Vigilada Mineducación

METODOLOGÍA PARA EL ANÁLISIS DE LA SIMILITUD ENTRE MARCAS MEDIANTE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO

Methodology for measuring trademark similarity using Machine Learning techniques

SANTIAGO ECHEVERRI CALDERÓN

Trabajo de grado

Asesor, docente

Edwin Nelson Montoya Múnera

UNIVERSIDAD EAFIT

ESCUELA DE INGENIERÍAS

MAESTRÍA EN CIENCIAS DE LOS DATOS Y LA ANALÍTICA

CONTENIDO

ÍNDICE DE TABLAS Y FIGURAS

RESUMEN

Las marcas son los signos distintivos que usa un empresario para identificar sus productos y servicios. Con frecuencia constituyen uno de los activos más valiosos de las empresas y es por esto por lo que existen normas para su registro y protección. Cuando una marca se registra, le genera a su titular el derecho de impedir que terceros comercialicen productos similares con marcas idénticas o similares.

En los procesos de registro y protección marcaria es necesario establecer la similitud entre 2 marcas y determinar posibles confusiones que se puedan generar en los consumidores. Tradicionalmente esta similitud se ha determinado mediante un diagnóstico cualitativo realizado por un humano, pero ante la creciente cantidad de marcas que buscan ser registradas mes a mes, se configura la necesidad de automatizar esta tarea.

En el presente proyecto se evalúan diferentes técnicas del Procesamiento del Lenguaje Natural (NLP por sus siglas en inglés), la Visión por Computador y la fonología, aplicadas en el contexto del cotejo de marcas, para así obtener un sistema de modelos que permita establecer la similitud entre marcas a nivel visual, ortográfico y fonético.

Los modelos se evalúan sobre un conjunto de datos de oposiciones reales en solicitudes de registro marcario presentadas ante la Superintendencia de Industria y Comercio de Colombia (SIC).

Palabras clave: *marcas; registro marcario; propiedad industrial; similitud de imágenes; similitud fonética; similitud de texto; aprendizaje automático.*

ABSTRACT

Trademarks consist of the symbols and words that businesses use to identify their products and services. They are often one of the most valuable assets of a company and therefore there are regulations for their registration and protection. When a trademark is registered, it gives its holder the right to prevent third parties from marketing similar products with identical or similar symbols.

In trademark registration and protection processes it is necessary to determine the similarity between 2 trademarks to detect potential confusion that may mislead consumers. Traditionally, this similarity has been established through a qualitative human assessment, but given the increasing number of trademarks registration, the need to automate this task is configured.

This research evaluates different techniques of Natural Language Processing (NLP), Computer Vision and phonology, applied in the context of trademark matching, to obtain a system of models that can measure visual, spelling, and phonetic similarity between trademarks.

The proposed method is evaluated on a dataset of trademark registration oppositions in applications filed with the Colombian Trademark Office (Superintendencia de Industria y Comercio).

Keywords: *trademarks; trademark registration; industrial property; image similarity; phonetic similarity; text similarity; machine learning.*

1 INTRODUCCIÓN

1.1 Planteamiento del problema

El registro de una marca es un mecanismo que otorga a su titular el derecho exclusivo de impedir que terceros comercialicen productos similares con una marca similar para evitar confusiones en los consumidores.

En Colombia, el registro marcario es administrado por la Superintendencia de Industria y Comercio (SIC), y se realiza mediante un proceso que consta de las siguientes etapas [1]:

i. **Radicación:** consiste en la presentación de la solicitud de registro ante la SIC.
ii. **Examen de forma:** verificación del cumplimiento de los requisitos previstos en la legislación por parte de la SIC.
iii. **Publicación:** divulgación de las solicitudes de marca que pretenden registrarse, que tiene por objeto permitir a los titulares de las marcas protegidas, oponerse a la solicitud de registro de las marcas que se está publicando. Esta difusión se realiza mediante la Gaceta de Propiedad Industrial.
iv. **Oposiciones de terceros:** trámite que puede iniciar cualquier persona que tenga legítimo interés para oponerse al registro de una marca solicitante.
v. **Examen de fondo:** examen de registrabilidad que hace la SIC del signo solicitado, éste considera los requisitos legales y las oposiciones presentadas por terceros.

