiene acceso total a lo que las herramientas del software hacen, y si hay alguna función que no se encuentre implementada no se puede hacer nada.

No se ha encontrado reportado en la bibliografía revisada hasta este momento una herramienta de uso libre que asista en la creación de diversos tipos de redes neuronales, reduciendo el tiempo de creación y que permita implementar dichas redes creadas en un sistema embebido listo para su implementación.

La investigación por realizar es del tipo cuantitativa debido a la naturaleza del problema, en este trabajo se propone una metodología de diseño para obtener una plataforma experimental capaz de asistir en la creación de redes neuronales. Dado que se busca un resultado único se contempla una investigación descriptiva, esto nos permite llegar a un único resultado, y es posible medir la exactitud de este, se definen los objetivos a alcanzar y la metodología para llegar a ellos.

2 Metodología propuesta

La definición de inteligencia artificial varía de autor en autor, dado que es prácticamente imposible alcanzar una definición de IA con la que todo el mundo esté de acuerdo, Keith Downing (Downing, 2015) propone que nos conformemos con una definición mucho más ambigua y menos ambiciosa: hacer lo correcto en el momento adecuado, desde el punto de vista de un observador humano externo. Otro punto de vista es el de Elaine Rich (Berzal, 2018), de la Universidad de Texas en Austin: el estudio de cómo hacer que los ordenadores hagan cosas que, por ahora, los humanos hacemos mejor. Ésta es una definición móvil de la IA, ya que, en cuanto un problema abordado por técnicas de IA se resuelve satisfactoriamente, pasa a quedar automáticamente fuera de su ámbito de actuación. Por otro lado, mantiene el interés de la IA centrandone nuestra atención en aquellos problemas que, actualmente, los seres humanos somos capaces de resolver mejor que las máquinas.

Aprendizaje automático

El aprendizaje automático, o machine learning, proporciona mecanismos mediante los cuales el ordenador es capaz de aprender por sí mismo a resolver un problema. En estos casos, el programador se encarga de diseñar un algoritmo de aprendizaje que resulte adecuado para el problema que se pretende resolver, pero es el ordenador el que resuelve el problema, aprovechando para ello los datos a los que tenga acceso y las heurísticas de aprendizaje incorporadas en el algoritmo de aprendizaje creado por el programador. En cierto modo, el ordenador es capaz de programarse a sí mismo.

En Inteligencia Artificial, el aprendizaje se entiende como un proceso por el cual un ordenador es capaz de mejorar su habilidad en la resolución de un problema a través de la adquisición de conocimiento, conocimiento que obtiene a través de la experiencia (Berzal, 2018). En la figura 1 se muestran las interacciones entre el problema, el proceso de aprendizaje, el conocimiento adquirido y la efectividad observada en la resolución del problema. Adaptado de Pat Langley,

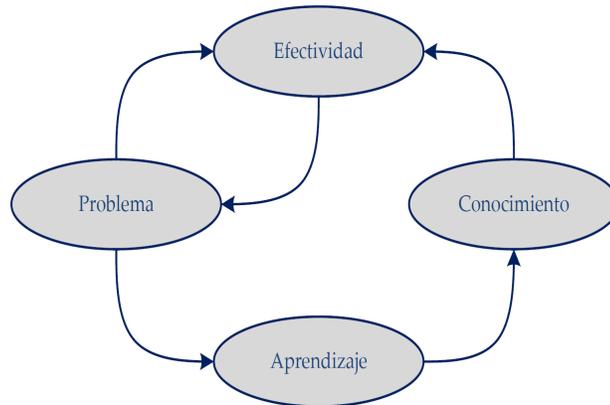


Figura 1 “Elements of Machine Learning”, 1995, donde entorno y rendimiento ocupaban los lugares de problema y efectividad.

Sistemas embebidos

La integración de microprocesadores en equipos y dispositivos de consumo comenzó antes de la aparición de la PC y consume la mayoría de los microprocesadores que se fabrican en la actualidad. De esta manera, los microprocesadores integrados están más arraigados en la vida cotidiana que cualquier otro circuito electrónico que se fabrique. Un automóvil grande puede tener más de 50 microprocesadores (Heath, 2002) que controlan funciones como el motor a través de sistemas de gestión del motor, frenos con frenos antibloqueo electrónicos, transmisión con control de tracción y cajas de cambios controladas electrónicamente, seguridad con sistemas de airbag, elevalunas eléctricos, aire acondicionado, etc.. Con un automóvil bien equipado, casi todos los aspectos tienen alguna forma de control electrónico asociado y, por lo tanto, la necesidad de un microprocesador dentro de un sistema integrado. Una lavadora puede tener un

microcontrolador que contiene los diferentes programas de lavado, proporciona el control de potencia para los distintos motores y bombas e incluso controles.

Microprocesador softcore

Las matrices de puertas programables FIELD (FPGA) son una opción de implementación popular para muchos sistemas integrados. Con su constante aumento en la capacidad lógica, los FPGA ahora pueden implementar complicados sistemas reconfigurables en chip (RSOC) (Altameemi & Bergmann, 2017), incluidos uno o más procesadores generales, además de motores de procesamiento personalizados y específicos de la aplicación. Los sistemas de procesamiento de señales digitales (DSP), incluidos los sistemas de procesamiento de imágenes y video, a menudo utilizan RSOC para la implementación. Las etapas algorítmicas clave se pueden implementar en circuitos de hardware reconfigurables dedicados, mientras que otras tareas se pueden implementar en procesadores softcore de propósito general. Como alternativa a los procesadores de núcleo blando que se implementan en la estructura lógica programable de la FPGA, algunos RSOC también incluyen uno o más bloques de procesadores duros separados de las celdas lógicas programables. La partición de un algoritmo en módulos de software y hardware RSOC es un paso clave en cualquier ciclo de diseño de algoritmos. Cualquier partición dependerá del costo relativo y el rendimiento de las implementaciones de hardware o software de un módulo, incluido el costo del diseño detallado del módulo de hardware. Por lo general, el tiempo de diseño y el costo de implementación de un módulo de software es menor que el de un módulo de hardware personalizado, por lo que generalmente solo los módulos más críticos se implementan en el hardware. La motivación de este artículo es la conjetura de que si el rendimiento del procesador en un RSOC puede mejorarse para aplicaciones DSP, entonces se pueden implementar más módulos en software, simplificando así el diseño del sistema. Los principales fabricantes de FPGA han diseñado o portado procesadores softcore que están optimizados para sus arquitecturas FPGA particulares, como MicroBlaze (Xilinx), Nios (Altera), Cortex (Microsemi) y LatticeMicro32 (Lattice). Además, diseños de procesadores de código abierto que pueden ser utilizados en múltiples plataformas FPGA están disponibles, como LEON, OpenSPARC y OpenRISC. Debido a que estos son procesadores de núcleo suave, los diseños permiten a los usuarios personalizar el diseño del procesador. El procesador se puede seleccionar con o sin cachés, con o sin multiplicadores de hardware, etc. (Altameemi & Bergmann, 2017). En los sistemas integrados convencionales, los microprocesadores se han utilizado durante mucho tiempo para la implementación del sistema DSP. Sin embargo, en los sistemas DSP, las mejores opciones arquitectónicas para un microprocesador difieren de las aplicaciones convencionales de propósito general. Esto ha llevado al desarrollo de una clase especial de microprocesadores denominados Procesadores de señales digitales, a los que aquí nos referiremos como procesadores DSP. Estos procesadores proporcionan soporte arquitectónico explícito para las operaciones DSP de uso común (Kuo et al., 2006). Por lo tanto, tienen un rendimiento de procesamiento superior para aplicaciones DSP en comparación con un procesador de propósito general similar.

Lenguajes de dominio específico

Los lenguajes específicos de dominio (DSL) son lenguajes adaptados a un dominio de aplicación específico. Ofrecen ganancias sustanciales en expresividad y facilidad de uso en comparación con los lenguajes de programación de propósito general en su dominio de aplicación. El desarrollo de DSL es difícil y requiere tanto conocimiento del dominio como experiencia en el desarrollo del lenguaje.

Muchos lenguajes de computadora son dominio específico en lugar de propósito general. Los lenguajes específicos de dominio (DSL) también se denominan orientados a aplicaciones (Sammet, 1972), propósito especial (Bergin & Gibson, 1996), especializado (Bergin & Gibson, 1996), lenguajes específicos para tareas (Nardi, 1993) o aplicaciones (Martin, 1985). Así llamado idiomas de cuarta generación (Martin & Leben, 1986) suelen ser DSL para aplicaciones de bases de datos.

Los DSL intercambian generalidad por expresividad en un dominio limitado. Al proporcionar notaciones y construcciones adaptadas a un dominio de aplicación particular, ofrecen ganancias sustanciales en expresividad y facilidad de uso en comparación con las GPL para el dominio en cuestión, con ganancias correspondientes en productividad y costos de mantenimiento reducidos. Además, al reducir la cantidad de conocimientos de dominio y programación necesarios, las DSL abren su dominio de aplicación a un grupo más grande de desarrolladores de software en comparación con las GPL.

Los DSL son ejecutables de varias formas y en varios grados, incluso hasta el punto de no ser ejecutables. En consecuencia, dependiendo del carácter del DSL en cuestión, los programas correspondientes a menudo se denominan más propiamente especificaciones, definiciones o descripciones. Identificamos algunos puntos en la escala de ejecución de DSL.

3 Avances preliminares

La metodología utilizada en esta investigación es híbrida ya que la investigación por realizar es cuantitativa y descriptiva. La investigación por realizar es del tipo cuantitativa debido a la naturaleza del problema, el resultado esperado es la obtención de una plataforma capaz de asistir en la creación de redes neuronales, por lo tanto se puede medir su desempeño con métricas establecidas. Dado que se busca un resultado único se contempla una investigación descriptiva, esto nos permite llegar a un único resultado, y es posible medir la exactitud de este.

Tanto la investigación, como la experimentación se llevará a cabo en el laboratorio de robótica de la maestría en Ciencias de la Electrónica, debido a que en este laboratorio se cuenta con diferentes robots con los cuales se puede poner a prueba la plataforma a desarrollar.

Las redes neuronales consideradas en este trabajo son: red neuronal multicapa y red neuronal convolucional, esta selección a partir de los trabajos reportados en el estado del arte se encontró que son las redes neuronales que más se utilizan en un sistema embebido.

En la figura 2 se muestra un diagrama general de la estructura propuesta del asistente.

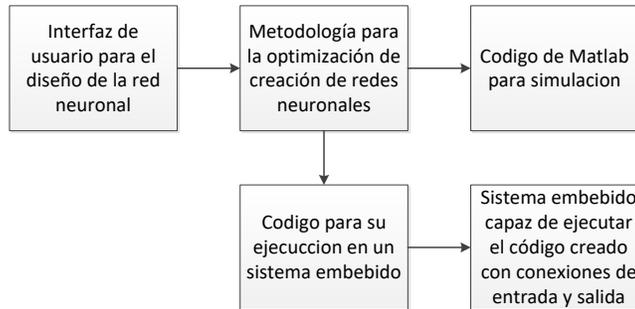


Figura 2 Diagrama a bloques general del sistema.

Se desarrollo con un sistema embebido (resultados en publicación) mostrado en la figura 3, con un FPGA, memoria RAM, memoria ROM, el cual se compone de un microprocesador, que se programa con un lenguaje de alto nivel propio, permitiendo la implementación de los algoritmos desarrollador en tiempo real en el robot.

Los resultados que se obtengan servirán para ajustar la metodología propuesta y así obtener un mejor rendimiento en el desempeño de las redes neuronales y de las estrategias que utiliza la plataforma.

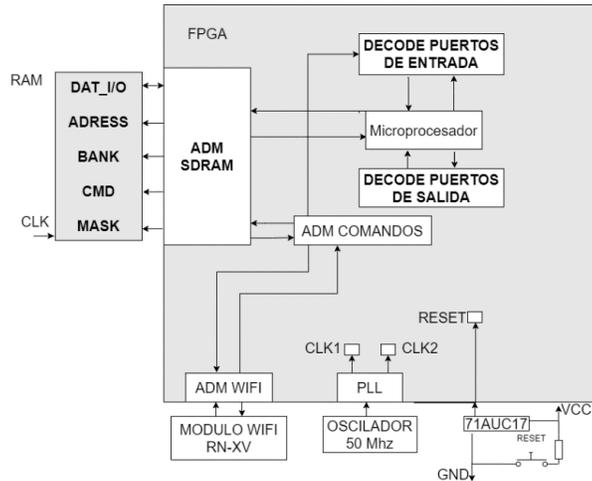


Figura 3 Arquitectura del sistema embebido

Identificación de la estructura de las redes neuronales (diagnostico e identificación): Se analiza la estructura de las redes neuronales que la plataforma será capaz de asistir (red neuronal multicapa y red neuronal convolucional) para identificar los procesos necesarios en su desarrollo, esto permite conocer los procesos repetitivos y desarrollar estrategias para el asistente.

Las unidades de procesamiento se organizan en capas. Hay tres partes normalmente en una red neuronal: una capa de entrada, con unidades que representan los campos de entrada; una o varias capas ocultas; y una capa de salida, con una unidad o unidades que representa el campo o los campos de destino. Las unidades se conectan con fuerzas de conexión variables (o ponderaciones). Los datos de entrada se presentan en la primera capa, y los valores se propagan desde cada neurona hasta cada neurona de la capa siguiente. al final, se envía un resultado desde la capa de salida.

Estrategia de simplificación (definición y evaluación de soluciones): Se desarrollará una estrategia para la simplificación de los datos de entrada que proporciona el usuario, para facilitar el uso de la plataforma así como la selección de los parámetros y variables de cada red neuronal.

Las variables con las que se va a trabajar son: número de entradas de la red neuronal, numero de capas ocultas, tipo de datos, en el caso de la red convolucional, numero de convoluciones a realizar a las imágenes, filtro o *kernel* a aplicar a la imagen.

La estrategia se centra en permitir modificar las variables con las que se va a trabajar cada red neuronal, para que el ajuste en la misma no requiera modificar toda la estructura de la red neuronal. Si se quiere agregar una capa o desconectar alguna conexión entre alguna neurona en específico el software lo realice de manera automática.

Estrategia para implementar una red neuronal en un sistema embebido (Definición y evaluación de soluciones): Conociendo los procesos necesarios para cada red neuronal, se desarrolla una estrategia que permita utilizar el sistema embebido para la implementación de la red neuronal creada.

Se tiene como base un compilador que traduce lenguaje de alto nivel a lenguaje ensamblador del procesador, este compilador necesita ser adecuado para que sea capaz de compilar instrucciones orientadas al uso de redes neuronales adecuando el sistema embebido a los requerimientos matemáticos necesarios.

Diseño de interfaz de uso de la plataforma (ejecución de solución): Al diseñar la interfaz de usuario se tiene que considerar la experiencia de usuario y la usabilidad de la plataforma.

Función: En el hardware, las señales electrónicas activan diferentes situaciones. Los datos se escriben, se leen, se envían, se reciben, se comprueban errores, etc.

En el software, las instrucciones activan el hardware a través de protocolos de enlace de datos, métodos de acceso, etc.

Fácil de usar: Si el producto carece de usabilidad nadie lo deseará. La facilidad con la que alguien utiliza el producto es lo que logrará el objetivo deseado.

Se debe considerar la usabilidad inherente de las interfaces para poder comprender y usar el sistema subyacente. La usabilidad debe ser sencilla si se desea que las personas lo usen ampliamente.

Desarrollo de un lenguaje de dominio específico para la implementación de redes neuronales en un procesador (ejecución de solución): Desarrollando un lenguaje de dominio específico (DSL) se puede tener control de la implementación de la red neuronal en un procesador dentro del sistema embebido, esto permite optimizar su funcionamiento y desarrollar funciones nuevas si es necesario.

Validación del desempeño del sistema desarrollado (seguimiento): La validación del sistema desarrollado se hará en base a métricas encontradas en el estado del arte.

Implementación de una red neuronal en el sistema (seguimiento):

Cuando se ponga a prueba la red neuronal con algún robot algunas de las variables que se podrían utilizar son: posición, velocidad, aceleración, esto puede variar con la aplicación que se le dé a la red neuronal creada con la plataforma.

4 Conclusiones

Después del análisis del estado del arte se concluye que la creación de una plataforma que asista en la creación de redes neuronales que permita no solo la simulación si no también la implementación en un sistema embebido que este optimizado para su uso en aplicación con robots o en cualquier aplicación que requiera de conexiones de entrada y salida y que pueda ser mejorada con una red neuronal.

La metodología propuesta contempla todos los pasos necesarios para que la plataforma cuente con una interfaz de usuario y la plataforma genere el archivo de simulación y de programación para el sistema embebido, es decir el producto final es el software del asistente compatible con la simulación en Matlab y con un sistema embebido basado en un FPGA.