Los titulares de marcas registradas deben entonces hacer un monitoreo manual a la Gaceta de Propiedad Industrial, con el fin de detectar cualquier solicitud de marca, en etapa de publicación, que pueda estar en conflicto con su marca registrada y de esta forma poder presentar oportunamente una oposición al registro de la marca solicitante [2].

Según el Régimen Común de Propiedad Industrial de la Comunidad Andina existe un conflicto entre 2 marcas si los productos que identifican tienen alguna conexidad competitiva y si los signos que componen la marca son idénticos o similares. Esta semejanza puede darse por cualquiera de los siguientes criterios [3]:

- Similitud ortográfica: se presenta por la semejanza de las letras entre los signos a compararse.
- Similitud fonética: ocurre entre signos que al ser pronunciados tienen un sonido similar.
- Similitud figurativa: se refiere a la semejanza de elementos gráficos de los signos en conflicto.

Tradicionalmente la similitud se ha determinado mediante un diagnóstico manual y cualitativo, esto implica que la valoración de similitud debe ser realizada por un humano y que su resultado puede ser subjetivo.

1.2 Justificación

En Colombia, durante el año 2021 se presentaron 54,941 solicitudes de registro de marca ante la Superintendencia de Industria y Comercio (SIC). Esto implica que el titular de una marca registrada debería revisar todas estas solicitudes y determinar si alguna constituye una amenaza para su marca, para así poder presentar una oposición oportuna.

Ante el creciente volumen de solicitudes de registro, resulta necesario buscar alternativas que permitan automatizar la tarea rutinaria que implica la comparación y apoyen los procesos de registro y vigilancia marcaria minimizando la carga operativa.

1.3 Objetivos

1.3.1 Objetivo general

Evaluar diferentes técnicas del Aprendizaje Automático para desarrollar y probar una metodología que permita medir la similitud entre marcas y detectar posibles conflictos de propiedad intelectual.

1.3.2 Objetivos específicos

- Medir el desempeño de diferentes técnicas analíticas de similitud ortográfica entre los componentes nominativos (texto) de pares de marcas y seleccionar la técnica que mejor se comporte en el contexto marcario.
- Medir el desempeño de diferentes técnicas analíticas de similitud entre los componentes figurativos (imagen) de pares marcas y seleccionar la técnica que mejor se comporte en el contexto marcario.
- Medir el desempeño de diferentes técnicas analíticas de similitud entre los componentes fonéticos de pares marcas y seleccionar la técnica que mejor se comporte en el contexto marcario.
- Realizar la integración de las 3 técnicas, crear un modelo integrado y realizar una evaluación del desempeño del modelo con una métrica de similitud que incluya los 3 criterios integrados (nominativo, figurativo y fonético).

2 MARCO TEÓRICO Y ESTADO DEL ARTE

A continuación se presenta el marco teórico y estado del arte para cada uno de las dimensiones que requiere el cotejo marcario: similitud ortográfica, la similitud fonética y la similitud figurativa. La búsqueda de literatura se realizó en Google Scholar.

2.1 Similitud ortográfica

La similitud ortográfica o de texto consiste en medir la similitud entre palabras, oraciones, párrafos y documentos es un componente importante en diversas tareas, como la recuperación de información, la clasificación de documentos y el resumen de textos [4]. Los métodos de similitud de texto pueden agruparse en 2 aproximaciones: similitud basada en caracteres y similitud basada en términos.

2.1.1 Similitud de texto basada en caracteres

Entre los métodos más adoptados se encuentran la *distancia de Damerau-Levenstein* [5], que evalúa la similitud como el costo de las operaciones de edición necesarias para transformar una cadena en otra; la *distancia de Jaro-Winkler* [6], que se basa en el número y orden de los caracteres entre dos cadenas; el algoritmo *de N-Gramas*, que se basa en comparar subsecuencias de n elementos de las cadenas de texto [7] y el algoritmo de *Subcadena Común más Larga (Longest Common Substring),* el cual considera que la similitud entre dos cadenas se basa en la longitud de la cadena contigua de caracteres que existen en ambas cadenas [8].

2.1.2 Similitud de texto basada en términos

La comparación de textos basada en términos se aborda como un problema de comparación de conjuntos, en el que las oraciones o párrafos se tratan como conjuntos de términos. Una vez se tienen las frases en esta representación se mide la similitud con métricas tradicionales para comparar conjuntos como el *Índice de Jaccard*, que se calcula como el número de términos compartidos sobre el número de todos los términos únicos en ambas cadenas [9], o el *Coeficiente de Dice*, que se define como el doble del número de términos comunes en las cadenas comparadas dividido por el número total de términos en ambas cadenas [10].

2.2 Similitud fonética

Los algoritmos de comparación fonética consisten en métodos para identificar cadenas de texto que suenan similares, independientemente de su ortografía. La primera solución a este problema fue propuesta por Robert Russell con el algoritmo *SoundEx* [11]. Este algoritmo está enfocado en la pronunciación de nombres y apellidos, y está basado en la codificación de cadenas texto mediante reglas para generar un código de cuatro caracteres, bajo esta codificación palabras que tengan pronunciaciones similares tendrán códigos *SoundEx* similares.

Con el tiempo han surgido múltiples variaciones y mejoras del algoritmo *SoundEx*, como el algoritmo *Metaphone*, el cual cuenta con un conjunto más amplio de reglas de pronunciación y permite diferentes longitudes de codificaciones [12]. Sin embargo, una limitación común es que estos algoritmos son específicos a un idioma, por ejemplo, el algoritmo *SoundEx* original sólo considera la pronunciación de palabras en el idioma inglés.

Como el presente proyecto se evaluará en el contexto colombiano se considerarán algoritmos adecuados para el idioma español como *PhoneticSpanish* [13], *Spanish Metaphone* [14] y *Metasoundex* [15].

PhoneticSpanish es un algoritmo de codificación fonética en español basado en *SoundEx*, en este se adaptan las reglas de codificación para incluir sonidos propios del español y letras como la *ñ*.

Spanish Methapone es una adaptación al español del algoritmo *Double Metaphone*, a diferencia de *PhoneticSpanish*, esta conserva la información relacionada con las vocales y por ende contiene información sobre los diptongos en la palabra.

MetaSoundex es un híbrido de *SoundEx* y *Metaphone*, en éste se hace una primera codificación utilizando las reglas de *Metaphone*, de manera que se conservan las vocales, el código obtenido es posteriormente codificado bajo las reglas de *SoundEx*. Este algoritmo puede usarse tanto en inglés como en español, en el segundo caso se utilizan las adaptaciones al español de *SoundEx* y *Metaphone* mencionadas previamente.

2.3 Similitud figurativa

En la literatura científica la mayoría de las investigaciones relacionadas con logos se han centrado en las áreas de *Búsqueda y Recuperación* y *Detección de logos en fotografías*. Ambos problemas requieren implícitamente un análisis de similitud de imágenes, por lo tanto, constituyen una buena referencia para el presente proyecto.

El primer paso para automatizar la comparación de imágenes es extraer características visuales para obtener representaciones numéricas de la imagen. Los atributos más comunes consisten en representaciones del color, la textura y la forma de la imagen [16]. Este tipo de atributos se conocen como atributos “manuales” o tradicionales.

En los últimos años, las *Redes Neuronales Convolucionales* (*CNNs* por sus siglas en inglés) se han utilizado ampliamente en diversas tareas de reconocimiento visual y han logrado mejores resultados en comparación con los métodos tradicionales [17].