Referencias

- Abduljabbar, R., Dia, H., Liyanage, S., & Bagloee, S. A. (2019). Applications of artificial intelligence in transport: An overview. *Sustainability (Switzerland)*, *11*(1). <https://doi.org/10.3390/su11010189>
- Akg̃ngör, A. P., & Dođan, E. (2009). An artificial intelligent approach to traffic accident estimation: Model development and application. *Transport*, *24*(2), 135–142. <https://doi.org/10.3846/1648-4142.2009.24.135-142>
- Budalakoti, S., Srivastava, A. N., & Otey, M. E. (2009). Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, *39*(1),

101–113. <https://doi.org/10.1109/TSMCC.2008.2007248>

- Crutchfield, J. P. (2017). The Origins of Computational Mechanics: A Brief Intellectual History and Several Clarifications. *ArXiv*, 1–9.
- Dia, H. (2001). An object-oriented neural network approach to short-term traffic forecasting. *European Journal of Operational Research*, *131*(2), 253–261. [https://doi.org/10.1016/S0377-2217\(00\)00125-9](https://doi.org/10.1016/S0377-2217(00)00125-9)
- Dia, H., & Rose, G. (1997). Development and evaluation of neural network freeway incident detection models using field data. *Transportation Research Part C: Emerging Technologies*, *5*(5), 313–331. [https://doi.org/10.1016/S0968-090X\(97\)00016-8](https://doi.org/10.1016/S0968-090X(97)00016-8)
- Doğan, E., & Akgüngör, A. P. (2013). Forecasting highway casualties under the effect of railway development policy in Turkey using artificial neural networks. *Neural Computing and Applications*, *22*(5), 869–877. <https://doi.org/10.1007/s00521-011-0778-0>
- Hamet, P., & Tremblay, J. (2017). Artificial intelligence in medicine. *Metabolism: Clinical and Experimental*, *69*, S36–S40. <https://doi.org/10.1016/j.metabol.2017.01.011>
- Huang, K. Y. (2007). Application of artificial neural network for detecting Phalaenopsis seedling diseases using color and texture features. *Computers and Electronics in Agriculture*, *57*(1), 3–11. <https://doi.org/10.1016/j.compag.2007.01.015>
- Huang, W., Song, G., Hong, H., & Xie, K. (2014). Deep architecture for traffic flow prediction: Deep belief networks with multitask learning. *IEEE Transactions on Intelligent Transportation Systems*, *15*(5), 2191–2201. <https://doi.org/10.1109/ITITS.2014.2311123>
- Jiang, H., Zou, Y., Zhang, S., Tang, J., & Wang, Y. (2016). Short-Term Speed Prediction Using Remote Microwave Sensor Data: Machine Learning versus Statistical Model. *Mathematical Problems in Engineering*, *2016*. <https://doi.org/10.1155/2016/9236156>
- Klügl, F., Bazzan, A. L. C., & Ossowski, S. (2010). Agents in traffic and transportation. *Transportation Research Part C: Emerging Technologies*, *18*(1), 69–70. <https://doi.org/10.1016/j.trc.2009.08.002>
- Król, A. (2016). The Application of the Artificial Intelligence Methods for Planning of the Development of the Transportation Network. *Transportation Research Procedia*, *14*, 4532–4541. <https://doi.org/10.1016/j.trpro.2016.05.376>
- Ledoux, C. (1997). An urban traffic flow model integrating neural networks. *Transportation Research Part C: Emerging Technologies*, *5*(5), 287–300. [https://doi.org/10.1016/S0968-090X\(97\)00015-6](https://doi.org/10.1016/S0968-090X(97)00015-6)
- Mahaman, B. D., Passam, H. C., Sideridis, A. B., & Yialouris, C. P. (2003). DIARES-IPM: A diagnostic advisory rule-based expert system for integrated pest management in Solanaceous crop systems. *Agricultural Systems*, *76*(3), 1119–1135. [https://doi.org/10.1016/S0308-521X\(02\)00187-7](https://doi.org/10.1016/S0308-521X(02)00187-7)
- Theofilatos, A., Yannis, G., Kopelias, P., & Papadimitriou, F. (2016). Predicting Road Accidents: A Rare-events Modeling Approach. *Transportation Research Procedia*, *14*, 3399–3405. <https://doi.org/10.1016/j.trpro.2016.05.293>
- Wang, R., Fan, S., & Work, D. B. (2016). Efficient multiple model particle filtering for joint traffic state estimation and incident detection. *Transportation Research Part C: Emerging Technologies*, *71*, 521–537. <https://doi.org/10.1016/j.trc.2016.08.003>
- Wu, Y., Tan, H., Qin, L., Ran, B., & Jiang, Z. (2018). A hybrid deep learning based traffic flow prediction method and its understanding. *Transportation Research Part C: Emerging Technologies*, *90*(5), 166–180. <https://doi.org/10.1016/j.trc.2018.03.001>

Capítulo 7

Modelado del 3 coloreo de árboles poligonales vía Satisfactibilidad Incremental

Cristina Lopez, Guillermo De Ita

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
Puebla, México

crityna2001@hotmail.com, deita@cs.buap.mx

Resumen. En el presente trabajo se presenta un método para modelar el 3 coloreo de grafos representados por árboles poligonales. La propuesta se basa en especificar lógicamente las restricciones generadas para que se de un 3 coloreo válido sobre grafos poligonales. Para que el proceso total sea de tiempo lineal, las restricciones lógicas son formadas de manera dinámica. Al mismo tiempo, que el grafo se va recorriendo en post orden, se generan instancias del problema de Satisfactibilidad Incremental en tiempo polinomial. Esta propuesta puede extenderse para considerar otras instancias del problema del 3 coloreo que puedan trabajar en tiempo polinomial.

Palabras Clave: Satisfactibilidad Incremental, Coloreo de grafos, Grafos poligonales, 3-coloreo.

1 Introducción

El coloreo de grafos consiste en colorear los vértices de un grafo con el menor número de colores posibles, de forma que dos vértices adyacentes no pueden tener un mismo color. Si existe tal coloreo con k colores, se dice que el grafo es k -coloreable. El número cromático de un grafo G , denotado como $\chi(G)$, representa el número mínimo de colores para un coloreo válido de G . El problema del k -coloreo consiste en determinar si un grafo de entrada es k -coloreable.

Hay varios trabajos para determinar cuándo un grafo es 3-coloreable, dado que el problema de los 3 colores es un problema difícil (un problema NP-completo). Los resultados más recientes para determinar la 3-colorabilidad de un grafo se basan en reconocer si el grafo de entrada es planar y sin triángulos (Grötzsch, 1959), o descubriendo alguna relación entre la topología del grafo y sus partes estructurales (Stacho, 2010; Lozin y Malyshev, 2017; Dvořák et al, 2009; Mertziyos y Spirakis, 2013).

El coloreo de vértices de grafos es un campo activo de investigación con muchos subproblemas interesantes. El problema de coloreo de grafos tiene muchas aplicaciones en áreas como problemas de programación, asignación de frecuencia, planificación, etc.

(Byskov, 2005; Dvořák et al, 2009; Mertziou y Spirakis, 2013). Los grafos planares juegan un papel importante tanto en la teoría de grafos como en el área de dibujo de grafos. De hecho, los grafos planares tienen varias propiedades interesantes: son dispersos, cuatro coloreables, y su estructura interna se describe de manera sucinta y elegante (Cortese y Patrignani, 2004).

La complejidad computacional inherente asociada con la resolución de problemas NP-difíciles ha motivado a la búsqueda de métodos alternativos, que permitan que algunas instancias de problemas NP difíciles se resuelvan en tiempo polinomial. Por ejemplo, en el caso del problema de coloreo de vértices, el 2-coloreo se puede resolver en tiempo polinomial. Además, para determinar la 3-colorabilidad para grafos AT-free y para grafos perfectos, así como para determinar $\chi(G)$ para algunas clases de grafos, tales como: grafos intervalo, grafos chordal y grafos de comparabilidad (Stacho, 2010). En todos esos casos, se han encontrado estructuras especiales (patrones) para caracterizar las clases de grafos que son coloreables y cuya complejidad de solución es de tiempo polinomial.

Una clase de grafos que se puede caracterizar, se forma por cadenas hexagonales, las cuales son representadas en grafos de moléculas de bencenoides catacondensados no ramificados. Las moléculas de bencenoide desempeñan un papel importante en la teoría de química de los hidrocarburos bencenoides (Shiu, 2008). La propensión de los átomos de carbono forman compuestos hechos de arreglos hexagonales fusionadas a lo largo de los bordes, esto motivó el estudio de las propiedades químicas de los hidrocarburos a través de cadenas hexagonales. Estos grafos han sido investigados ampliamente y representan un área de interés relevante en química matemática, ya que se utilizan para cuantificar detalles relevantes de la estructura molecular de los hidrocarburos bencenoides (Došlić y Måløy, 2010; Shiu, 2008; Wagner y Gutman, 2010).

Si el arreglo de polígonos sigue la estructura de un árbol, donde en lugar de nodos tenemos polígonos, y dos polígonos consecutivos comparten un borde, entonces llamamos a ese grafo un árbol poligonal (De Ita et al, 2018).

Consideramos aquí un procedimiento lógico para colorear un árbol poligonal. Mostramos en este artículo, la relevancia de reconocer patrones estructurales en grafos, lo que permite diseñar algoritmos eficientes para colorear ese tipo de grafos.

Por otro lado, SAT es un problema teórico importante que representa la dificultad computacional de la clase NP-completo. A pesar de la dificultad teórica de SAT, los procedimientos actuales de decisión de última generación, conocidos como solucionadores SAT, se han vuelto sorprendentemente eficientes. Posteriormente, estos solucionadores han sido utilizados en aplicaciones industriales (Alexander et al, 2016). Estas aplicaciones rara vez se limitan a resolver un solo problema de decisión, en cambio, una aplicación generalmente resuelve una secuencia de problemas relacionados. Los solucionadores modernos de SAT manejan tales secuencias de problemas como una instancia del problema de satisfactibilidad incremental (ISAT) (Wieringa, 2014).

El problema ISAT es considerado como un problema de satisfactibilidad incremental dinámico. ISAT inicia considerando una fórmula satisfactible F_0 , y en cada iteración se va adicionando un nuevo conjunto de cláusulas: F_1, \dots, F_n . Cada F_i , resulta de un cambio en la fórmula anterior F_{i-1} .

ISAT es de gran interés a una variedad de aplicaciones que necesitan ser procesadas en un entorno evolutivo. Por ejemplo, en aplicaciones como: planificación y programación reactiva, optimización combinatoria dinámica, revisión de fallas en circuitos combinatorios, diseñar algoritmos para el coloreo de grafos, satisfacción de restricciones dinámicas y aprendizaje de máquinas en un entorno dinámico (Mouhoub y Sadaoui, 2007).

Se ha demostrado que tanto SAT como el 3 coloreo son problemas difíciles (problemas NP-completos). Proponemos un algoritmo novedoso para el 3 coloreo en grafos de árboles poligonales. Nuestra propuesta se basa en la especificación lógica de las restricciones dadas de un 3 coloreo de un arreglo poligonal, formando de esta manera, una instancia en tiempo polinomial del problema de satisfactibilidad incremental.

2 Preliminares

Sea $G = (V, E)$ un grafo simple no dirigido (es decir, finito, sin bucles y sin aristas múltiples) con el conjunto de vértices V (o $V(G)$) y el conjunto de aristas E (o $E(G)$). Dos vértices v y w son llamados adyacentes, si $\{v, w\} \in E$, y la arista $\{v, w\}$ es incidental a u y v . Dos aristas son adyacentes si tienen un vértice en común.

La vecindad de $x \in V$ es $N(x) = \{y \in V: \{x, y\} \in E\}$ y su vecindad cerrada es $N(x) \cup \{x\}$, que se denota por $N[x]$. Denotamos la cardinalidad de un conjunto A por $|A|$. El grado de un vértice $x \in V$, denotado por $\delta(x)$, es $|N(x)|$. El grado máximo de G , o simplemente el grado de G , es $\Delta(G) = \max\{\delta(x): x \in V\}$.

Un camino desde un vértice v a w es una secuencia de aristas: $v_0v_1, v_1v_2, \dots, v_{n-1}v_n$ tal que $v = v_0, v_n = w, v_k$ es adyacente a v_{k+1} y la longitud del camino es n . Un camino simple es un camino tal que $v_0, v_1, \dots, v_{n-1}, v_n$ son todos diferentes. Un ciclo es simplemente un camino no vacío en el que el primero y el último de los vértices son idénticos; y un ciclo simple es un ciclo en el que no se repite vértice, excepto el primer y último vértice. Un k -ciclo es un ciclo de longitud k (tiene k aristas). Un ciclo de longitud impar se denomina ciclo impar, mientras que un ciclo de longitud par se denomina ciclo par. Un grafo sin ciclos se llama acíclico.

Dado un subconjunto de vértices $S \subseteq V$, el subgrafo de G donde S es el conjunto de vértices y el conjunto de aristas es $\{\{u, v\} \in E: u, v \in S\}$, se llama subgrafo de G inducido por S y se denota por $G|S$. $G - S$ denota el grafo $G|(V - S)$. El subgrafo inducido por $N(v)$ se denota como $H(v) = G|N(v)$, el cual contiene todos los nodos de $N(v)$ y todas las aristas que los conectan.

Un conjunto independiente o estable es un conjunto de vértices en un grafo donde ninguno de sus vértices es adyacente a otro. Es decir, es un conjunto $S \subseteq V(G)$ de vértices tal que para cualquier par de ellos no hay una arista que los conecte. El tamaño de un conjunto independiente es el número de vértices que contiene. Un conjunto independiente S es maximal si no es un subconjunto propio de otro conjunto independiente, y es máximo si no hay otro conjunto independiente en G con una cardinalidad mayor que $|S|$.

El coloreo de un grafo $G = (V, E)$ es una asignación de colores a sus vértices. El coloreo es válido si los vértices adyacentes siempre tienen colores diferentes. Un k -coloreo de G es

un mapeo de V al conjunto $\{1, 2, \dots, k\}$ de k "colores". El número cromático de G denotado por $\chi(G)$ es el valor mínimo k tal que G tiene un k -coloreo propio. Si $\chi(G) = k$, entonces se dice que G es k -cromático o k -coloreable.

Sea $G = (V, E)$ un grafo. G es un grafo bipartito si V se puede dividir en dos subconjuntos U_1 y U_2 , llamados conjuntos parciales, de modo que cada arista de G se une a un vértice de U_1 a un vértice de U_2 . Si $G = (V, E)$ es un grafo k -cromático, entonces es posible dividir V en k conjuntos independientes V_1, V_2, \dots, V_k , llamadas clases de color, pero no es posible dividir V en $k - 1$ conjuntos independientes.

3 Grafos de árboles poligonales

Dado un grafo conectado no dirigido $G = (V, E)$, al aplicar la búsqueda primero en profundidad sobre G se produce un grafo de árbol T_G , donde $V(T_G) = V(G)$. Las aristas en T_G se llaman aristas de árbol, mientras que a las aristas en $E(G) \setminus E(T_G)$ se les llama aristas frond (aristas de retroceso).

Sea $e \in E(G) \setminus E(T_G)$ una arista frond, la unión del camino en T_G entre los vértices finales de e con la misma arista e conforma un ciclo simple, dicho ciclo se denomina ciclo básico (o fundamental) de G con respecto a T_G . Cada arista frond indica el camino máximo contenido en el ciclo básico del que forma parte. Los nodos finales de un ciclo son los nodos que forman parte de la arista frond del ciclo.

Sea $C = \{C_1, C_2, \dots, C_k\}$ el conjunto de ciclos fundamentales formados durante la búsqueda primero en profundidad sobre G . Dado un par de ciclos básicos C_i y C_j de C , si C_i y C_j comparten alguna arista, a los ciclos se les llama intersectados; de lo contrario, se llaman ciclos independientes. Un conjunto de ciclos es independiente si no se intersectan cualesquiera dos ciclos del conjunto. En particular, si dos ciclos comparten solo una arista, se les llama ciclos adyacentes.

Para dos ciclos intersectados C_i y C_j , el nuevo ciclo formado por la diferencia simétrica entre su conjunto de aristas (denotado como $C_{ij} = C_i \Delta C_j$) es un ciclo compuesto de G y C_i y C_j son llamados ciclos compuestos de C_{ij} . Una forma de calcular el ciclo compuesto C_{ij} es $C_{ij} = (E(C_i) \cup E(C_j)) - (E(C_i) \cap E(C_j))$. Notar que el conjunto $E(C_{ij})$ es independiente del orden de las aristas consideradas en sus ciclos componentes, ya que las aristas comunes entre C_i y C_j son siempre las mismas.

Sea C_k un grafo de ciclo simple de tamaño k . C_k también se le denomina polígono de tamaño k . Una cadena poligonal $P_{k,t}$ es un grafo que se obtiene de identificar un número finito de t polígonos de al menos tamaño k , de manera que cada polígono, excepto el primero y el último, es adyacente exactamente a dos polígonos. Cuando cada polígono en $P_{k,t}$ tiene el mismo número de k vértices, entonces $P_{k,t}$ es un arreglo lineal de t k -gons, denotado P_t .

La forma en que se unen dos polígonos adyacentes, a través de un vértice común o una arista común, define diferentes clases de grafos moleculares (Shiu, 2008). Sea $G = (V, E)$ un grafo molecular, el cual es una representación de la fórmula estructural de un compuesto químico en términos de teoría de grafos.

Sea $P_i = h_1, h_2, \dots, h_t$ una cadena poligonal con t polígonos, donde cada h_i y h_{i+1} comparten exactamente una arista en común e_i , $i = 1, 2, \dots, t-1$. Una cadena poligonal con al menos dos polígonos tienen dos polígonos-finales: h_1 y h_t . Mientras tanto h_2, \dots, h_{t-1} son los polígonos internos de la cadena. En una cadena poligonal, cada vértice tiene grado 2 o 3. Los vértices de grado 3 son exactamente los puntos finales de las aristas comunes entre dos polígonos consecutivos.

El reconocimiento de patrones de estructuras repetitivas en grafos es esencial para el diseño de algoritmos combinatorios eficientes. Por ejemplo, los patrones básicos de los grafos que se estudiarán aquí son polígonos que siguen una estructura de árbol.

Muchos problemas difíciles se pueden resolver de manera eficiente en grafos que pueden no ser árboles, pero en cierto sentido todavía son lo suficientemente parecidos a los árboles. Dado que los grafos a considerar tienen una topología de árbol, permite el diseño de un algoritmo eficiente para el 3-coloreo de sus vértices. Extendemos la definición de *árboles poligonales* introducido en (De Ita et al, 2018) con la siguiente caracterización de un árbol poligonal $G_T = (V, E)$ como se ve en la Figura 1.

Caracterización de árboles poligonales:

1. Toda componente acíclica de G_T se deja fuera de las caras internas de los polígonos.
2. Dos polígonos son adyacentes si comparten una arista común, un vértice común, o están unidos por una sola arista que une un vértice de cada polígono.
3. Cualquier cadena de polígonos adyacentes no debe formar un ciclo de polígonos. Esto significa que el camino de los polígonos adyacentes sigue la estructura de un árbol donde cada polígono podría ser sustituido por un solo vértice de un árbol.

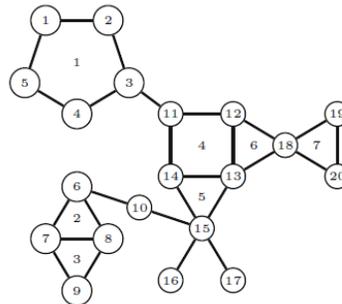


Figura1. Un árbol poligonal.

El hecho de que un árbol poligonal tenga una caracterización tipo árbol, nos ayuda a proponer un algoritmo eficiente para su 3-coloreo. En el siguiente capítulo se muestra como construir en tiempo polinomial sobre el tamaño de la entrada del grafo, un 3 coloreo válido para estos árboles poligonales.

4 Coloreo de un árbol poligonal

Sea K una forma normal conjuntiva (FNC) y l una literal, la reducción de K por l , denotado por $K[l]$, es la fórmula generada al eliminar las cláusulas que contienen l de K (subsumida), y eliminando \bar{l} de las cláusulas restantes (resolución unitaria).

La reducción $K[s]$, donde s es un conjunto de literales $s = \{l_1, l_2, \dots, l_k\}$ se define aplicando sucesivamente $K[l_i]$, $l_i, i = 1, \dots, k$. La reducción de K por l_1 da la fórmula $K[l_1]$, después de una reducción de $K[l_1]$ por l_2 , da como resultado $K[l_1, l_2]$ y así sucesivamente. El proceso continúa hasta que $K[s] = K[l_1, \dots, l_k]$ es alcanzado. En caso de que $s = \emptyset$ entonces $K[s] = K$.

Sea K una FNC y s una asignación parcial de K . Si se obtiene un par de cláusulas unitarias contradictorias mientras $K[s]$ es calculada, entonces K es falsificada por la asignación s . Además, durante el cálculo de $K[s]$, se pueden generar nuevas cláusulas unitarias. Por lo tanto, la asignación parcial s se extiende agregando las cláusulas unitarias que se van formando, es decir, $s = s \cup \{u\}$ donde $\{u\}$ es una cláusula unitaria. $K[s]$ puede reducirse nuevamente usando las nuevas cláusulas unitarias que se van formando. A este proceso iterativo se le denomina Propagación Unitaria (K, s).

Por simplicidad, se abrevia Propagación Unitaria (K, s) como UP (K, s).