Las *CNNs* son una arquitectura de redes neuronales artificiales enfocada en el campo de reconocimiento de patrones en imágenes [18]. Estas redes están compuestas por 3 tipos de capas:

- Capas convolucionales que se encarga de obtener las características de la imagen aplicando diferentes filtros o *kernels*, como filtros de desenfoque o de detección de bordes.
- Capas de submuestreo o *pooling* que buscan reducir la cantidad de parámetros reteniendo la información relevante.
- Capas completamente conectadas que consiste en un perceptrón de varias capas tradicionales y que utiliza una función de activación *SoftMax* en la capa de salida. El término "totalmente conectada" significa que cada neurona en la capa anterior está conectada a cada neurona en la siguiente capa.

Investigaciones como [19] han demostrado que las redenes neuronales convolucionales tienen una gran capacidad para extraer características complejas de las imágenes y han concluido que en el caso de similitud de logos los atributos profundos (obtenidos mediante *CNNs*) mejoran el desempeño de los modelos de similitud sobre los modelos que usan atributos tradicionales.

En el presente trabajo, para comparar los componentes figurativos de las marcas, se compararán 2 metodologías que utilizan *CNNs* y que demostrado buenos resultados en el contexto de logos en [19 y 20], *Transfer Learning* y *One-shot Learning* respectivamente.

2.3.1 Transfer learning

Transfer Learning o *Conocimiento por Transferencia* es un método en el que un modelo desarrollado para una tarea se reutiliza como punto de partida para resolver otra tarea relacionada [21]. En el caso particular de comparación de logos, el *Conocimiento por Transferencia* se ha aplicado utilizando redes neuronales convolucionales entrenadas para clasificación de imágenes. De éstas se aprovecha su capacidad para extraer atributos de las imágenes y se omite la capa de clasificación. Posteriormente los vectores de atributos de las imágenes pueden compararse con una medida de similitud (como la similitud coseno) para establecer la semejanza entre las imágenes o logos.

En el presente trabajo se utilizaron modelos entrenados para la clasificación de imágenes con el dataset *ImageNet* [22]. El reto *ImageNet - reconocimiento visual a gran escala* (*ILSVRC* por sus siglas en inglés) se ha convertido en un punto de referencia para comparar modelos de clasificación de imágenes, el cual se compone de cientos de categorías de objetos y millones de imágenes. Como resultado, los modelos pre-entrenados han aprendido representaciones ricas en funciones para una amplia gama de imágenes. Los siguientes modelos se utilizaron en el presente trabajo como generadores de atributos para la comparación de imágenes:

2.3.1.1 VGG16

VGG16 es un modelo propuesto por Karen Simonyan y Andrew Zisserman del *Visual Geometry Group Lab* de la Universidad de Oxford en 2014 [24]. Es una red neuronal convolucional que consta de 16 capas de profundidad (13 capas convolucionales y 3 capas completamente conectadas), en la cual se reemplazan los filtros o *kernels* de gran tamaño por varios *kernels* de tamaño 3×3 uno tras otro.

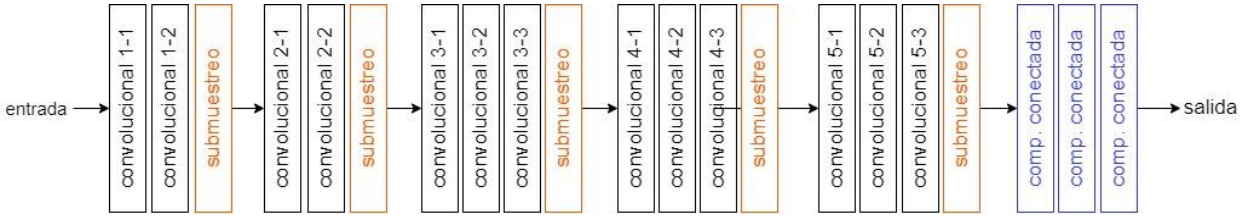


Figura 1 - arquitectura VGG16

2.3.1.2 ResNet50

La arquitectura *ResNet (Residual Networks)* [25] fue propuesta en 2015 por investigadores de Microsoft Research con el objetivo de resolver el problema de la "*desaparición o explosión*" de gradientes en redes neuronales, en el cual el gradiente tiende a 0 o a infinito en la medida en la que se incrementa el número de capas de la red.

Para resolver el problema esta red introduce el concepto de *bloques residuales*, el cual consiste en omitir el entrenamiento de algunos bloques de capas y en su defecto agregar la entrada original x a la salida del bloque convolucional. El objetivo detrás de esto es que, en lugar de que las capas aprendan el mapeo inicial $H(x)$, se hace que la red ajuste la diferencia o *residual* entre el mapeo inicial y su entrada, es de decir:

$$F(x) = H(x) - x, \text{ obteniendo:}$$

$$H(x) = F(x) + x$$

De esta forma el aprendizaje de funciones $H(x)$ que puede resultar complejo para una secuencia de capas se simplifica aprendiendo las funciones residuales correspondientes $H(x) - x$.

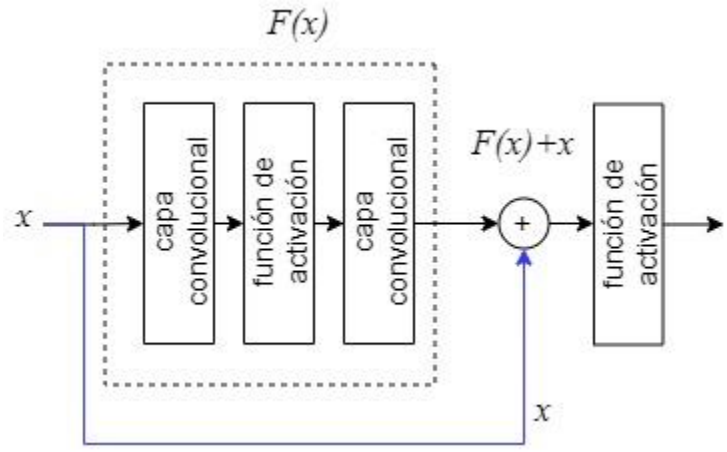


Figura 2 - bloque residual de ResNet

Existen diferentes variantes de la arquitectura *ResNet*, las cuales conservan el concepto de bloques residuales, pero varían en la cantidad de capas. En el presente proyecto se trabajó con *ResNet50V2* [26], una red neuronal convolucional con arquitectura *ResNet* que tiene 50 capas de profundidad:

- 48 capas de convolución.
- 1 capa de submuestreo máximo (*Max Pooling),* que calcula el valor máximo de los parámetros.
- 1 capa de submuestreo máximo (*Average Pooling),* que calcula el valor promedio de los parámetros.

2.3.1.3 Xception

Xception es una arquitectura de redes neuronales convolucionales propuesta por Google en 2017 [27]. Esta arquitectura además de utilizar bloques residuales como *ResNet*, involucra el concepto de *convoluciones separadas por profundidad*.

Típicamente las imágenes digitales están conformadas por matrices bidimensionales (ancho y alto) en 3 canales de color (RGB). La principal diferencia entre las convoluciones convencionales y las *convoluciones separadas por profundidad* es que las convoluciones convencionales se realizan en los 3 canales de entrada simultáneamente, mientras que en las *convoluciones por profundidad*, cada canal se mantiene separado. Esto se traduce en que el número de conexiones sea menor y el modelo sea más ligero.

La arquitectura *Xception* consta de 3 partes:

- Un flujo de entrada que cuenta con un bloque convencional de capas convolucionales tradicionales y 3 bloques residuales de capas *convolucionales separadas por profundidad*.
- Un flujo intermedio con un bloque de capas *convolucionales separadas por profundidad*, el cual se repite 8 veces.
- Finalmente, un flujo de salida que clasifica la imagen con una regresión logística.