La aplicación de UP (K, s) genera una nueva asignación s' que extiende a s , y una nueva subfórmula K' formada por las cláusulas de K que no son satisfactibles por s' . Denotamos a la fórmula K' como $K' = \text{UP}(K, s)$. Notar que si s falsifica a K entonces s' puede tener literales complementarias y K' contiene la cláusula nula. Y cuando s satisfice a K , entonces K' es el conjunto vacío. Denotamos con $v(K)$ al conjunto de variables que aparecen en la fórmula K . La aplicación de UP (K, s) se puede hacer en un tiempo lineal sobre las longitudes $|K| + |s|$.

Presentamos la codificación del 3 coloreo de un grafo como un problema de Satisfactibilidad al considerar la reducción polinomial clásica del 3 coloreo a una instancia 3-SAT.

Lema1. El 3-Coloreo es polinomialmente reducible al problema SAT ($K \wedge \phi$) con K una 2-FNC y ϕ una 3-FNC.

Prueba. Sea $G = (V, E)$ un grafo donde $n = |V|$, $m = |E|$. Definimos las variables lógicas $x_{v,c}$ para indicar que al vértice $v \in V$ se le ha asignado el color $c \in \{1, 2, 3\}$. Para cada vértice $v \in V$, tres variables lógicas $x_{v,1}, x_{v,2}, x_{v,3}$ son creadas. Por tanto habrá $3 * n$ variables booleanas en $v(K) \cup v(\phi)$. Definimos primero las restricciones que formarán a la fórmula K . Para cada arista $e = \{u, v\} \in E$, u y v deben tener un color diferente. Esta restricción se modela mediante las siguientes tres cláusulas binarias: $(\bar{x}_{u,1} \vee \bar{x}_{v,1}) \wedge (\bar{x}_{u,2} \vee \bar{x}_{v,2}) \wedge (\bar{x}_{u,3} \vee \bar{x}_{v,3})$. Existen $3 * |E|$ cláusulas binarias de esta clase.

Otra clase de restricciones binarias permitirá definir el hecho de que cada vértice no debe tener más de un color. Esta restricción se modela mediante las siguientes tres cláusulas binarias: $(\bar{x}_{v,1} \vee \bar{x}_{v,2}) \wedge (\bar{x}_{v,2} \vee \bar{x}_{v,3}) \wedge (\bar{x}_{v,3} \vee \bar{x}_{v,1})$, para cada vértice $v \in V$. Existen $3 * |V|$ cláusulas binarias de esta clase. Ambos conjuntos de $3 * (|V| + |E|)$ cláusulas binarias conforman la fórmula K que estará en 2-FNC.

En este caso, SAT (K) no es suficiente para determinar un 3-coloreo de G , ya que aunque K fuera satisfactible, no hay un 3-coloreo de G deducible de SAT (K).

Para construir soluciones del 3-coloreo de G , una 3-FNC ϕ debe estar formada por las cláusulas que modelan la restricción de que a cada vértice se le debe asignar al menos un color. Entonces, para cada vértice $v \in V$ la siguiente cláusula es generada: $(x_{v,1} \vee x_{v,2} \vee x_{v,3})$. ϕ tiene $|V|$ 3-cláusulas. Además, cada uno de los $3 * n$ variables de v (K) tiene solo una ocurrencia en ϕ .

Esta reducción se realiza en tiempo polinomial sobre el tamaño n y m , ya que consiste en crear $3 * (n + m)$ cláusulas binarias para K y (n) cláusulas ternarias para ϕ . Además, se cumple que G tiene un 3-coloreo si y solo si $(K \wedge \phi)$ es satisfactible.

Dado el conjunto de restricciones $(K \wedge \phi)$, determinar SAT $(K \wedge \phi)$ requiere de un tiempo exponencial en el tamaño $|K \cup \phi|$, lo que a su vez, nos permite determinar un 3-coloreo de G . El carácter no determinista en el proceso de asignar un color a cada vértice de G se refleja al requerir un tiempo exponencial para resolver su respectiva instancia 3-SAT. Sin embargo, para topologías de grafos especiales, como es el caso de los grafos outerplanar y árboles poligonales, nuevas restricciones agregadas a $(K \wedge \phi)$ son útiles para resolver la nueva instancia satisfactible incremental en un tiempo polinomial, ya que el proceso UP (K, s) trabaja en un tiempo lineal.

Supongamos como entrada un grafo conectado $G = (V, E)$, con $n = |V|$ y $m = |E|$, fijamos una búsqueda primero en profundidad sobre G , comenzando la búsqueda con un vértice $v \in V$ de menor grado y visitando los vértices de grado más bajo cada vez que hay múltiples posibles vértices para visitar. Denotamos esta búsqueda primero en profundidad como $dfs(G)$. Sea $G' = dfs(G)$ el nuevo grafo generado por la búsqueda primero en profundidad. dfs nos permite detectar si G tiene ciclos o no, y si estos ciclos son pares e impares en tiempo $O(m*n)$.

Dado $G' = dfs(G)$, sea T_G el árbol de expansión de G' . Note que $V(T_G) = V(G)$. Sea $C = \{C_1, C_2, \dots, C_k\}$ el conjunto de ciclos fundamentales encontrados por $dfs(G)$. Si G es un grafo acíclico, entonces $T_G = G'$ y $C = \emptyset$.

Lema2. Si $G' = dfs(G)$ es acíclico o G' es bipartito, entonces G' es 2-coloreable.

Prueba. Si G' es acíclico o contiene solo ciclos fundamentales de longitud par, entonces G es bipartito y por tanto es 2-coloreable, ya que los vértices en G pueden ser coloreados por niveles, es decir, todos los vértices en el mismo nivel tienen el mismo color y los nodos de dos niveles consecutivos estarán coloreados con los dos colores de forma alternada.

Consideremos ahora un árbol poligonal G_p cuya especificación para cualquier 3 coloreo ha sido codificado por $\phi_0 = (K \wedge \phi)$, una 3-FNC. Presentamos un orden para realizar el 3-coloreo de los vértices de G_p . Esto es estratégico para encontrar un 3 coloreo válido. Sea F_r el conjunto de vértices estratégicos para ser coloreados primero. F_r se forma de la siguiente manera:

1. Todos los puntos finales de las aristas comunes entre polígonos adyacentes deben estar en F_r

2. Además, los vértices comunes entre polígonos adyacentes unidos por un vértice se agregan a F_r .
3. Los puntos finales que son los puentes entre dos polígonos adyacentes se agregan a F_r .
4. Finalmente, todos los elementos en F_r se ordenan en caminos que formen trayectorias maximales entre vértices en F_r .

Sea $x_{v,c}$ la variable lógica que denota que al vértice v se le asigna el color $c \in \{1, 2, 3\}$. Sea $Tres = \{1, 2, 3\}$ el conjunto que contiene tres colores posibles. Para cada vértice $v \in V(G_p)$ un conjunto $Tabu(v)$ es asociado. $Tabu(v)$ indica los colores prohibidos para el vértice v . De hecho, $Tabu(v)$ contiene los colores asociados a los vértices vecinos de v . Notar que $|Tabu(v)| < 2, \forall v \in V$, porque cuando $|Tabu(v)| = 2$, la cláusula $(x_{v,1} \vee x_{v,2} \vee x_{v,3})$ asigna un color de forma inmediata al vértice v .

A partir de F_r se adicionan nuevas cláusulas unitarias a $(K \wedge \phi)$. Estas cláusulas unitarias fijan un color para los vértices de G_p . Los elementos en F_r están ordenados para formar caminos maximales. Mostramos ahora el proceso para asignar un color al elemento actual de F_r de la Figura 1 (ver Figuras 2, 3 y 4).

$$F_r = \{\{14, 13\}, \{12, 13\}, \{7, 8\}, \{18\}, \{3, 11\}\}$$

Ordenando los vértices en F_r para formar caminos maximales:

$$13 - 18 - 12 - 13 - 14 - 11 - 3, 7 - 8$$

$$\{13, 18\} \rightarrow (x_{13,1}), (x_{18,2}) \rightarrow (\bar{x}_{12,1}), (\bar{x}_{12,2}), (\bar{x}_{14,1}), (\bar{x}_{15,1}) \rightarrow (x_{12,3}) \rightarrow (\bar{x}_{11,3})$$

$$\{18, 12\} \rightarrow \emptyset$$

$$\{12, 13\} \rightarrow \emptyset$$

$$\{13, 14\} \rightarrow (x_{14,2}) \rightarrow (\bar{x}_{11,2}), (\bar{x}_{15,2}) \rightarrow (x_{11,1}), (x_{15,3}) \rightarrow (\bar{x}_{3,1})$$

$$\{14, 11\} \rightarrow \emptyset$$

$$\{11, 3\} \rightarrow (x_{3,2})$$

$$\{7, 8\} \rightarrow (x_{7,1}), (x_{8,2}) \rightarrow (\bar{x}_{6,1}), (\bar{x}_{6,2}) \rightarrow (x_{6,3}), (x_{9,3})$$

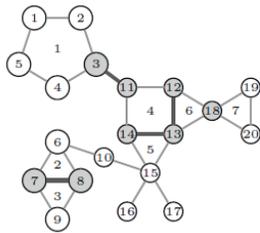


Figura2. Los elementos de F_r

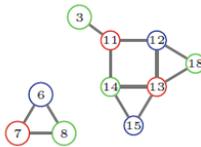


Figura3. Primera parte del colorado

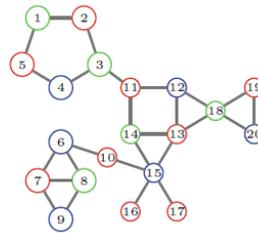


Figura4. 3-colorado final

Algoritmo 3-coloreo (T_G)

Eliminar cualquier vértice de grado como máximo dos, así como cualquier subgrafo acíclico de G {Estos subgrafos pueden ser coloreados al final del proceso}

While $((F_r \neq \emptyset)$ and $(\phi_0 \neq \emptyset))$ **do**

Sea $e = \text{push}(F_r)$;

$e = u$ OR $e = \{u, v\}$

If u (and v when $e == \{u, v\}$) ya ha sido coloreado **then**

Continuar {considerar el siguiente elemento de F_r }

Else

If $e = \{u\}$ **then**

for $v \in N(x)$ and $a \in \text{Tabu}(v) - \text{Tabu}(u)$ **do**

$s = \{X_{u,a}\}$

end for

else

if u ya ha sido coloreado **then**

$b = \min \{Tres - \text{Tabu}(v)\}$

end if

else

if v ya ha sido coloreado **then**

$a = \min \{Tres - \text{Tabu}(u)\}$

end if

else

if $(\text{Tabu}(v) == \text{Tabu}(u) = \emptyset)$ **then**

$a = 1; b = 2;$

end if

else

if $\text{Tabu}(u) \neq \emptyset$ **then**

$a = \min \{Tres - \text{Tabu}(u)\}; \text{Tabu}(v) = \text{Tabu}(v) \cup \{a\};$

end if

else

if $\text{Tabu}(v) \neq \emptyset$ **then**

$b = \min \{Tres - \text{Tabu}(v)\}; \text{Tabu}(u) = \text{Tabu}(u) \cup \{b\};$

$a = \min \{Tres - \text{Tabu}(u)\};$

end if

end if

end if

Sea $s = \{(X_{u,a}), (X_{v,b})\}$ una asignación que determina el color para u, v

Aplicar $\phi_0 = \text{UP}(\phi_0, s)$

end while

Retroceder a los vértices eliminados en el paso 1;

Aplicar el 2-coloreo asignándoles un color diferente al de sus vecinos.

end Algoritmo

La correctés de nuestra propuesta se apoya de los siguientes Lemas:

Lema 3. Un arreglo triangular es 3-coloreable.

Prueba. Dados dos triángulos t_i, t_j unidos por una arista en común $e = \{i, j\}$, la Satisfactibilidad de las restricciones $(x_{i,1} \vee x_{i,2} \vee x_{i,3}), (x_{j,1} \vee x_{j,2} \vee x_{j,3}), (\bar{x}_{i,1} \vee \bar{x}_{i,2}), (\bar{x}_{i,3} \vee \bar{x}_{i,2}), (\bar{x}_{i,1} \vee \bar{x}_{i,3}), (\bar{x}_{j,1} \vee \bar{x}_{j,2}), (\bar{x}_{j,3} \vee \bar{x}_{j,2}), (\bar{x}_{j,1} \vee \bar{x}_{j,3})$, y $s = \{(x_{i,1}, x_{j,2})\}$ determinan un 3 coloreo.

Lema 4. Cada arreglo poligonal donde los polígonos tienen más de 4 lados, es 3-coloreable.

Prueba. Si eliminamos todos los vértices de grado 2 del arreglo, entonces obtenemos las aristas de F_r . El algoritmo anterior determina un 3 coloreo para todos los vértices involucrados en el conjunto F_r . Después de eso, los vértices de grado 2 regresan, y son 3-coloreables ya que sus vértices vecinos han usado solo dos colores.

Lema 5. Cada arreglo poligonal donde los polígonos tienen un número par de lados es 3-coloreable.

Prueba. El procedimiento se basa en el Lema 2, ya que el subgrafo formado por polígonos pares consecutivos es bipartito.

Por lo tanto, para cualquier patrón utilizado para unir los polígonos en un árbol poligonal, nuestra propuesta nos da un procedimiento de tiempo polinomial para el 3-coloreo de tal árbol poligonal.

5 Conclusiones

Se presentó un método para modelar el problema del 3 coloreo sobre árboles poligonales. La propuesta se basa en la especificación de las restricciones lógicas generadas por un 3 coloreo de patrones básicos: polígonos y subárboles que aparecen en el árbol poligonal. Se presentó un algoritmo de tiempo polinomial que resuelve instancias específicas del problema de satisfactibilidad incremental. Nuestro método puede extenderse a grafos 3 coloreables que contienen otro tipo de grafos, como por ejemplo, polígonos y subgrafos outerplanar.

Referencias

- Alexander, N., Vadim, R., y Ofer, S. (2016). "Ultimately incremental sat", *Lógica/Lenguajes, Algoritmos y Nuevos Métodos de Razonamiento*, vol. 8561, pp. 206–218.
- Byskov, J. (2005). *Exact algorithms for graph colouring and exact satisfiability*. Ph.D. thesis, University of Aarhus, Denmark.
- Cortese, P., y Patrignani, M. (2004), *Planarity Testing and Embedding*, Press LLC, pp.1-39.

- De Ita, G., Marcial, R., Lopez, P., y González, M. (2018). “Linear-time algorithms for computing the Merrifield-Simmons index on polygonal trees”, *MATCH Commun. Math. Comput. Chem*, vol. 79, pp. 55–78.
- Došlić, T., y Måløy, F. (2010). “Chain hexagonal cacti: matchings and independent sets”, *Discret. Math*, vol. 310, pp. 1676–1690.
- Dvořák, Z., Král, D., y Thomas, R. (2009). “Three-coloring triangle-free graphs on surfaces”, en: *Proceedings of 20th ACM-SIAM Symposium on Discrete Algorithms*, pp. 120–129.
- Grötzsch, H. (1959). “Ein Dreifarbensatz für dreikreisfreie Netze auf der Kugel”, *Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg, Math.-Natur. Reihe*, vol. 8, pp. 109-120.
- Lozin, V., y Malyshev, D. (2017). “Vertex coloring of graphs with few obstructions”, *Discrete Applied Mathematics*, vol. 216, pp. 273-280, Elsevier.
- Mertzios, G., y Spirakis, P. (2013). “Algorithms and almost tight results for 3-colorability of small diameter graphs”, *Lecture Notes in Computer Science*, vol. 7741, pp. 332,343.
- Mouhoub, M., y Sadaoui, S. (2007). “Systematic versus non-systematic methods for solving incremental satisfiability”, *Int. J. Artif. Intell. Tools*, vol. 16, pp. 543–551.
- Stacho, J. (2010). “3-colouring AT-free graphs in polynomial time”, en Cheong, O., Chwa, K.-Y., Park, K. (eds.), *ISAAC 2010. LNCS*, vol. 6507, (pp. 144–155), Springer, Heidelberg. https://doi.org/10.1007/978-3-642-17514-5_13.
- Shiu, W. (2008). “Extremal Hosoya index and Merrifield-Simmons index of hexagonal spiders”, *Discret. Appl. Math*, vol. 156, pp. 2978–2985.
- Wagner, S., y Gutman, I. (2010). “Maxima and minima of the Hosoya index and the Merrifield-Simmons index”, *Acta Applicandae Mathematicae*, vol. 112, pp. 323–346.
- Wieringa, S. (2014). *Incremental satisfiability solving and its applications*. Ph.D. thesis, Department of Computer Science and Engineering, Alto University.

Capítulo 8

Influencia de la postura corporal sobre la atención

Alfredo Garcia¹, Juan Manuel Gonzalez¹, and Amparo Palomino²

¹Facultad de Ciencias de la Computación, BUAP, Puebla, México
alfredo_amigo18@hotmail.com, jumagoca78@gmail.com

²Facultad de Ciencias de la Electrónica, BUAP, Puebla, México
ampalomino@gmail.com

Resumen. El nivel de atención de un estudiante en su entorno escolar está relacionado con diferentes variables fisiológicas del cuerpo humano. El estudio de las señales fisiológicas relacionadas con el nivel de atención como ondas cerebrales, frecuencia cardíaca, entre otros, ha revelado grandes avances en los últimos tiempos, obteniendo resultados significativos en aplicaciones de diferentes campos de estudio como medicina, robótica, psicología, entre otros.

El grado de influencia de cada tipo de variable fisiológica sobre el nivel de atención es un caso de estudio independiente, debido a la complejidad en el comportamiento de cada una de ellas. En este trabajo se presenta el desarrollo de un sistema para medir la atención en tiempo real y de forma cuantitativa. Las variables fisiológicas obtenidas para poder determinar el nivel de atención de un usuario son: Las ondas cerebrales tipo beta y dos ángulos que describen la postura corporal del estudiante.

Se relaciona el porcentaje de atención que se obtiene de las ondas cerebrales, con los ángulos obtenidos de la postura corporal a través de un análisis matemático cuyo fin es encontrar un modelo de correlación entre estas dos variables. Como resultado se obtiene un modelo matemático para una población de estudiantes con el Trastorno de Déficit de Atención e Hiperactividad (TDAH) y se obtiene otro modelo matemático para una población de estudiantes sin TDAH. Para ello se emplea el método de extracción de características de ambas variables, para posteriormente aplicar la técnica llamada Regresión lineal múltiple. El coeficiente de correlación máximo obtenido es de 0.9785 para el caso de niños con TDAH, este valor se ubica en un intervalo de correlación considerable a correlación muy fuerte. El coeficiente de correlación resultante denota que la postura corporal puede considerarse como un parámetro que influye en el nivel de atención de los estudiantes.

Palabras clave: Trastorno de déficit de atención e hiperactividad, atención, cerebro, postura corporal, procesos cognitivos, pruebas cognitivas, monitoreo de progreso, habilidad académica.

1 Introducción

La necesidad de actualización permanente de las prácticas educativas, la transferencia de conocimientos y la evolución tecnológica dinámica a las que nos enfrentamos con las tecnologías de la información y comunicación en concordancia con las nuevas metodologías de enseñanza-aprendizaje, son aspectos fundamentales que aseguran el impacto de las acciones formativas futuras (Campazzo et al., 2014).

Recientemente diversas técnicas de análisis de sentimientos y de emociones se han utilizado para determinar el estado emocional y cognitivo de una persona debido a que el estado de ánimo influye directamente en el nivel de atención de las personas (Hernández et al., 2016) y en consecuencia en el desempeño académico de los estudiantes (Aballay et al., 2015; Aymerich-franch, 2012; Hernández, 2015).

Se han identificado distintos trastornos que influyen en el nivel de atención de las personas como se presenta en Campazzo et al. (2013) and Aymerich-franch (2012), tanto en la infancia como en la edad adulta (Asiry et al., 2015). Uno de los trastornos más reconocidos es el Trastorno por Déficit de Atención e Hiperactividad (TDAH) y generalmente se diagnostica por primera vez en la infancia aunque los síntomas persisten en la adolescencia e incluso hasta la edad adulta (Huh & Ackerman, 2010; Centers for Disease Control and Prevention, 2020). Algunas formas utilizadas para saber si una persona presenta TDAH (Pascual et al., 2014; Sonne et al., 2016) son a través de: cuestionarios, pruebas intelectuales, tipos de comportamiento, diagnósticos médicos entre otras en Weisberg et al. (2014) y Anuradha et al. (2010). Estas pruebas requieren un largo período de tiempo donde se realiza una observación y un proceso de análisis para obtener un diagnóstico.

Para conocer el grado de afectación que produce el TDAH es necesario que las personas tengan herramientas que puedan proporcionar una retroalimentación del porcentaje de atención al ejecutar una tarea específica (Ravichandran & Huang, 2009). Actualmente existe una variedad de dispositivos comerciales que proporcionan cuantitativamente el nivel de concentración, meditación, relajación y atención (Desney & Nijholt, 2010), pero en algunos casos se logran de manera invasiva, afectando la respuesta del usuario y en consecuencia el diagnóstico final. Las ondas cerebrales son la variable fisiológica relacionada directamente con el nivel de atención del usuario y la postura corporal es la variable fisiológica que se consigue correlacionar con el nivel de atención (GEO tutoriales, 2015), por medio del análisis matemático presentado en este trabajo.

Este sistema tiene un subsistema de retroalimentación (McLaren & Antle, 2017; Regan et al., 2013) físico (Horii et al., 2017) y virtual (Zuckerman et al., 2016) el cual está representado a través de un semáforo. El cambio de color del semáforo indica un cambio brusco en el porcentaje de atención del usuario. Este cambio puede ser ascendente o descendente, dependiendo de la diferencia entre el valor actual y el valor anterior.

Los sensores utilizados en este sistema son colocados de forma superficial en el usuario, con el propósito de estimular una respuesta natural y generar un manejo no invasivo en el desarrollo de la experimentación.

2 Variables fisiológicas relacionadas con el nivel de atención

Las variables fisiológicas utilizadas en este trabajo para calcular el nivel de atención son: las ondas cerebrales y la postura corporal. Las ondas cerebrales están directamente relacionadas con el porcentaje de atención cuando alcanzan un rango de frecuencia de: 14Hz a 26Hz, que pertenece a las ondas tipo beta (Torres et al., 2016). A partir del porcentaje de atención y los ángulos de la postura corporal obtenidos se realiza un análisis matemático para encontrar el grado de correlación entre estas dos variables y determinar si la postura corporal influye de una manera significativa en el nivel de atención del usuario.

En la literatura, la variable fisiológica más utilizada para medir el nivel de atención en las personas son las ondas cerebrales (Desney & Nijholt, 2010), estas se obtienen a través de sensores implementados en forma de diademas, para este trabajo es utilizado el dispositivo MindWave del fabricante Neurosky.

2.1 Ondas Cerebrales

Las señales obtenidas a través de un electroencefalograma (EEG) se pueden clasificar en 5 tipos de ondas con diferentes características (Torres et al., 2016; Guin & Baishya, 2013).

Beta: están en el rango de frecuencia entre 14Hz y 26Hz, pero a menudo se dividen en beta baja y beta alta para un análisis más específico. Las ondas beta son pequeñas y rápidas. Están asociadas con la concentración focalizada. Cuando el movimiento es resistido o suprimido o, al resolver una tarea matemática hay un aumento en la actividad de las ondas beta. Un estado de pánico también puede causar un aumento en el nivel de ondas beta (Torres et al., 2016; Guin & Baishya, 2013).

Gamma: estas ondas están en el rango de frecuencias mayor de 30 Hz. Su amplitud es muy pequeña, y su ocurrencia es rara, por lo que están relacionados con ciertas enfermedades del cerebro. Se cree que refleja el mecanismo de la conciencia. Las ondas beta y gamma juntas se han asociado con atención, percepción y cognición (Torres et al., 2016; Guin & Baishya, 2013).

2.2 Postura Corporal

La postura del cuerpo se define de acuerdo con los tres planos en el espacio, es ideal cuando refleja la simetría de los movimientos. La pérdida de verticalidad pasa por una adaptación de los diferentes segmentos del cuerpo con respecto a los tres planos del espacio. La actitud natural no es consciente ni voluntaria; eso es

una forma de reacción debido a un estímulo constante: la gravedad (Doberssan & Rodríguez, 2004).

La postura no es solo un estado estable o una posición estable, sino es una unidad dinámica de movimiento, en la cual los brazos de palanca y los ángulos de las articulaciones son armónicos, las articulaciones pueden desarrollar un efecto de amortiguación. La postura es una forma fisiológica de usar el cuerpo.

El concepto del centro de gravedad del cuerpo es básico para analizar cualquier posición de descanso o movimiento. Es un punto ubicado en el centro exacto de la masa corporal, su ubicación tiene pequeñas variaciones según la constitución del cuerpo, también en el mismo sujeto se mueve hacia arriba, hacia abajo o hacia los lados, de acuerdo con los cambios en posición de los segmentos orgánicos durante la actividad.

El objeto se comporta como si toda su masa estuviera centrada en ese punto, aunque en realidad, el cuerpo humano está formado por varios segmentos móviles, cada uno de los cuales tiene su propio centro. Sin embargo, cuando se evalúa la alineación de la postura erguida, se considera el cuerpo entero como un todo. El centro de gravedad se encuentra en el 55% de la altura del sujeto, por encima de la segunda vértebra sacra (Doberssan & Rodríguez, 2004).

El mantenimiento o recuperación de la postura normal se produce por medio de reacciones reflejas en las cuales participan receptores, vías nerviosas, centros nerviosos, reflejos miotáticos, actividad muscular tónica y fásica. Estas reacciones son: reacciones de sostén tónico-estáticas; reacciones de adaptación postural; reacciones de enderezamiento y reacciones de equilibrio o tónico- cinéticas.

Las reacciones de sostén tónico-estáticas se producen para mantener la postura normal. Incluyen reacciones reflejas locales (originadas por el mismo miembro) y segmentarias (reacciones cruzadas). Mantienen el cuerpo erguido en contra de la fuerza de gravedad.

3 Pruebas Experimentales

El experimento consistió en aplicar a los estudiantes una prueba para identificar colores, la cual fue obtenida de los demos de la compañía Brain HQ y se explica detalladamente en García et al. (2019a).

Se aplica a una muestra de 22 estudiantes de nivel primaria con TDAH diagnosticado y a una muestra de 11 estudiantes de nivel primaria sin TDAH cuyas edades oscilaron en el rango de 6 a 12 años.

Las muestras fueron capturadas cada 500 milisegundos y los elementos utilizados fueron: 1 tablet (para reproducir la prueba de identificación de colores) utilizada por el usuario, 1 laptop (para monitorear los datos leídos en tiempo real a través de una interfaz gráfica) utilizada por un observador, una banda para medir la postura corporal y una diadema de ondas cerebrales (MindWave). El desarrollo práctico del experimento se muestra en la figura 1.

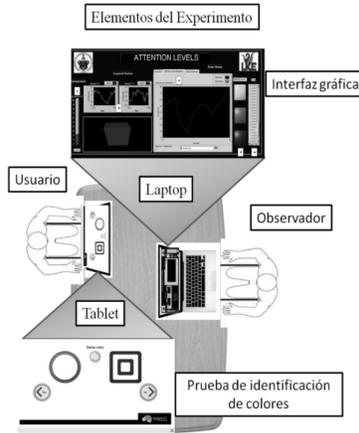


Fig. 1. Diagrama de los elementos del experimento.

La recolección de muestras se llevó a cabo utilizando el dispositivo comercial MindWave de la empresa Neurosky para detectar el nivel de atención sobre las ondas cerebrales tipo beta como se describe en Garcia et al. (2018) y Garcia et al. (2019b). El sensor MPU-6050 implementado en un chaleco se utilizó para medir la postura corporal a través de los ángulos obtenidos en los estudiantes de una escuela primaria en México. El sensor MPU-6050 se colocó inicialmente en un chaleco para poder recolectar las muestras de la postura corporal, el ajuste del chaleco por parte de los estudiantes no era práctico para su talla, esto implicaba un tiempo prolongado en la recolección de muestras, por lo tanto, fue reemplazado por una banda de ajuste, la cual resultó de fácil manejo y mejor acoplamiento a diferentes tallas. La orientación del sensor MPU-6050 con respecto al sistema tridimensional (X, Y, Z) se adaptó de forma que el eje Z fuera perpendicular a la superficie de la tierra. Los ángulos medidos están formados por los movimientos alrededor del eje X y los movimientos alrededor del eje Y; la causa es porque en estos ángulos está presente el soporte de las reacciones tónico-estáticas, estas reacciones ocurren para mantener una postura normal o recta, mantienen el cuerpo erguido en contra de la fuerza de gravedad, una mala postura hace que el usuario requiera un equilibrio neuromotor en relación con su neuro-eje lo cual tiene un impacto directo en su nivel de atención y concentración (Doberssan & Rodríguez, 2004). La postura del usuario es paralela al eje Z y relacionada con el centro de gravedad del cuerpo porque es el punto para analizar cualquier posición ya sea en reposo o en movimiento.

El análisis de datos para encontrar la correlación entre la postura corporal y el nivel de atención, comienza con la extracción de 10 características estadísticas de los ángulos de la postura corporal y 1 característica extraída del porcentaje de atención de las ondas cerebrales tipo beta. Finalmente se compara el desempeño

de 3 modelos matemáticos utilizando la técnica de regresión lineal múltiple, como se aplica en Celaya-Padilla et al. (2018) y se explica en Flores (2019) y Mathworks (2019).

El análisis de datos realizado tiene el siguiente estándar de componentes:

Preprocesamiento. Identificación de valores atípicos y valores faltantes. Suavizado de datos para identificar posibles modelos visuales (Mathworks, 2019).

Resumen o extracción de características. Cálculo de estadísticas básicas para describir la ubicación, escala y forma generales de los datos (Mathworks, 2019).

Visualización: Representación gráfica de datos para identificar patrones y tendencias (Mathworks, 2019).

Modelado: Descripción más detallada de las tendencias presentes en los datos, útil para predecir valores nuevos (Mathworks, 2019).

4 Resultados

En la figura 2 se ilustran los comportamientos de los modelos de regresión lineal múltiple con mayor desempeño comparados con el porcentaje de atención original tanto para el caso de la muestra de niños con TDAH (ecuación 1), como para la muestra de niños sin TDAH (ecuación 2).

$$Y = -610.9 + 328.3x_1 - 108.2x_2 + 233x_3 + 471.1x_4 - 816.7x_5 + 6521.1x_6 - 6476.1x_7 - 5985.9x_8 + 0x_9 - 375.6x_{10} \quad (1)$$

$$Y = -639 + 460x_1 + 109x_2 - 35x_3 + 522x_4 - 1797x_5 - 48559x_6 + 48548x_7 + 49324x_8 + 0x_9 - 447x_{10} \quad (2)$$

Donde:

Y: Es el vector de porcentaje de atención calculado a partir de las características extraídas de la postura corporal.

x_1 : Es el vector de la característica “media” extraída de la postura corporal.

x_2 : Es el vector de la característica “varianza” extraída de la postura corporal.

x_3 : Es el vector de la característica “oblicuidad” extraída de la postura corporal.

x_4 : Es el vector de la característica “curtuosis” extraída de la postura corporal.

x_5 : Es el vector de la característica “desviación estándar” extraída de la postura corporal.

x_6 : Es el vector de la característica “máximo” extraída de la postura corporal.

x_7 : Es el vector de la característica “mínimo” extraída de la postura corporal.

x_8 : Es el vector de la característica “rango” extraída de la postura corporal.

x_9 : Es el vector de la característica “moda” extraída de la postura corporal.

x_{10} : Es el vector de la característica “mediana” extraída de la postura corporal.

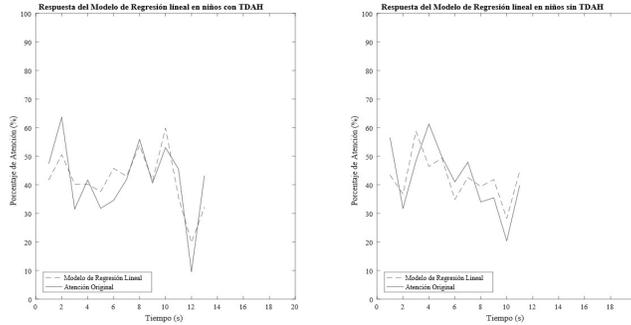


Fig. 2. Respuesta de los modelos de Regresión Lineal.

Conclusiones

El coeficiente de correlación ρ_{xy} puede variar de -1 a 1, donde el signo indica la dirección de la correlación y el valor numérico, la magnitud de la correlación. En este contexto se puede decir que los coeficientes de correlación obtenidos tanto para el caso de los niños con TDAH ($\rho_{xy} = 0.9785$), como para el caso de los niños sin TDAH ($\rho_{xy} = 0.9888$), están en el intervalo de correlación positiva muy fuerte a correlación positiva perfecta, descritos en GEO tutoriales (2015).

La principal desventaja del «Modelo de regresión lineal» obtenido es el uso de 9 de las 10 características extraídas lo cual eleva el coste computacional al realizar la predicción de los datos.

Como trabajo futuro se debe realizar una experimentación con diferentes poblaciones que tengan las mismas características empleadas en este trabajo, con el objetivo obtener el porcentaje de atención a partir de los modelos propuestos y comprobar de forma experimental las variaciones en los coeficientes de correlación obtenidos. Además se debe aplicar la prueba a una mayor población para validar el comportamiento de los modelos descritos en este trabajo, ya que a mayor población menor variación en el error obtenido. Finalmente se propone utilizar otras estrategias de modelado, basadas en el dominio de la frecuencia de las ondas Beta y Gamma.

Referencias

1. Aballay, L., Aciar, S. & Reategui, E. (2015). Propuesta de un método para detección de emociones en E-Learning. 44 JAIIO - ASAI 2015 16^o Simposio Argentino de Inteligencia Artificial, 1(1), 121-128. <https://bit.ly/2ZpOZ14>
2. Anuradha, J., Tisha, B., Ramachandran, V., Arulalan, K.V., & Tripath, B.K. (2010). Diagnosis of ADHD using svm algorithm. In R.K. Shyamasundar, & P. Deshpande, (Eds.), COMPUTE '10: Proceedings of the Third Annual ACM Bangalore Conference

- (pp. 1–4). Association for Computing Machinery. <https://doi.org/10.1145/1754288.1754317>
3. Asiry, O., Shen, H., & Calder, P. (2015). Extending attention span of ADHD children through an eye tracker directed adaptive user interface. In F. Kuo, S. Marshall, H. Shen, M. Stumptner, & M.A. Babar, (Eds.), *ASWEC '15 Vol. II: Proceedings of the ASWEC 2015 24th Australasian Software Engineering Conference* (pp. 149–152). Association for Computing Machinery. <https://doi.org/10.1145/2811681.2824997>
 4. Avila, S.O., Rakona, B., & Mistry, P. (2015). Non-invasive optical detection of hand gestures. In S. Nanayakkara, & E. Yi-Luen, (Eds.), *AH '15: Proceedings of the 6th Augmented Human International Conference* (pp. 179–180). Association for Computing Machinery. <https://doi.org/10.1145/2735711.2735801>
 5. Aymerich-franch, L. (2012). Los juegos en entornos virtuales como herramientas de aprendizaje: Estudio de la respuesta emocional de los participantes. *SPHERA PÚBLICA Revista de Ciencias Sociales y de la Comunicación*, 12(12), 183-197. <https://bit.ly/2Zrve9q>
 6. Bandodkar, A.J., & Wang, J. (2014). Non-invasive wearable electrochemical sensors: A review. *Trends in Biotechnology*, 32(7), 363-371. <https://doi.org/10.1016/j.tibtech.2014.04.005>
 7. Blank, M., & Sinclair, M. (2011). Demo: Non-invasive and long-term core temperature measurement. In R. Ganti, (Ed.), *mHealthSys '11: Proceedings of the First ACM Workshop on Mobile Systems, Applications, and Services for Healthcare* (pp. 1–2). Association for Computing Machinery. <https://doi.org/10.1145/2064942.2064957>
 8. Campazzo, E.N., Martínez, M., Guzmán, A.E., & Agüero, A.L. (2013). Desarrollo de interface de detección de emociones para su utilización en redes sociales y entornos virtuales de aprendizaje. *SEDICI*. <https://bit.ly/2YNg51N>
 9. Campazzo, E.N., Martínez, M., Guzmán, A.E., & Agüero, A.L. (2014). Entornos virtuales de aprendizaje integrado a tecnología móvil y detección de emociones. *SEDICI*. <https://bit.ly/38Px8E>
 10. Celaya-Padilla, J.M., Galván-Tejada, C.E., López-Monteagudo, F.E., Alonso-González, O., Moreno-Báez, A., Martínez-Torteya, A., Galván-Tejada, J.I., Arceo-Olague, J.G., Luna-García, & H., Gamboa-Rosales, (2018). Speed bump detection using accelerometric features: A genetic algorithm approach. *Sensors*, 18(2), 443-448. <https://doi.org/10.3390/s18020443>
 11. Centers for Disease Control and Prevention (Ed.)(2020). *Attention-deficit / hyperactivity disorder (ADHD)*. <https://bit.ly/391E4O5>
 12. Desney, S.T., & Nijholt, A. (2010). *Brain-computer interfaces*. Springer. <https://doi.org/10.1007/978-1-84996-272-8>
 13. Doberssan, L., & Rodríguez, C.A. (2004). *La postura corporal en el deporte simétrico y asimétrico* [Disertación de Maestría, Universidad Abierta Interamericana]. Studylib. <https://bit.ly/38ZGa0W>
 14. Domínguez, C. (2015). *Las ondas binaurales y sus efectos* [Disertación de Maestría, Ciudad Cooperativa Cruz Azul]. Versión online. <https://bit.ly/38htYbB>
 15. Eadie, M., & Steele, R.J. (2017). Non-invasive blood glucose monitoring and data analytics. In S. Zhang, & H. Kettani, (Eds.), *ICDA '17: Proceedings of the International Conference on Compute and Data Analysis* (pp. 138–142). Association for Computing Machinery. <https://doi.org/10.1145/3093241.3093283>
 16. Flores, M.A. (Ed.)(2019). *Implementando una regresión lineal múltiple con MATLAB*. <https://bit.ly/3hbCZWX>
 17. García, A., Calleros, J.M., Guerrero, J., & Palomino, A. (2019). Development of a system for the identification of ADHD in children: attention monitor. In H.M. Fardoun, S. Albiol, A.A.M. Hassan, & E. Guia (Eds.), *REHAB '19: Proceedings of the 5th Work-*