2.3.2 One-shot learning

Habitualmente las redes neuronales convolucionales requieren una gran cantidad de datos etiquetados para ser entrenadas. La metodología *One-Shot Learning* tiene como objetivo resolver este problema al requerir una o pocas observaciones de entrenamiento para cada clase. Esto se logra mediante un sistema de *redes neuronales siamesas*, es decir, 2 redes idénticas con los mismos parámetros y pesos, que se utilizan para aprender una función de similitud, la cual toma como entrada 2 imágenes y determina qué tan similares son [23].

La red neuronal convolucional que mapea imágenes de entrada x puede tener internamente cualquier arquitectura y su salida es un *vector de características* o *embbedings* $f(x)$.

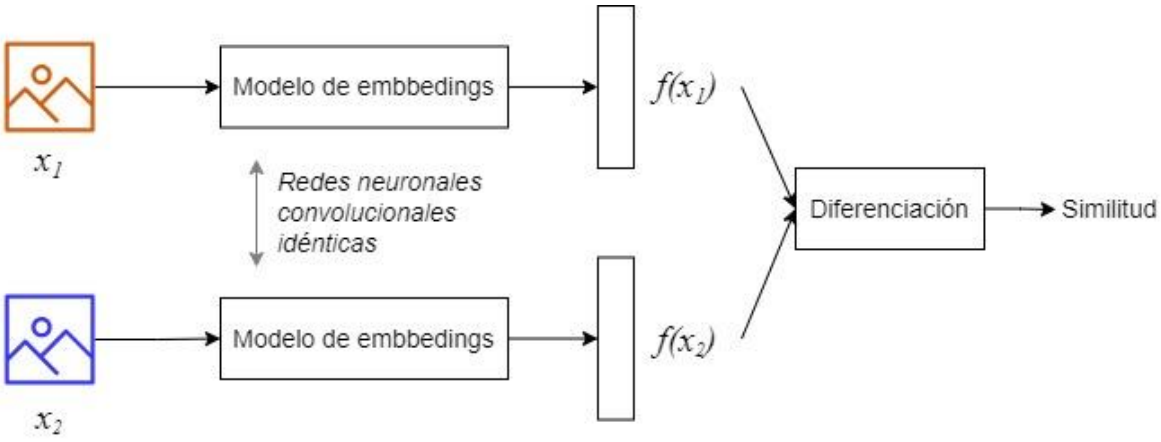


Figura 3 – arquitectura de redes siamesas

El objetivo de la arquitectura siamesa no es clasificar las imágenes de entrada, sino diferenciarlas. Por lo tanto, las funciones de pérdida habitualmente usadas para problemas de clasificación (como la entropía cruzada) no son adecuadas. En redes siamesas principalmente se utilizan la función la función de *pérdida contrastiva* y la función de *pérdida con tripletas*:

2.3.2.1 Redes siamesas con pérdida contrastiva

La *pérdida contrastiva*, introducida en [28], es una pérdida basada en distancia y no en error de predicción. Se utiliza para aprender representaciones en las que dos puntos similares tienen una distancia euclidiana pequeña y dos puntos diferentes tienen una distancia euclidiana grande.

Para entrenar una red siamesa con pérdida contrastiva se requieren pares de imágenes x_1, x_2 con una etiqueta binaria Y que indica si las imágenes son similares o diferentes.