- shop on ICTs for improving Patients Rehabilitation Research Techniques (pp. 87–91). Springer. <https://doi.org/10.1145/3364138.3364159>
18. Garcia, A., Calleros, J.M., & Palomino, A. (2019). Identification of patterns in children with ADHD based on brain waves. In P.H. Ruiz, & V. Agredo-Delgado (Eds.), 5th Iberoamerican Workshop, HCI-Collab 2019 HCI-COLLAB 2019 (pp. 255–268). Springer. <https://doi.org/10.1007/978-3-030-37386-3>
 19. Garcia, A., Calleros, J.M., & Palomino, A. (2018). Data acquisition system for the monitoring of attention in people and development of interfaces for commercial devices. In P.H. Ruiz, & V. Agredo-Delgado (Eds.), 4th Iberoamerican Workshop, HCI-Collab 2018 (pp. 83–97). Springer. https://doi.org/10.1007/978-3-030-05270-6_7
 20. García, A.E. (2015). Análisis de ondas cerebrales para determinar emociones a partir de estímulos visuales [Disertación de Maestría, Universidad Veracruzana]. Repositorio institucional. <https://bit.ly/38enLgB>
 21. Garcia, A., Muñoz, J.M., & Suarez, C. (2008). Gestión de emociones en espacios virtuales de formación. *Investigación Educativa*, 12(21), 45 - 65. <https://bit.ly/3enf1X0>
 22. GEO tutoriales (Ed.)(2015). Cómo se relaciona el Coeficiente de Correlación de Pearson (r) y el Coeficiente de Determinación r Cuadrado (r²). <https://bit.ly/30iuu5l>
 23. Guin, A., & Baishya, B.B. (2013). Brain-controlled wheel chair using LABVIEW [Master dissertation, SRM University]. Researchgate. <https://doi.org/10.13140/RG.2.1.4303.7686>
 24. Hernández, A.M. (2015). Aprendizaje Electrónico Afectivo: Un modelo Innovador para desarrollar una acción tutorial virtual de naturaleza inclusiva. *Formación universitaria*, 8(2), 19-26. <https://bit.ly/2Wjgrvc>
 25. Hernández, A., Vásquez, R., Olivares, B.A., Cortes G., & López I. (2016). Sistema de detección de emociones para la recomendación de recursos educativos. *Revista Ibérica de Sistemas y Tecnologías de Información*, 1(17), 80-95. <http://dx.doi.org/10.17013/risti.17.80-95>
 26. Horii, T., Nagai, Y., & Asada, M. (2017). Active perception based on energy minimization in multimodal human-robot interaction. In B. Wrede, & Y. Nagai, (Eds.), HAI '17: Proceedings of the 5th International Conference on Human Agent Interaction (pp. 103–110). Association for Computing Machinery. <https://doi.org/10.1145/3125739.3125757>
 27. Huh, J., & Ackerman, M.S. (2010). Exploring social dimensions of personal information management with adults with AD/HD. In E. Mynatt, & D. Schoner, (Eds.), CHI EA '10: CHI '10 Extended Abstracts on Human Factors in Computing Systems (pp. 3715–3720). Association for Computing Machinery. <https://doi.org/10.1145/1753846.1754044>
 28. Marín, E.J. (2014). Detección de emociones del usuario [Disertación de Maestría, Pontificia Universidad Católica de Valparaíso]. Versión online. <https://bit.ly/2ZmSvb3>
 29. Mathworks (Ed.)(2019). Análisis de datos. <https://bit.ly/3hhI3c6>
 30. McLaren, E.S., & Antle, A.N. (2017). Exploring and evaluating sound for helping children self-regulate with a brain-computer application. In P. Blikstein, & B. Abrahamson, (Eds.), IDC '17: Proceedings of the 2017 Conference on Interaction Design and Children (pp. 393–398). Association for Computing Machinery. <https://doi.org/10.1145/3078072.3084299>
 31. Naylamp Mechatronics (Ed.)(2016). Tutorial MPU6050, acelerómetro y giroscopio sensor MPU-6050. Attention-deficit / hyperactivity disorder (ADHD). <https://bit.ly/3fuhJLw>
 32. Pascual, M.F., Begoña, Z., & Buldian, K.C. (2014). Adaptive cognitive rehabilitation interventions based on serious games for children with ADHD using biofeedback techniques: Assessment and evaluation. In S. Boll, & F. Köhler, (Eds.), Perva-

- siveHealth '14: Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare (pp. 321–324). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). <https://doi.org/10.4108/icst.pervasivehealth.2014.255249>
33. Perakakis, M., & Potamianos, A. (2013). An affective evaluation tool using brain signals. In J. Kim, (Ed.), IUI '13 Companion: Proceedings of the companion publication of the 2013 international conference on Intelligent user interfaces companion (pp. 105–106). Association for Computing Machinery. <https://doi.org/10.1145/2451176.2451222>
 34. Pinto, R.D., & Ferreira, H.A. (2015). Development of a non-invasive brain computer interface for neurorehabilitation. In H.M. Fardoum, P. Gamito, V.M.R. Penichet, & D.M. Alghazzawi, (Eds.), REHAB '15: Proceedings of the 3rd 2015 Workshop on ICTs for improving Patients Rehabilitation Research Techniques (pp. 126–130). Association for Computing Machinery. <https://doi.org/10.1145/2838944.2838975>
 35. Ravichandran, S., & Huang, J. (2009). Motivating children with attention deficiency disorder using certain behavior modification strategies. In C.T. Lim, & J.C.H. Goh, (Eds.), IFMBE Proceedings (pp. 1057–1068). Springer. <https://doi.org/10.1007/978-3-540-92841-6>
 36. Regan, L.M., Shane, D., Kalyn, R.M., Bertram, P.C., Gaetz, M., Doucette, A., Taylor, B.A., Orr, A.P., & Keiver, K. (2013). Games as neurofeedback training for children with FASD. In N. Sawhney, & E. Reardon, (Eds.), IDC '13: Proceedings of the 12th International Conference on Interaction Design and Children (pp. 165–172). Association for Computing Machinery. <https://doi.org/10.1145/2485760.2485762>
 37. Roig-Maimó, M.F., Mackenzie, I.S., Manresa-Yee, C., & Varona, J. (2017). Evaluating fitts' law performance with a non-ISO task. In J.M. González-Calleros, (Ed.), Interacción '17: Proceedings of the XVIII International Conference on Human Computer Interaction (pp. 1–8). Association for Computing Machinery. <https://doi.org/10.1145/3123818.3123827>
 38. Rojas, S., Garzón, J., Martínez, D., Escobar, M., & Robayo, C. (2012). Lector de ondas cerebrales para implementar un sistema alternativo y aumentativo de comunicación. In M.M. Larrondo, H. Alvarez, I.E. Esparragoza, & C. Rodríguez (Eds.), 10th Latin American and Caribbean Conference for Engineering and Technology. Mega-projects: Building Infrastructure by fostering engineering collaboration, efficient and effective integration and innovative planning (pp. 1–9). LACCEI. <https://bit.ly/3dO3uPS>
 39. Saneiro, M.M. (2015). Apoyo psicoeducativo y afectivo en entornos virtuales de aprendizaje. *International Journal of Developmental and Educational Psychology*, 1(2), 233-241. <https://doi.org/10.17060/ijodaep.2015.n2.v1.338>
 40. Sonne, T., & Jensen, M.M. (2016). ChillFish: A respiration game for children with ADHD. In S. Bakker, C. Hummels, & B. Ullmer, (Eds.), TEI '16: Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction (pp. 271–278). Association for Computing Machinery. <https://doi.org/10.1145/2839462.2839480>
 41. Sonne, T., Marshall, P., Obel, C., Thompsen, P.H., & Grønbaek, K. (2016). An assistive technology design framework for ADHD. In H. Duh, & C. Lueg, (Eds.), OzCHI '16: Proceedings of the 28th Australian Conference on Computer-Human Interaction (pp. 60–70). Association for Computing Machinery. <https://doi.org/10.1145/3010915.3010925>
 42. Torres, F., Sánchez, C., & Palacios, K. (2016). Adquisición y análisis de señales cerebrales utilizando el dispositivo MindWave. *Maskana*, 5(1), 83-93. <https://bit.ly/3ilOLzw>

Capítulo 9

Optimización del Consumo Eléctrico mediante la Heurística Cúmulo de Partículas

Blanca Nydia Perez-Camacho¹, Juan Manuel Gonzalez-Calleros¹, Gustavo Rodriguez-Gomez²

¹ Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación

² Instituto Nacional de Astrofísica, Óptica y Electrónica

blancanydia.perezcgmail.com, jumagoca@gmail.com y
grodrig@inaoep.mx

Resumen. En el presente trabajo se muestra el proceso de optimización del consumo eléctrico de una casa habitación implementando la técnica metaheurística de cúmulo de partículas. Los datos de consumo de los dispositivos domésticos son guardados en vectores de consumo-hora, cada uno de ellos es usado por la heurística de cúmulo de partículas para encontrar en el espacio de búsqueda usando la función objetivo un vector que cumple con el objetivo de optimización.

Palabras Clave: consumo eléctrico, optimización del consumo, cúmulo de partículas.

1 Introducción

En recientes años ha sido objeto de interés el desarrollo de sistemas que ayuden de manera controlada la forma en la cual se lleva a cabo el consumo eléctrico en casas-habitación [(Chen et al., 2017), (Tan, Shan, Hu, & Wu, 2012), (Huang, Tian, & Wang, 2015), (Blecic, Cecchini, & Trunfio, 2007), (Sun, 2017)]. El objetivo primario del control del consumo eléctrico de una casa-habitación está relacionado con la optimización del consumo eléctrico y se enfoca en reducir el consumo de manera que no se vea alterado el perfil del usuario con respecto al uso de los dispositivos [(Zhou et al., 2016), (Mohsin, Javid, Madani, & Abbas, 2018), (Yao, Shen, & Lim, 2017)].

El consumo eléctrico total de una casa es la suma del consumo hora-watts de cada uno de los dispositivos que conforman la infraestructura de dispositivos domésticos. Estos requieren estar conectados a una fuente de alimentación, ya sea porque están conectados todo el tiempo o hasta que tengan la suficiente carga para poder funcionar.

Para el análisis de los datos de consumo hora-watts de los dispositivos se agrupan estos datos en un vector [(Majeed, Bhana, & Parvez, 2018), (Lotfi & Abdi, 2017), (Nadeem, Javaid, Malik, & Iqbal, 2018), (Adika & Wang, 2014)]. La técnica metaheurística de optimización de cúmulo de partículas (PSO) que busca el consumo óptimo de cada uno de los dispositivos de una hora (t), usa el vector de datos de consumo para compararlo con el vector de datos que se ha encontrado en el espacio de búsqueda usando la función objetivo (minimización del consumo).

La organización de este artículo es la siguiente: en la sección 2, se justifica la selección del algoritmo bioinspirado PSO como técnica de búsqueda de optimización a través de la literatura relacionada. La descripción del algoritmo, y de las partes que se compone se dan en la sección 3. En la sección 4, se indica el valor de cada una de las variables usadas en el algoritmo de PSO. Los resultados de la implementación del algoritmo se muestran en la sección 5, y en la sección 6 se enuncian las conclusiones.

2 Preliminares

El problema de la búsqueda del consumo óptimo de un conjunto de dispositivos es del tipo np- completo (Huang et al., 2015), y para darle solución se han implementado técnicas de programación lineal, algoritmos genéticos, teoría de juegos y algoritmos bioinspirados, siendo estos últimos los más usados para estos problemas [(Kakran & Chanana, 2018), (Hao & Wang, 2017), (Yao et al., 2017), (Barbato et al., 2014), (Rasheed et al., 2016), (Nadeem et al., 2018), (Javaid, Naseem, et al., 2017)]

PSO es un algoritmo bioinspirado desarrollado por Kennedy y Eberhart en el año de 1995 (Kennedy & Eberhart, 1995). Éste algoritmo ha sido analizado teóricamente y su convergencia ha sido validada en forma exhaustiva, lo que lo convierte en un algoritmo bioinspirado muy confiable dentro de la gama de este tipo de algoritmos [(Selvarajah, Kadirkamanathan, & Fleming, 2006), (Clerc & Kennedy, 2002), (Trelea, 2003), (Kim & Shin, 2006), (Emara & Fattah, 2004), (Jiang, Luo, & Yang, 2007), (Zeng, Zhang, & Yan, 2010), (Lian & Zhu, 2008), (Van Den Bergh & Engelbrecht, 2006)]. Es por el bajo costo computacional que PSO ha sido usado para reducir los costos en el recibo de la energía eléctrica [(Rahim et al., 2016), (Huang et al., 2015), (Javaid, Hussain, et al., 2017)]. Por los motivos anteriores, en esta investigación el algoritmo PSO se seleccionó para realizar la búsqueda del consumo óptimo de un conjunto de dispositivos eléctricos.

3 Descripción del algoritmo PSO

El algoritmo de optimización por Cúmulo de Partículas (PSO) es una metáfora del comportamiento social de las aves y los peces. PSO es un algoritmo de optimización de búsqueda estocástica conformada por cúmulo de partículas, dónde cada partícula es un candidato de solución, el valor de esta se determina por una función objetivo. Cada una de las partículas se mueve a través del cúmulo ajustando sus trayectorias para encontrar la mejor solución. La posición de cada una de las partículas, x_i , se determina por la velocidad v_i , coeficiente de aceleración, inercia y tamaño del cúmulo de partículas.

Las ecuaciones que definen la velocidad y la posición de las partículas son:

$$v_{ij}(t+1) = wv_{ij}(t) + \varphi_{1j}(t)(y_{ij}(t) - x_{ij}(t)) + \varphi_{2j}(t)(\hat{y}_{ij}(t) - x_{ij}(t)) \quad (1)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{1j}(t+1) \quad (2)$$

En la Tabla 1, se describen los componentes de las ecuaciones 1 y 2.

Tabla 1. Descripción de los componentes de las ecuaciones de velocidad y posición.

Variables	Descripción
i	$1, \dots, s$
j	$1, \dots, n$
w	Factor inercia, este reduce o aumenta la velocidad de la partícula
$\varphi_{1j}(t)$	$c_1 r_{1j}(t)$
$\varphi_{2j}(t)$	$c_2 r_{2j}(t)$
s	Es el número de partículas en el enjambre
n	Número de parámetros de la función que va a ser optimizada
c_1, c_2	Coefficientes de aceleración $0 < c_1, c_2 < 1$, c_1 es el peso cognitivo y c_2 es el peso social
$r_{1j}(t), r_{2j}(t)$	$U(0,1)$, números aleatorios uniformemente distribuidos en el intervalo de (0,1)
$x_i(t)$	Es la posición de la partícula i en el momento t
$v_i(t)$	Es la velocidad de la partícula i en el momento t
$y_i(t)$	Es la mejor solución particular encontrada por la partícula i en el momento t
$\hat{y}_i(t)$	Es la mejor posición global encontrada por la partícula

i en el momento t

Para garantizar la convergencia del algoritmo PSO se debe satisfacer que $0 < c_1, c_2 < 1$, y r_1, r_2 sean número aleatorios uniformemente distribuidos en el intervalo de (0,1) [(Kim & Shin, 2006),(Van Den Bergh & Engelbrecht, 2006),(Jiang et al., 2007),(Trelea, 2003),]

De la ec.(1), la velocidad de la partícula es determinada por los componentes que se mencionan en la Tabla 2.

Tabla 2. Componentes de la ecuación de velocidad.

Componente	Función
$v_i(t)$	Sirve para prevenir la oscilación en la búsqueda de la dirección
$\varphi_1(t)(y_i(t) - x_i(t))$	Componente cognitivo. El cual representa la distancia de la partícula a la mejor solución $y_i(t)$ encontrada por ella. El componente cognitivo le sirve a cada partícula para regresar a donde ha encontrado la mejor solución
$\varphi_2(t)(\hat{y}_i(t) - x_i(t))$	Componente social. Este representa la distancia entre la partícula y la mejor posición global. El componente social representa la tendencia de cada partícula para lograr el éxito de las otras partículas.
$\hat{y}_i(t)$	Representa la mejor solución global encontrada por la partícula i.

El diagrama de flujo PSO se presenta en la Figura 1, y el pseudo código, con función objetivo de minimización, se presenta en Pseudo Código 1.

Pseudo Código 1. Pasos de PSO

Se crea e inicializa el cumulo de partículas de n-dimensiones

```

repeat
  for cada_particula  $i=1, \dots, s$ 
    if  $f(S \cdot x_i) < f(S \cdot y_i)$ 
       $S \cdot y_i < S \cdot x_i$ 
    if  $f(S \cdot y_i) < f(S \cdot \hat{y})$ 
       $S \cdot \hat{y} < S \cdot y_i$ 

```

for *cada_particula* $i=1, \dots, s$
 actualizar la velocidad usando la ec. (1)
 actualizar la posición usando la ec. (2)
until condición_paro es verdadera

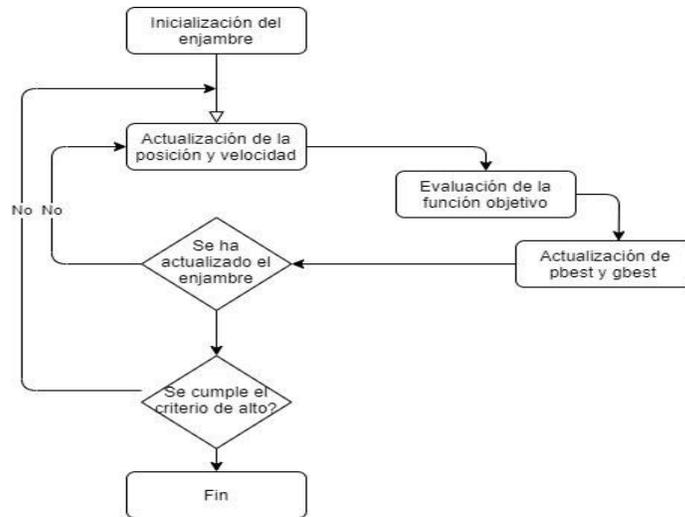


Figura 1. Diagrama de flujo de PSO.

4 PSO aplicado a la búsqueda del consumo óptimo

En nuestro caso particular de aplicación del algoritmo PSO para la búsqueda del estado óptimo de los dispositivos, el espacio de búsqueda se conforma por los vectores de consumo de 10 dispositivos, y un enjambre de 1000 partículas.

El objetivo de la implementación es la búsqueda del consumo mínimo del conjunto de los diez dispositivos. Para llevar a cabo el experimento, la inercia es de 0.8, el peso cognitivo es de 1, el peso social es 1. Los criterios de paro de la ejecución del programa son, ya sea de 50 ciclos o cuando el valor mínimo ya se ha repetido 5 veces. El código del algoritmo se escribió en el lenguaje Python usando el editor Jupyter en una máquina Dell Inspiron 15 5000 series con un procesador Intel Core i7, memoria ram de 8.0 Gb, y sistema operativo Windows 10.

La correspondencia del problema particular a resolver con el algoritmo PSO son:

Cada uno de los elementos del espacio de búsqueda es representado mediante un vector de 10 parámetros $[d_1, \dots, d_{10}]$, donde cada parámetro corresponde al consumo de un dispositivo en el momento m y d_{ij} . Cada vector del espacio de búsqueda se crea colocando en cada parámetro un valor aleatorio obtenido del rango del consumo mínimo ($cMin$) y el consumo máximo ($cMax$) correspondientes al dispositivo. El vector queda de la siguiente manera:

$[ventilador_{consumo}, \text{aire acondicionado}_{consumo}, estufa_{consumo}, foco_{consumo}, calefactor_{consumo}, lavatrastes_{consumo}, microondas_{consumo}, secadora_{consumo}, lavadora_{consumo}, refrigerador_{consumo}]$

Se generan 1000 partículas. El mejor valor encontrado por la partícula que cumple con la función objetivo es almacenado en la variable $bpar$. Cuando la partícula ha terminado de realizar la búsqueda se compara el valor almacenado en su variable $bpart$ con respecto al valor almacenado en la variable $bglobal$. En $bglobal$ se almacena el mejor de los valores encontrados por todas las partículas que cumplen con la función objetivo, esta variable es inicializada con el valor de la $bpart$ de la primera partícula, posteriormente el valor de la $bglobal$ es comparado con el valor $bpart$ de cada una de las partículas y si alguna de ellas es mejor, de acuerdo con la función objetivo, entonces se le es asignado ese valor.

Función objetivo: para este caso de estudio el objetivo es el de minimizar el consumo eléctrico, la función objetivo se define como $\min \sum_{a_i \in A} Y_{a_i}^{\tau}$. Donde A es el conjunto de dispositivos, a_i es un elemento del conjunto A , τ es una hora del día y se encuentra en el rango $[0, 23]$, $Y_{a_i}^{\tau}$ es el consumo por hora del elemento a_i en la hora τ .

En la función objetivo es evaluado cada uno de los elementos del espacio de búsqueda, si alguno de ellos cumple con el criterio de ser menor que el valor actual almacenado en $bglobal$ entonces se actualizan tanto la posición y el valor por los nuevos valores encontrados.

5 Resultados

La base de datos de consumo para la cual se aplica el PSO, está conformada por 24 x 365 registros. Son 24 registros, uno por cada hora del día; para los 365 días del año. Cada uno de estos registros es usado por la técnica PSO como parámetros de estado y consumo de cada uno de los dispositivos y contra el cual se lleva a cabo una comparación con respecto al vector de consumo encontrado por PSO (ver Figura 2).

Se hicieron 100 repeticiones del experimento (100 propuestas de consumo para cada uno de los perfiles de consumo hora-día x 365 días). Se hace un análisis con los datos obtenidos de cada uno de los experimentos. Se obtiene el promedio estadístico de los cien experimentos para cada uno de los diez dispositivos, que refleja el consumo promedio para cada uno de estos. Se calcula la desviación estándar dispositivo-experimento, luego se obtiene el promedio de las desviaciones de los 100 experimentos. Los resultados se muestran en la *Tabla 3*. Para la estufa, foco, lavadora y secadora tienen un mismo consumo sin importar la función que estén realizando, por lo que sus promedios son el consumo normal, y su desviación estándar es de cero ya que no hay una variación.

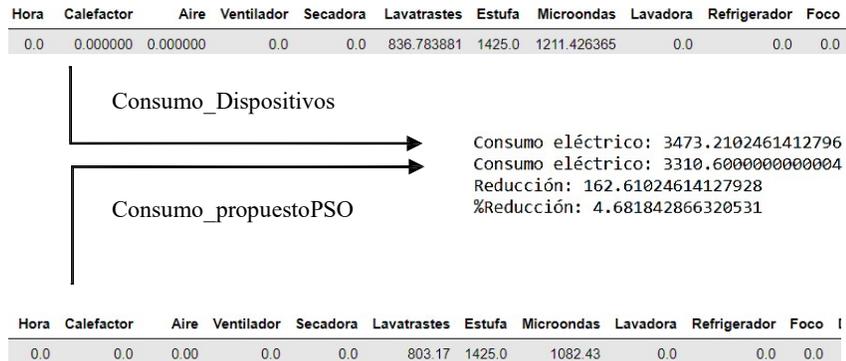


Figura 2. Consumo vs consumo propuesto por PSO

Tabla 3. Análisis de los datos de consumo propuestos por PSO

Dispositivo	Promedio (kW/h)	Desviación Estándar
Aire Acondicionado	1496.341	9.0e-05

Calefactor	8.0800005	5.760072e-02
*Estufa	1425.0	0
*Foco	10.5	0
*Lavadora	895.5	0
Lava trastes	803.1705	2.25e-05
Microondas	1082.43	9e-11
Refrigerador	2861.7000000000003	4.54747350886461e-13
*Secadora	980.625	0
Ventilador	88.81999999999998	1.4210854715202004e-14

6 Conclusiones

En este artículo, se presenta la implementación de la técnica de cúmulo de partículas como método para el procesamiento del consumo eléctrico de 10 dispositivos eléctricos de uso doméstico, con el objetivo de proponer el consumo que cada uno debería de tener para lograr bajar el consumo total sin necesidad de apagar alguno de los dispositivos que se encuentran en estado activo.

Referencias

- Adika, C. O., & Wang, L. (2014). Autonomous Appliance Scheduling for Household Energy Management. *IEEE Transactions on Smart Grid*, 5(9), 673–681.
- Barbato, A., Capone, A., Carello, G., Delfanti, M., Falabretti, D., & Merlo, M. (2014). A framework for home energy management and its experimental validation, 1013–1052. <https://doi.org/10.1007/s12053-014-9269-3>
- Blecic, I., Cecchini, A., & Trunfio, G. A. (2007). A decision support tool coupling a causal model and a multi-objective genetic algorithm, 125–137. <https://doi.org/10.1007/s10489-006-0009-z>
- Chen, S., Liu, T., Gao, F., Ji, J., Xu, Z., Qian, B., ... Guan, X. (2017). Butler, Not Servant: A Human-Centric Smart Home Energy Management System. *IEEE Communications Magazine*, 55(2), 27–33. <https://doi.org/10.1109/MCOM.2017.1600699CM>
- Clerc, M., & Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1), 58–73. <https://doi.org/10.1109/4235.985692>
- Emara, H. M., & Fattah, H. A. A. (2004). Continuous swarm optimization technique with stability analysis. *Proceedings of the American Control Conference*, 3(2), 2811–2817. <https://doi.org/10.1109/ACC.2004.182533>
- Hao, Y., & Wang, W. (2017). Optimal Home Energy Management With PV System in Time of use Tariff Environment. *IEEE Transactions on Consumer Electronics*, (51641702), 2693–2697.
- Huang, Y., Tian, H., & Wang, L. (2015). Demand response for home energy management system. *International Journal of Electrical Power and Energy Systems*, 73, 448–455. <https://doi.org/10.1016/j.ijepes.2015.05.032>

- Javaid, N., Hussain, S., Ullah, I., Noor, M., Abdul, W., Almogren, A., & Alamri, A. (2017). Demand Side Management in Nearly Zero Energy Buildings Using Heuristic Optimizations. *Energies*, *10*(8), 1131. <https://doi.org/10.3390/en10081131>
- Javaid, N., Naseem, M., Rasheed, M. B., Mahmood, D., Khan, S. A., Alrajeh, N., & Iqbal, Z. (2017). A new heuristically optimized Home Energy Management controller for smart grid. *Sustainable Cities and Society*, *34*(July), 211–227. <https://doi.org/10.1016/j.scs.2017.06.009>
- Jiang, M., Luo, Y. P., & Yang, S. Y. (2007). Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. *Information Processing Letters*, *102*(1), 8–16. <https://doi.org/10.1016/j.ipl.2006.10.005>
- Kakran, S., & Chanana, S. (2018). Energy Scheduling of Smart Appliances at Home under the Effect of Dynamic Pricing Schemes and Small Renewable Energy Source. *International Journal of Emerging Electric Power Systems*, 1–12. <https://doi.org/10.1515/ijeeps-2017-0187>
- Kim, D. H., & Shin, S. (2006). Self-organization of decentralized swarm agents based on modified particle swarm algorithm. *Journal of Intelligent and Robotic Systems: Theory and Applications*, *46*(2), 129–149. <https://doi.org/10.1007/s10846-006-9047-3>
- Lian, Z., & Zhu, F. (2008). The Analysis of Particle Swarm Optimization Algorithm. *Journal of Intelligent and Robotic Systems: Theory and Applications*, *46*(2), 129–149. <https://doi.org/10.1007/s10846-006-9047-3>
- Lotfi, J., & Abdi, F. (2017). Smart Home Energy System Modeling and Implementation. *2017 European Conference on Electrical Engineering and Computer Science (EECS)*, 402–404. <https://doi.org/10.1109/EECS.2017.80>
- Majeed, A., Bhana, R., & Parvez, S. (2018). Controlling Energy Consumption by Internet of Things (IoT) Applications. *Journal of Fundamental and Applied Sciences*, *10*(No 4S(2018)), 608–611. <https://doi.org/http://dx.doi.org/10.4314/jfas.v10i4s.13>
- Mohsin, S. M., Javaid, N., Madani, S. A., & Abbas, S. K. (2018). Appliance Scheduling in Smart Homes with Harmony Search Algorithm for different Operation Time Intervals Appliance Scheduling in Smart Homes with Harmony Search Algorithm for different Operation Time Intervals, *1*(February).
- Nadeem, Z., Javaid, N., Malik, A. W., & Iqbal, S. (2018). Scheduling appliances with GA, TLBO, FA, OSR and their hybrids using chance constrained optimization for smart homes. *Energies*, *11*(4), 1–30. <https://doi.org/10.3390/en11040888>
- Rahim, S., Javaid, N., Ahmad, A., Khan, S. A., Khan, Z. A., Alrajeh, N., & Qasim, U. (2016). Exploiting heuristic algorithms to efficiently utilize energy management controllers with renewable energy sources. *Energy and Buildings*, *129*, 452–470. <https://doi.org/10.1016/j.enbuild.2016.08.008>
- Rasheed, M., Javaid, N., Awais, M., Khan, Z., Qasim, U., Alrajeh, N., ... Javaid, Q. (2016). Real Time Information Based Energy Management Using Customer Preferences and Dynamic Pricing in Smart Homes. *Energies*, *9*(7), 542. <https://doi.org/10.3390/en9070542>
- Selvarajah, K., Kadiramanathan, V., & Fleming, P. J. (2006). Stability analysis for the stochastic best particle dynamics of a continuous-time Particle Swarm Optimizer. *Proceedings of AISB'06: Adaptation in Artificial and Biological Systems*, *2*(3), 202–209.
- Sun, X. (2017). An Optimized Scheduling Strategy for Smart Home Users under the Limitation of Daily Electric Charge. *IEEE*.
- Tan, X., Shan, B., Hu, Z., & Wu, S. (2012). Study on demand side management decision supporting system. *2012 IEEE International Conference on Computer Science and Automation Engineering*, 111–114. <https://doi.org/10.1109/ICSESS.2012.6269417>
- Trelea, I. C. (2003). The particle swarm optimization algorithm: Convergence analysis and parameter selection. *Information Processing Letters*, *85*(6), 317–325. [https://doi.org/10.1016/S0020-0190\(02\)00447-7](https://doi.org/10.1016/S0020-0190(02)00447-7)

- Van Den Bergh, F., & Engelbrecht, A. P. (2006). A study of particle swarm optimization particle trajectories. *Information Sciences*, 176(8), 937–971. <https://doi.org/10.1016/j.ins.2005.02.003>
- Yao, L., Shen, J. Y., & Lim, W. H. (2017). Real-Time Energy Management Optimization for Smart Household. *Proceedings - 2016 IEEE International Conference on Internet of Things; IEEE Green Computing and Communications; IEEE Cyber, Physical, and Social Computing; IEEE Smart Data, IThings-GreenCom-CPSCoM-Smart Data 2016*, 20–26. <https://doi.org/10.1109/iThings-GreenCom-CPSCoM-SmartData.2016.31>
- Zeng, W. F., Zhang, Y. J., & Yan, L. (2010). Mechanism of particle swarm optimization and analysis on its convergence. *Proceedings - 3rd International Symposium on Information Processing, ISIP 2010*, 63–66. <https://doi.org/10.1109/ISIP.2010.46>
- Zhou, B., Li, W., Chan, K. W., Cao, Y., Kuang, Y., Liu, X., & Wang, X. (2016). Smart home energy management systems: Concept, configurations, and scheduling strategies. *Renewable and Sustainable Energy Reviews*, 61, 30–40. <https://doi.org/10.1016/j.rser.2016.03.047>

Capítulo 10

Creación automática de redes complejas a partir de secuencias de videovigilancia

Diana Karina Guevara Flores¹, David Eduardo Pinto Avendaño¹, José Antonio Méndez Bermúdez²

¹Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación

²Benemérita Universidad Autónoma de Puebla
Instituto de Física
n.edd.a@hotmail.com

Resumen. En el presente trabajo se lleva a cabo la revisión metodológica del proceso para la creación automatizada de redes complejas a partir de secuencias de videovigilancia, lo que constituye una nueva estrategia para llevar a cabo análisis de alto nivel, como detección de anomalías en video empleando herramientas del procesamiento de texto.

Palabras clave: Redes complejas, Detección de objetos, Sistema automatizado.

1 Introducción

El diseño e implementación de un sistema automatizado que permita crear redes complejas a partir de secuencias de video, es una técnica innovadora que permitirá trasladar el dominio de problemas tradicionalmente ejecutados con herramientas del procesamiento de imágenes o video.

Se ha diseñado una metodología que permite llevar a cabo el diseño automático de redes complejas, que son modelos matemáticos que permiten analizar la topología de sistemas complejos una dinámica emergente distinta de la simple suma de sus partes. La metodología propuesta, parte de la captura de secuencias de video en ambientes reales y la posterior generación de secuencias de texto de magnitud variable, con herramientas de código libre, como descriptores, anotadores automáticos o detectores de imágenes o acciones, entre otros.

Los sistemas de código libre actuales, que permiten generar automáticamente anotaciones descriptivas a partir de frames provenientes de secuencias de video, no poseen la precisión requerida para obtener resultados relevantes a través del análisis de lenguaje natural. Se determinó que los sistemas de detección de objetos de código libre permiten extraer las propiedades emergentes a nivel sistema ocasionadas por las interacciones individuales entre las unidades; en consecuencia, las redes complejas constituyen una metodología apropiada para el modelado y análisis, de manera análoga al caso de estudio del lenguaje natural (Cong et al., 2014).

2 Diseño Metodológico

A continuación, se describe la metodología propuesta para la creación automática de la red compleja a partir de secuencias de videovigilancia reales, que puede implementarse para llevar a cabo análisis en tiempo real.

El sistema propuesto consta de cuatro módulos, el primero de ellos, es el de captura de video, y consta de un sistema de videovigilancia que puede ser móvil o fijo. El siguiente módulo lleva a cabo el preprocesamiento de las secuencias de video, extrae frames con información relevante para el análisis e integra bases de datos. El tercer módulo emplea un sistema de código libre para generar texto a partir de los frames. Finalmente, el cuarto módulo se encarga de crear la red compleja.

Adicionalmente a los módulos constituyentes, se creó un módulo para el proceso de validación, que se compone de una interfaz gráfica de usuario que permite realizar la anotación manual de los frames por un anotador humano. A continuación, se describen detalladamente los módulos que conforman el sistema propuesto y los resultados obtenidos a partir de cada uno de ellos.

Para llevar a cabo la captura de video se empleó la versión fija del sistema de videovigilancia diseñado, a partir del cual, se generó una base de datos de diez horas de video de un escenario urbano entre las 12:00 y las 14:00 horas de mes de febrero de 2020. La información se encuentra dividida en 10 secuencias con una duración aproximada de una hora, cada una.

Como parte del módulo de preprocesamiento, se llevó a cabo la extracción de frames y un módulo para la detección de cambios. Para la generación del texto se empleó un módulo de detección de objetos a través del cual, es posible identificar instancias de la escena urbana como automóviles, camionetas, bicicletas y personas.

La base de datos de frames preprocesados para las diez horas de video, se compone de aproximadamente 20,000 frames que ocupan un espacio en disco de 3.2 GB. y tienen una resolución de 768 x 432 píxeles. Cada uno de estos frames contiene información de un evento relevante y debe ser analizado para detectar cual es la categoría del objeto que ha aparecido en la escena y generado el cambio.

Para la implementación del módulo de generación automática de texto se analizaron los sistemas más relevantes de anotación automática dentro del estado del arte y se implementaron y probaron sistemas de código libre basados en redes neuronales (Bengochea, 2018. Vinyals, 2015. Xu, 2015).

Como resultado de la implementación de los sistemas de anotación automática, se encontró que la precisión de los resultados tenía una muy alta dependencia del dominio o escenario a analizar, por lo que no fue posible su implementación sin pérdida de

generalidad. Como consecuencia, el módulo de generación de texto se implementó por medio de un detector de objetos de código libre.

El sistema de detección de objetos empleado es YOLO (Shafiee, 2017. Huang, 2018), y el entrenamiento se realizó con la base de datos COCO (Lin, 2014), que cuenta con 80 categorías de objetos. Algunas de las categorías no son relevantes para el dominio del escenario urbano de este análisis, por lo que no serán consideradas y no se integrarán en las bases de datos.

Se han obtenido resultados a través de tres tipos de anotación, diferenciados por el grado de automatización del proceso. Para la prueba de concepto, se llevó a cabo la anotación manual, que consistió en el análisis visual, de cada frame sin filtrar aquellos sin cambios, y anotando una nueva línea de un archivo CSV de acuerdo a los objetos observados. Este proceso resultó extremadamente lento, complejo y susceptible de errores.

El segundo método consistió en llevar a cabo la anotación por medio de una interfaz gráfica de usuario y se optimizó el proceso añadiendo un módulo para la detección de cambios, con lo que fue posible simplificar la tarea y reducir drásticamente el tiempo requerido para la anotación. Finalmente, para la etapa de automatización, se empleó un detector de objetos existente y se adaptó para integrarlo con los módulos restantes. Para cada método de anotación empleado, se utilizó un conjunto de clases distinto. Para el método de anotación manual fue posible anotar cada instancia observada, sin importar la peculiaridad de la observación (como el triciclo pepenador o el remolque de bicicleta) o la baja frecuencia de su aparición, por lo que se obtuvieron 33 clases. Adicionalmente, en las anotaciones manuales se observa un alto grado de subjetividad a pesar de haber sido realizadas por una sola persona, lo que elimina el factor de la diferencia de criterio entre anotadores.

Para el método de anotación por medio de interfaz gráfica de usuario, se redujeron las clases a las 16 más frecuentes, debido a la necesidad de mantener el proceso lo más simple y optimizado para que el usuario pueda realizar la tarea en un menor tiempo y con la menor cantidad de errores.

Finalmente, para el método de anotación automática, debido a las características intrínsecas del sistema de detección de objetos empleado, se emplearon 16 clases de las 80 disponibles, algunas de ellas distintas de las utilizadas en la anotación por medio de interfaz gráfica de usuario.

Una característica destacable de este conjunto de clases, radica en que los “hombres”, “mujeres”, “niños” y “bebés” se engloban en una sola clase: la de “persona”, y ya que no existen clases como “policía”, “pepenador”, “vendedor”, “gasero” o “repartidor”, estas instancias también se suman a la clase de “persona”. Así mismo, las clases de “mensajería”, “taxi” e instancias afines, se añaden a las clases de auto o camioneta.

Como resultado del proceso de anotación automática, se obtuvo una matriz de dieci-

séis columnas y 19,531 filas, donde la primera columna es un identificador único que corresponde a fecha y hora de procesamiento de cada frame. Adicionalmente, como resultado de esta etapa del procesamiento, se almacenan las imágenes de las detecciones en un directorio único.

Cada imagen almacenada corresponde a cada una de las filas de la matriz y muestra las detecciones realizadas dentro de un recuadro delimitador. Las imágenes se almacenan en formato JPG, poseen la misma resolución del frame original (768 x 432 píxeles) y ocupan alrededor de 130 KB de espacio en disco cada una.

Para la integración del módulo de creación de la red compleja, se automatizó el acceso a la base de datos en formato CSV generada en el módulo de detección, así como el procesamiento de la matriz resultante. Ya que las clases para este análisis son reducidas, éstas se incluyeron manualmente dentro del código como un arreglo que corresponderá con los nodos de la red. Este ajuste manual de parámetros no afecta la automatización del sistema completo debido a que las clases del análisis son fijas para todo el estudio y solo deberán modificarse en caso de ajustar el sistema para un tipo de escenario distinto.

El resto de parámetros para la creación de la red se automatizó, exceptuando los relacionados con la representación visual, que permanecen fijos para cada caso de análisis. Actualmente es posible implementar la red compleja y crear automáticamente los arreglos necesarios para la creación de nodos y enlaces.

Adicionalmente, se automatizó el proceso de despliegue visual de la red compleja. Ya que la red compleja es un modelo matemático que permite el análisis de sistemas complejos, no se requiere de una visualización gráfica, sin embargo, se ha integrado como funcionalidad adicional del sistema para aumentar la usabilidad y hacerlo amigable con el usuario.

Para un despliegue óptimo, el ajuste del tamaño de los nodos y grosor de los enlaces se ha realizado de forma manual respetando las proporciones reales, pero con base en requerimientos estrictamente visuales. Para redes con menor cantidad de información, como las diseñadas a partir de la anotación manual, se obtuvieron resultados óptimos con una proporción de 1:100 para el grosor de los enlaces, y para la red completa con datos provenientes del anotador automático, se estableció una proporción de 1:400.

Una diferencia significativa de los resultados del etiquetado automático con respecto a los resultados de la anotación manual, radica en que el etiquetador automático lleva a cabo la detección de personas dentro de los vehículos, consideración que no se tomó durante la anotación manual. Debido a esto, los resultados del número de personas detectadas son considerablemente mayores para la anotación automática.

Otro efecto resultante de la anotación de las personas en los vehículos, radica en la relación encontrada entre ambas categorías. En el caso de la anotación manual, el enlace entre personas y vehículos era considerablemente menor debido a que solo

ocurría cuando una persona caminando y un auto se presentaban simultáneamente en el mismo frame. Sin embargo, para el análisis de la anotación automática, el enlace se presenta también en el caso donde el auto circula en dirección suroeste, debido a que el chofer se encuentra dentro del rango visual de la cámara, o bien, si viaja un copiloto o pasajero del lado del vehículo más cercano a la cámara.

Esta condición deberá considerarse para cada cambio en el escenario o en la orientación de la cámara y contrarresta notablemente el efecto de sub-detección de peatones surgido por la poco significativa área visual en dos dimensiones que ocupan con respecto al total de píxeles del frame y el consecuente poco significativo cambio en las métricas para la detección.

Inicialmente se emplearon dieciséis clases para el análisis, sin embargo, se eliminó la de “celular”, debido a que el módulo de detección no fue capaz de identificar instancias de este tipo a causa de la distancia que separa a la cámara de los objetivos, lo que implica un tamaño muy reducido de estos y una probabilidad muy baja de detección.

3 Conclusiones

Se llevó a cabo la implementación de un anotador automático de imágenes de código libre y se realizaron pruebas con frames de un sistema de videovigilancia real. Para validar los resultados del proceso de anotación automático se diseñó una interfaz gráfica de usuario que permite anotar manualmente los frames resultantes del módulo de preprocesamiento.

Se presentan resultados para tres procesos con distinto grado de automatización y se propone una metodología que puede aplicarse a distintos dominios sin pérdida de generalidad. Las pruebas realizadas para esta investigación se llevaron a cabo en un escenario urbano.

Se diseñó una metodología para la creación automática de redes complejas a partir de secuencias de videovigilancia reales. Este proceso permite trasladar el dominio del análisis de video y emplear herramientas del procesamiento del lenguaje natural.

Referencias

- Bengoechea Isasa, J. I. (2018). “Let me see: diseño de un generador automático de descripciones de imágenes”.
- Cong, J., & Liu, H. (2014). “Approaching human language with complex networks”. *Physics of life reviews*, 11(4), 598-618.

Huang, R., Pedoeem, J., & Chen, C. (2018, December). "YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers". In 2018 IEEE International Conference on Big Data (Big Data) (pp. 2503-2510). IEEE.

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.

Shafiee, M. J., Chywl, B., Li, F., & Wong, A. (2017). "Fast YOLO: A fast you only look once system for real-time embedded object detection in video". arXiv preprint arXiv:1709.05943.

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). "Show and tell: A neural image caption generator". In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3156-3164).

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., ... & Bengio, Y. (2015, June). "Show, attend and tell: Neural image caption generation with visual attention". In International conference on machine learning (pp. 2048-2057). PMLR.

Capítulo 11

Análisis de Elevaciones en Superficies Viales mediante Visión Estéreo

Ana L. Ballinas-Hernández, Ivan Olmos-Pineda y J. Arturo Olvera-López

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, 14 Sur y
San Claudio, Ciudad Universitaria, 72570 Puebla, México
analuisa.ballinas@correo.buap.mx, {iolmos, aolvera}@cs.buap.mx

Resumen. Las superficies de calles públicas presentan irregularidades que dificultan la seguridad y el confort en la conducción autónoma. Las principales irregularidades que se presentan son los reductores de velocidad y los baches. El análisis de irregularidades puede ayudar a la identificación de estos obstáculos. En este trabajo se presenta una metodología, basada en visión estéreo, para la obtención de mallas de superficies de calles y el coloreo de curvaturas para visualizar sus irregularidades. Se realiza la reconstrucción 3D de calles donde las nubes de puntos obtenidas son convertidas a mallas triangulares mediante la técnica de triangulación de Delaunay. Los resultados muestran la implementación de un algoritmo para la visualización de irregularidades en las calles mediante el coloreo de curvaturas gaussianas de mallas 3D. Actualmente no se realiza la detección automática de irregularidades, sin embargo, a futuro se aplicarán técnicas de aprendizaje automático para detectar las irregularidades más significativas en las calles.

Palabras Clave: Irregularidades en Calles, Visión Estéreo, Reconstrucción 3D, Mallas de Superficies Triangulares, Curvaturas Gaussianas.

1 Introducción

La navegación de autos autónomos se lleva a cabo de manera aceptable bajo condiciones óptimas de los ambientes de tránsito vehicular. Las superficies de las calles en estos entornos son uniformes y no presentan irregularidades significativas que afecten la conducción. Sin embargo, las vialidades en países en vías de desarrollo presentan comúnmente diversas irregularidades en las superficies, tales como: reductores de velocidad, baches y algunos deterioros. La detección temprana de estos obstáculos ayuda a que la navegación sea segura y confortable.

El objetivo de este trabajo consiste en implementar una metodología para generar mallas de superficies tridimensionales mediante visión estéreo. Además, se aplica el método de coloreo de curvaturas gaussianas para visualizar las irregularidades más significativas en las superficies de calles.

2 Preliminares

Las principales técnicas que se han empleado para analizar irregularidades o para detectar obstáculos en las calles son: sensado de vibraciones mediante sensores de teléfonos inteligentes, visión estéreo, sensores LIDAR (*Laser Imaging Detection and Ranging*) y algunas de estas aplican algoritmos de aprendizaje automático para la detección.

Se han desarrollado aplicaciones móviles que lanzan alertas al detectar irregularidades en las calles que capturan vibraciones mediante el análisis de señales de aceleración en las superficies. Utilizan acelerómetros y sensores ultrasónicos para detectar baches y reductores de velocidad (RV) almacenando sus coordenadas (Salau et al, 2019; Sujitha et al, 2016; Singh et al, 2017; Harikrishnan y Gopi, 2017, 2016; Al-Shargabi et al, 2020; Edwan et al, 2019; Lanjewar et al, 2015). Los sistemas de Posicionamiento Global (GPS) y técnicas de procesamiento de señales se emplean para conocer en tiempo real las ubicaciones de estos obstáculos en las vialidades (Singh et al, 2018; Salau et al, 2019). Además, se han desarrollado algoritmos de extracción de datos obtenidos por estos sensores que aplican algoritmos de minería de datos para encontrar anomalías en las superficies mediante el reconocimiento de patrones (Silva et al, 2017; Souza, 2018). Se ha aplicado la técnica de *Crowdsourcing* para recuperar la ubicación de reductores de velocidad y baches obtenida por sensores y por los usuarios conectados a una red almacenada en la nube (Singh et al, 2017; Chen et al, 2019; Alam et al, 2020). Estos sensores presentan errores de precisión altos, retardos grandes y pueden sobrecargar la red a la que están conectados.

El uso de sensores LIDAR ha sido muy popular en la navegación de autos autónomos. A través de estos sensores es posible obtener datos precisos del contenido de una escena mediante la obtención de un conjunto de datos llamados nubes de puntos (Hernández et al, 2009; Serna y Marcotegui, 2014). Se han analizado superficies para detectar guarniciones de banquetas y la geometría de las carreteras (Vosselman y Zhou, 2009), obteniendo detalles de su morfología y de las irregularidades en las superficies viales (Cavalli et al 2008). Para ello se diseña un algoritmo paralelo que aplica un filtrado de datos de la superficie y de los objetos, así como la segmentación de datos usando operadores morfológicos. A pesar de la alta precisión de estos sensores presentan algunas limitantes: su costo es muy elevado, no siguen protocolos internacionalmente compatibles, requieren un alto poder de cómputo para el procesamiento de grandes cantidades de datos, además, las señales de varios sensores pueden superponerse.

Algunos algoritmos de visión estéreo han sido empleados para analizar superficies de calles y detectar obstáculos sobre el piso (Oniga y Nedevschi, 2010; Broggi et al, 2013; Oniga y Nedevschi, 2011). Se realiza la reconstrucción tridimensional de las escenas a partir de la estimación de imágenes estéreo. También, se han diseñado algoritmos para detectar calles, banquetas y obstáculos identificados por elevaciones en las superficies (Oniga y Nedevschi, 2010). Se ha realizado la estimación de superficies en base a la pendiente de los terrenos y a la presencia de obstáculos mediante visión estéreo y procesamiento de superficies usando el método *B-Splines* (Broggi et al, 2013). De esta forma se generan mapas de elevación de las superficies y se detectan obstáculos como autos o peatones de

acuerdo con la altura en las superficies. Se desarrolló un algoritmo para representar escenas mediante la técnica de *stixel-world* que permite identificar obstáculos por elevación (Badino et al, 2009). La principal desventaja de la visión estéreo es el alto costo computacional al procesar los datos, además de que presentan niveles altos de error en la detección cuando los obstáculos se encuentran a grandes distancias de profundidad.

La Tabla 1 muestra un resumen de algunos trabajos relacionados con el análisis de superficies de calles indicando los métodos empleados y los resultados obtenidos. En estos trabajos se detectan anomalías en las calles pero ninguno ha aplicado la técnica de visión estéreo, abordada en este trabajo, para monitorear irregularidades en las superficies.

Referencia	Método/sensores	Anomalía detectada	Resultados	Limitante
Al-Shargabi et al, 2020	Acelerómetros y GPS	Reductores de velocidad (RV) y baches	Aplicación móvil que envía alertas	Error de detección alto sujeto a los sensores
Alam et al, 2020	<i>Crowdsourcing</i> , clasificación con árboles de decisión	RV, baches y parches	98% de precisión para RV y 92% para baches	Error sujeto al crowdsourcing
Chen et al, 2019	Acelerómetros, análisis de señales, aprendizaje automático	RV y baches	Medida F del 90%	Error sujeto a los acelerómetros
Edwan et al, 2019	Acelerómetros y GPS	RV y baches	Aplicación móvil que envía alertas	Alto error por sensores
Harikrishnan y Gopi, 2017	Acelerómetros y procesamiento de señales	RV y baches	Error de 34.8%	Error alto
Silva et al, 2017	Minería de datos y sensores de <i>smartphone</i>	Varios tipos de RV	Precisión del 87%	La precisión no es muy alta
Singh et al, 2017	Aprendizaje automático y sensores de teléfono	RV y baches	Precisión del 88.66%	La precisión no es muy alta
Souza, 2018	Acelerómetros y aprendizaje supervisado	Calidad del pavimento	Precisión entre 80% y 98%	Error sujeto al acelerómetro
Sujitha et al, 2016	Sensor ultrasónico y GPS	RV y baches	Aplicación móvil que envía alertas	Error alto por los sensores

Tabla 1. Resumen de trabajo relacionado con el análisis de irregularidades en superficies viales.

3 Metodología

La metodología propuesta para la visualización de irregularidades en superficies de calles basada en visión estéreo es mostrada en la Figura 1.

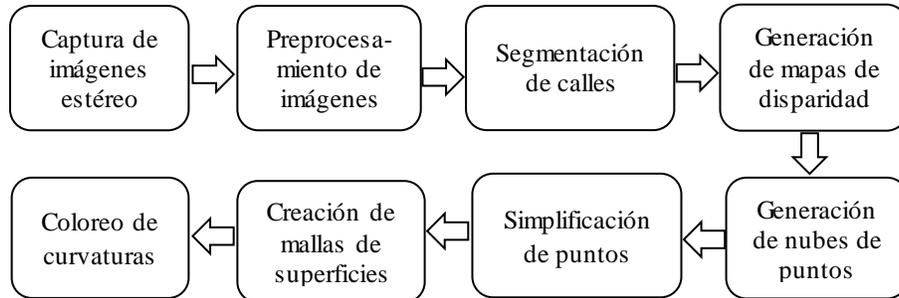


Figura 1. Metodología para la visualización de irregularidades en las calles.

A continuación se describe cada fase de la metodología:

Captura de imágenes estéreo: en esta fase se realiza la recolección de datos de calles reales capturadas con un sistema de visión estéreo que consta de dos cámaras alineadas a la misma altura y a una determinada distancia de separación.

Preprocesamiento de imágenes: se aplica un filtro bilateral para reducir el ruido en las imágenes preservando los bordes mediante el cálculo de la media ponderada de los valores de intensidad de los píxeles cercanos.

Segmentación de calles: se emplea un modelo pre-entrenado del aprendizaje profundo denominado FCN (Fully Convolutional Networks) para la segmentación semántica de calles con el objetivo de encontrar la región de interés. Se aplica una arquitectura de red VGG de 16 capas para la segmentación (Long, et al; 2015). Una FCN es un caso particular de una CNN (*Convolutional Neural Network*) empleada para asociar una categoría con cada pixel de la imagen.

Generación de mapas de disparidad: se aplican las fases de la visión estéreo para reconstruir las escenas tridimensionales: calibración de cámaras, rectificación de imágenes y generación de mapas de disparidad (De la Escalera, et al; 2010). Un mapa de disparidad estima la profundidad de los objetos a partir de un par estéreo.

Generación de nubes de puntos: se realiza la reconstrucción 3D de la calle segmentada usando el mapa de disparidad y aplicando un proceso de triangulación de la visión estéreo para estimar los distintos puntos (x, y, z) de la escena (Cyganek y Siebert, 2011).

Simplificación de puntos: se aplica el algoritmo de muestreo de disco de Poisson para simplificar el total de puntos de la escena y reducir el costo computacional para el procesamiento de los puntos (Corsini et al, 2012). El algoritmo toma muestras de puntos aleatorias y revisa que la distancia entre puntos sea mayor o igual a un determinado radio de separación. De otra forma se van descartando algunos puntos.

Creación de mallas de superficies: las nubes de puntos no estructuradas son utilizadas para generar mallas tridimensionales de las superficies aplicando el método de triangulación de Delaunay (Wöhler, 2013). La condición de Delaunay de un triángulo establece que la circunferencia circunscrita no debe contener ningún otro vértice de la triangulación en su interior.

Coloreo de curvaturas: se aplica el método de coloreo de curvaturas gaussianas para visualizar las elevaciones de las superficies. Las curvaturas principales en un plano p , denotadas como k_1 y k_2 , son los valores máximo y mínimo de esta curvatura. Cuando ambos valores son cero, la curvatura es un plano (Wang et al, 2012).

Para la codificación de la metodología propuesta se aplica el Algoritmo 1 que recibe como entrada un conjunto de imágenes estéreo $E = \{(I_1, D_1), \dots, (I_n, D_n)\}$. Como salida se genera un conjunto de imágenes planas $M = \{M_1, \dots, M_n\}$ correspondientes a las mallas 3D coloreadas mediante el cálculo de curvaturas gaussianas.

Algoritmo 1: Curvaturas de Superficies 3D.

```

Input: conjunto E = {(I1, D1), ..., (In, Dn)}
Output: conjunto M = {M1, ..., Mn}
1 parametrosEstereo ← calibracionEstereo( )
2 for cada par de imágenes Ei = (Ii, Di) en E do
3   (ISE_i, DSE_i) ← segmentacionCalles (Pi)
4   mallaCurvaturas i ← generarMallas3D((ISE_i, DSE_i), parametrosEstereo)
5 end
6
7 Def segmentacionCalles(Ei):
8   Fi ← filtroBilateral(Ei)
9   Si ← probarModeloSegmentacion(Fi)
10  mascara_i ← generarMascaraROI(Si)
11  ROI_i ← recortarRegionInteres(mascara_i)
12 return ROI_i
13
14 Def generarMallas3D((IS_i, DS_i), parametrosEstereo):
15  (IR_i, DR_i) ← rectificacionEstereo((IS_i, DS_i), parametrosEstereo)
16  MD_i ← generarMDSGBM(IR_i, DR_i)
17  MDP_i ← filtroWLS(MD_i)
18  NP_i ← generarNubesPuntos(MDP_i)
19  NPS_i ← simplificarNubesPuntos(NP_i)
20  malla_i ← convertirPuntosAMallas(NPS_i)
21  mallaCurvaturas_i ← colorearCurvaturasGaussianas(malla_i)
22 return mallaCurvaturas_i

```

La primera tarea del algoritmo consiste en obtener los parámetros de calibración de cámaras a partir de la estimación de los parámetros intrínsecos y extrínsecos de las cámaras (De la Escalera, 2010). El algoritmo recorre el conjunto de pares de imágenes estéreo $E_i = (I_i, D_i)$ realizando las siguientes funciones:

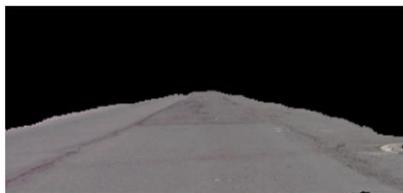
Segmentación de vialidades (*segCalles*(E_i)): esta función recibe como entrada un par de imágenes estéreo E_i y genera como salida un par de imágenes de calles segmentadas con la región de interés ROI_i . Se aplica un filtro bilateral para suavizar las imágenes manteniendo los bordes (*filtroBilateral*(E_i)). El modelo de red neuronal convolucional FCN (*Fully Convolutional Networks*) pre-entrenado es empleado para la segmentación semántica de calles mediante el método *probarModeloFCNPreentrenado*(). Se genera una máscara sobre

cada par de imágenes de acuerdo con la ROI obtenida por la segmentación (*generarMascaraROI(Si)*). Finalmente, se recorta la región de interés correspondiente a la calle (*ROI_i*).

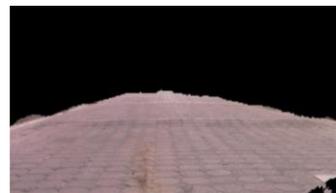
Generar mallas de superficies 3D (*generarMallas3D(Si, parametrosEstereo)*): esta función se encarga de generar mallas 3D a partir de cada par de imágenes estéreo segmentadas con la *ROI_i*. Para ello, se rectifican las imágenes estéreo, izquierda (*IS_i*) y derecha (*DS_i*), a partir de los parámetros de calibración estéreo (*rectificacionEstereo(IS_i,DS_i, parametrosEstereo)*). Se genera un mapa de disparidad aplicando el algoritmo SGBM para encontrar la correspondencia entre píxeles con el enfoque de coincidencia por bloques semi-global (Hirschmuller, 2007). Se aplica un filtro WLS de mínimos cuadrados ponderados para suavizar y hacer más uniforme el mapa de disparidad (*filtroWLS(MD_i)*). Posteriormente, se generan nubes de puntos 3D a partir del mapa de disparidad procesado (*generarNubesPuntos(MDP_i)*). Se simplifica el número total de puntos aplicando el algoritmo de disco de Poisson. Las nubes de puntos se convierten en mallas triangulares aplicando el algoritmo de triangulación de Delaunay. Por último, se genera un mapa de color de la superficie mediante las curvaturas gaussianas para visualizar las elevaciones.

4 Resultados experimentales

En esta sección se presentan resultados considerando un conjunto de imágenes estéreo capturadas bajo condiciones controladas del ambiente. Para la captura se utiliza un sistema de visión estéreo que consta de dos videocámaras Cannon Vixia HR R72, ambas con las mismas características, separadas a una distancia de 20cm. Se consideran imágenes de calles sin oclusiones y de cualquier tipo de material en el piso. El Algoritmo 1 es implementado en el lenguaje Python y es aplicado al conjunto de imágenes estéreo. Al aplicar el modelo FCN VGG de 16 capas pre-entrenado, indicado en el Algoritmo 1, se segmenta la ROI correspondiente a la región de la calle (Long et al, 2015). La Figura 2 muestra dos ejemplos de la ROI en las imágenes de entrada: a) corresponde a una calle sin elevaciones ocasionadas por reductores de velocidad (RV) y b) presenta una calle con elevaciones altas causadas por un reductor de velocidad.



a) Calle sin elevaciones por RV



b) Calle con elevaciones por RV

Figura 2. Segmentación de región de interés en imágenes estéreo.

Al aplicar la calibración de cámaras y la rectificación de la visión estéreo se generan mapas de disparidad donde las intensidades de color indican la profundidad de los objetos en la escena. La Figura 3 a) y b) muestra dos ejemplos de la generación de mapas de disparidad sobre la región de las calles con y sin irregularidad. La Figura 3 c) muestra la paleta de intensidades de color del mapa de disparidad, donde los tonos más oscuros indican que la región está más profunda o alejada y los tonos claros que la región se encuentra más cerca.

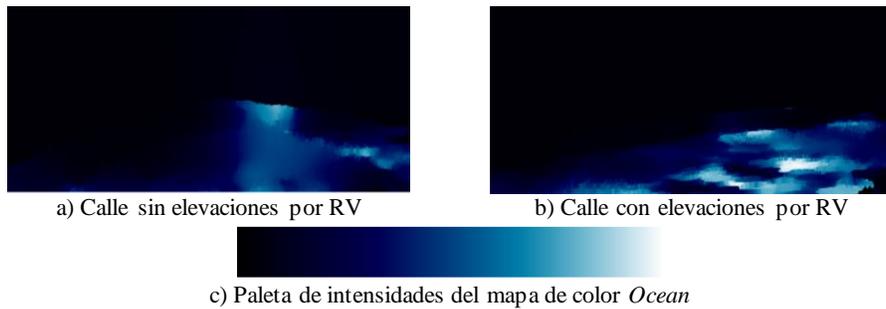


Figura 3. Mapas de disparidad.

La Figura 4 presenta los resultados de la reconstrucción 3D de las calles mediante nubes de puntos. Al aplicar la simplificación de nubes de puntos se obtiene un conjunto compacto de puntos (ver Figura 5).

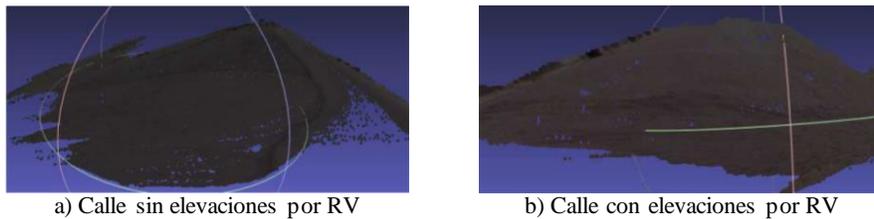


Figura 4. Reconstrucción 3D mediante nubes de puntos.

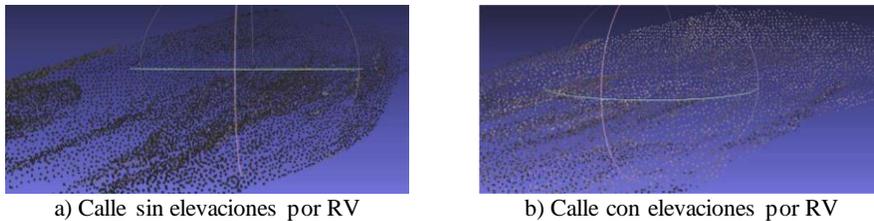


Figura 5. Simplificación de nubes de puntos.

Las nubes de puntos simplificadas se convierten en mallas triangulares que representan las superficies de las calles donde se generan triángulos a partir de tres vértices aleatorios unidos por aristas. Los resultados sobre la región de las calles son mostrados en la Figura 6 donde las mallas pueden ser utilizadas para analizar la forma de las superficies de las calles.

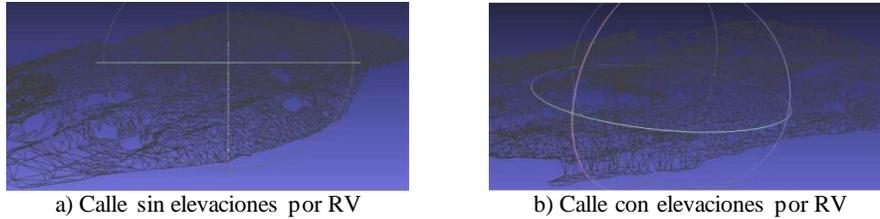


Figura 6. Mallas triangulares de superficies 3D.

Finalmente, la Figura 7 muestra un mapa coloreado de curvaturas gaussianas de las superficies tridimensionales. La curvatura gaussiana de una superficie en un punto es el producto de las curvaturas principales que corresponden a las direcciones en las que la curvatura normal alcanza los valores máximo k_1 y mínimo k_2 y es útil para identificar inflexiones en las superficies. El color rojo en los mapas coloreados indica un valor positivo en las curvaturas gaussianas, el verde se asigna a un valor cero y el azul denota un valor negativo. Esta visualización puede ayudar al análisis de elevaciones en las superficies de calles para la localización de irregularidades. Como se puede observar en ambas imágenes de la Figura 7 se presentan irregularidades sobre las superficies ocasionada por los defectos en las calles. No obstante, la Figura 7 b) muestra alturas más grandes en las irregularidades que la Figura 7 a) debido a que la superficie presenta un reductor de velocidad.

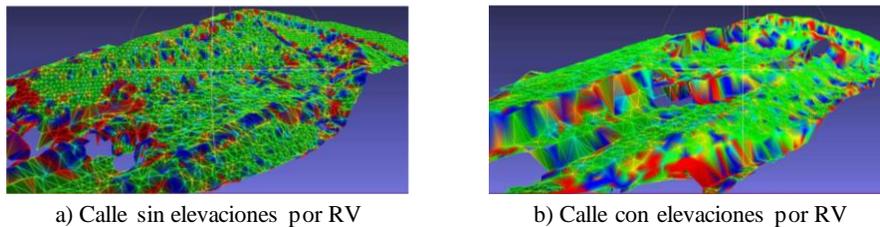


Figura 7. Curvaturas gaussianas de mallas tridimensionales.

5 Conclusiones

En este trabajo se presenta una metodología para generar mallas 3D mediante visión estéreo. Los resultados presentados muestran la correcta aplicación de la visión estéreo para reconstruir escenas 3D y para generar mallas triangulares cuyos vértices están simplificados

para trabajar con una muestra representativa de los datos. Se muestra el coloreo de curvaturas de las mallas tridimensionales donde los colores indican el área plana de la superficie de las calles y sus irregularidades. El coloreo de curvaturas puede ser empleado para visualizar las irregularidades en las superficies de calles. Como trabajo a futuro, se aplicarán técnicas de procesamiento de mallas de superficies 3D para mejorar su calidad. Además, se realizará la detección de irregularidades, tales como reductores de velocidad y baches, usando técnicas del aprendizaje automático y de esta forma los resultados serían comparados con el trabajo relacionado.

Referencias

- Al-Shargabi, B., Hassan, M., y Al-Rousan, T. (2020). “A novel approach for the detection of road speed bumps using accelerometer sensor”, *Technology education management informatics*, vol. 9(2), pp. 469–476. (Artículo de revista)
- Alam, M.Y., Nandi, A., Kumar, A., Saha, S., Saha, M., Nandi, S., y Chakraborty, S. (2020). “Crowdsourcing from the true crowd: Device, vehicle, road-surface and driving independent road profiling from smartphone sensors”, *Pervasive and Mobile Computing*, vol. 61, 101103. (Artículo de revista)
- Broggi, A., Cardarelli, E., Cattani, S., y Sabbatelli, M. (2013). “Terrain mapping for off- road autonomous ground vehicles using rational b-spline surfaces and stereo vision”, *Intelligent Vehicles Symposium (IV) 2013 IEEE*, pp. 648–653. (Memoria en extenso)
- Cavalli, M., Tarolli, P., Marchi, L., y Dalla Fontana, G. (2008). “The effectiveness of airborne lidar data in the recognition of channel-bed morphology”, *Catena*, vol. 73(3), 249–260. (Artículo de revista)
- Chen, Y., Zhou, M., Zheng, Z., y Huo, M. (2019). “Toward practical crowdsourcing-based road anomaly detection with scale-invariant feature”, *IEEE Access*, pp. 67666–6767. (Artículo de revista)
- Cyganek, B., y Siebert, J. P. (2011). “An introduction to 3D computer vision techniques and algorithms”. *John Wiley & Sons*. (Libro)
- Corsini, M., Cignoni, P., y Scopigno, R. (2012). “Efficient and flexible sampling with blue noise properties of triangular meshes”, *IEEE transactions on visualization and computer graphics*, vol. 18(6), pp. 914–924. (Artículo de revista)
- Edwan, E., Sarsour, N., y Alatrash, M. (2019). “Mobile application for bumps detection and warning utilizing smartphone sensors”, *2019 International Conference on Promising Electronic Technologies (ICPET)*, IEEE, pp. 50–54. (Memoria en extenso)
- De la Escalera, A., Armingol, J. M., Pech, J. L., y Gómez, J. (2010). “Detección automática de un patrón para la calibración de cámaras”, *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 7(4), pp. 83–94. (Artículo de revista)
- Harikrishnan, P., y Gopi, V.P. (2017). “Vehicle vibration signal processing for road surface monitoring”, *IEEE Sensors Journal*, vol. 17(16), pp. 5192–5197. (Artículo de revista)
- Hernández, J., y Marcotegui, B. (2009). “Filtering of artifacts and pavement segmentation from mobile lidar data”, *ISPRS Workshop Laserscanning 2009*. (Memoria en extenso)

- Hirschmuller, H. (2007). "Stereo processing by semiglobal matching and mutual information", *IEEE Transactions on pattern analysis and machine intelligence*, vol. 30(2), pp. 328–341. (Artículo de revista)
- Lanjewar, B., Khedkar, J., Sagar, R., Pawar, R., y Gosavi, K. (2015). "Survey of road bump and intensity detection algorithms using smartphone sensors", *International Journal of Computer Science and Information Technologies*, vol. 6(6), pp. 5133–5136. (Artículo de revista)
- Long, J., Shelhamer, E., y Darrell, T. (2015). "Fully convolutional networks for semantic segmentation", *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440. (Memoria en extenso)
- Oniga, F., y Nedeveschi, S. (2010). "Processing dense stereo data using elevation maps: Road surface, traffic isle, and obstacle detection", *IEEE Transactions on Vehicular Technology*, vol. 59(3), pp. 1172–1182. (Artículo de revista)
- Oniga, F., y Nedeveschi, S. (2011). "Curb detection for driving assistance systems: A cubic spline-based approach", *Intelligent Vehicles Symposium (IV), 2011 IEEE*. pp. 945–950. (Memoria en extenso)
- Salau, H. B., y Onumany, A. J., AB Nu, A. M., Onwuka, E. N., Duk Ya, J. J., ZE, H. (2019). "A survey of accelerometer-based techniques for road anomalies detection and characterization", *International Journal of Engineering Science and Application*, vol. 3(1), pp. 8–20. (Artículo de revista)
- Serna, A., y Marcotegui, B. (2014). "Detection, segmentation and classification of 3d urban objects using mathematical morphology and supervised learning", *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 93, pp. 243–255. (Artículo de revista)
- Silva, N., y Soares, J., Shah, V., Santos, M. Y., Rodríguez, H. (2017). "Anomaly detection in roads with a data mining approach". *Procedia computer science*, vol. 121, pp. 415–422. (Artículo de revista)
- Singh, G., y Bansal, D., Sofat, S., Aggarwal, N. (2017). "Smart patrolling: An efficient road surface monitoring using smartphone sensors and crowdsourcing", *Pervasive and Mobile Computing*, vol. 40, pp. 71–88. (Artículo de revista)
- Singh, G., Kumar, R., y Kashtriya, P. (2018). "Detection of potholes and speed breaker on road", *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, *IEEE*, pp. 490–495. (Memoria en extenso)
- Souza, V. M. (2018). "Asphalt pavement classification using smartphone accelerometer and complexity invariant distance", *Engineering Applications of Artificial Intelligence*, vol. 74, pp. 198–211. (Artículo de revista)
- Sujitha, M.S., Ramesh, N., y Kotamraju, S. K. (2016). "Automatic speed controlling of vehicle and detection and notification of potholes and humps" *Journal of Engineering and Applied Sciences*, vol. 100(9), pp. 1921–1924. (Artículo de revista)
- Vosselman, G., y Zhou, L. (2009). "Detection of curbstones in airborne laser scanning data", *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38, pp. 111–116. (Artículo de revista)
- Wang, J., Gu, D., Yu, Z., Tan, y C., Zhou, L. (2012). "A framework for 3d model reconstruction in reverse engineering", *Computers & Industrial Engineering*, vol. 63 (4), pp. 1189–1200. (Artículo de revista)
- Wöhler, C. (2012). "3D computer vision: efficient methods and applications", *Springer Science & Business Media*. (Libro)

Índice de autores

Nombre del Autor	Nacionalidad	
Adelina Escobar Acevedo	Mexicana	
Alfredo Garcia	Mexicana	
Amparo Palomino	Mexicana	
Ana L. Ballinas Hernández	Mexicana	
Blanca Nydia Perez Camacho	Mexicana	
Carlos Leopoldo Carreón Díaz de León	Mexicana	
Cristina Lopez	Mexicana	
Daniel Marcelo González Arriaga	Mexicana	
Daniel Sánchez Ruiz	Mexicana	
David Eduardo Pinto Avendaño	Mexicana	
David Ferreiro Piñeiro	Mexicana	
Diana Karina Guevara Flores	Mexicana	
Guillermo De Ita	Mexicana	
Gustavo Rodriguez Gomez	Mexicana	
Iván Olmos Pineda	Mexicana	
J. Arturo Olvera López	Mexicana	
José Antonio Méndez Bermúdez	Mexicana	
Josefina Guerrero García	Mexicana	Editora
Juan Manuel González Calleros	Mexicana	Editor
María Aurora Diozcora Vargas Treviño	Mexicana	
Orlando Ramos Flores	Mexicana	
Sergio Vergara Limon	Mexicana	

Compiladores

Mireya Tovar Vidal
Claudia Zepeda Cortés
Darnes Vilariño Ayala
Juan Manuel González Calleros
Josefina Guerrero García

Revisores

Abraham Sánchez López
Amparo Dora Palomino Merino
Beatriz Beltrán Martínez
Beatriz Bernabe Loranca
Carmen Cerón Garnica
Claudia Zepeda Cortés
Daniel Valdéz Amaro
Darnes Vilariño Ayala
David Eduardo Pinto Avendaño
Etelvina Archundia Sierra
Fermín Martínez Solís
Georgina Flores Becerra
Guillermina Sánchez Román
Guillermo De Ita Luna
Hilda Castillo Zacatelco
Irene Olaya Ayaquica Martínez
Iván Olmos Pineda
Ivo H. Pineda Torres
Jesús López Gómez
José Andrés Vázquez Flores
José Arturo Olvera López

José de Jesús Lavalle Martínez
José Luis Carballido Carranza
José Raymundo Marcial Romero
Josefa Somodevilla García
Josefina Guerrero García
Juan Manuel González Calleros
Luis Enrique Colmenares Guillén
Manuel Martín Ortiz
Marciano Vargas Treviño
María Aurora Diozcora Vargas Treviño
María Auxilio Medina Nieto
Mario Rossainz López
Mireya Tovar Vidal
Omar Flores Sánchez
Pedro García Juárez
Rafael de la Rosa Flores
Rafael Lemuz López
Reyna Carolina Medina Ramírez
Sergio Vergara Limón

Editores

Mireya Tovar Vidal
Claudia Zepeda Cortés
Darnes Vilariño Ayala
Juan Manuel González Calleros
Josefina Guerrero García

Lenguaje, conocimiento y tecnología educativa: avances recientes
Coordinado por
Mireya Tovar Vidal
Claudia Zepeda Cortés
Darnes Vilariño Ayala
Juan Manuel González Calleros
Josefina Guerrero García
está disposición en PDF en la página
de la Facultad de Ciencias de la Computación
de la Benemérita Universidad Autónoma de Puebla (BUAP)
<https://www.cs.buap.mx/mtovar/doc/Libros/LibroCDLKE21.pdf>
a partir de diciembre de 2021
Peso del archivo: 6.9 MB