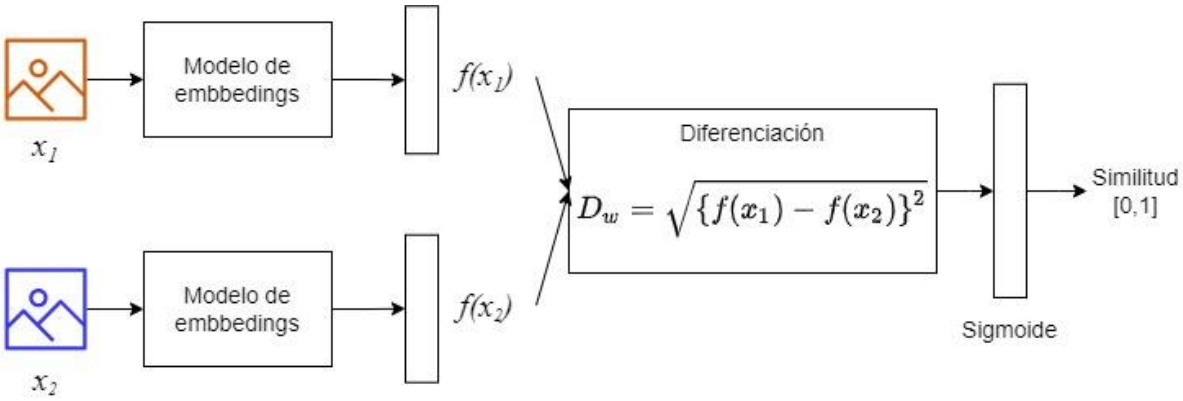


Figura 4 – red siamesa con pérdida contrastiva

La pérdida en el entrenamiento de la red se calcula con la siguiente función:

$$L = (1 - Y)\frac{1}{2}(D_W)^2 \ + \ (Y)\frac{1}{2}\{max\ (0, m \ - \ D_W)\}^2$$

En donde D_W es la distancia euclidiana entre los vectores de características de las 2 imágenes de entrada y m es un margen positivo que define un radio alrededor del vector de características, de modo que sólo los pares de imágenes diferentes contribuyan a la función de pérdida si D_W se encuentra dentro del radio.

2.3.2.2 Redes siamesas con pérdida de tripletas

La función de *Pérdida de Tripletas* fue propuesta en el sistema de reconocimiento facial *FaceNet* [29]. Este concepto consiste en usar tripletas de imágenes en el conjunto de entrenamiento de la siguiente forma:

- Una imagen base, llamada imagen ancla.
- Una imagen diferente, pero de la misma clase que la imagen ancla, llamada imagen positiva.
- Una imagen de una clase diferente a la del ancla, llamada imagen negativa.

Estas imágenes se procesan a través de una red neuronal convolucional con el fin de obtener sus *embeddings* o *vectores de características*, de manera que se pueda comparar la similitud entre estos vectores.

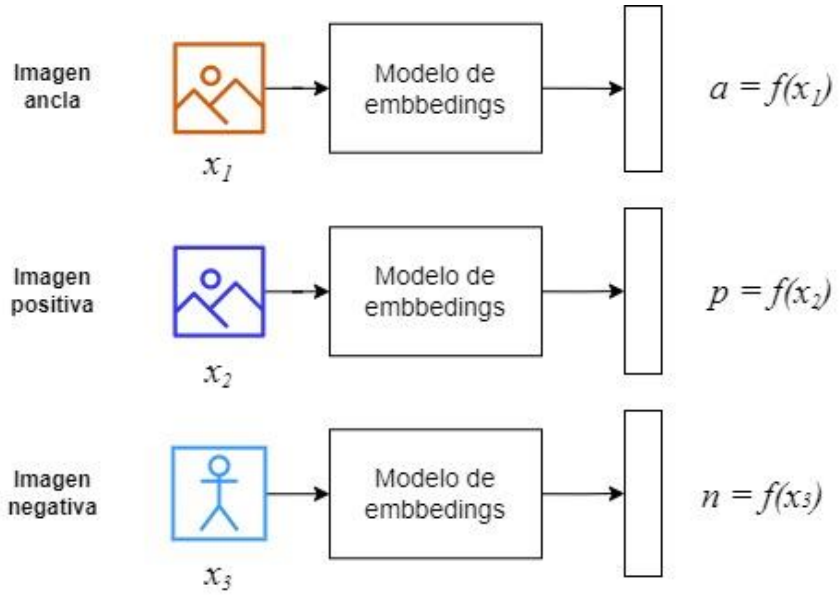


Figura 5 – red siamesa con pérdida de tripletas

La distancia entre un par de las imágenes de entrada se calcula como la norma de la diferencia entre sus vectores de características:

$$d(x_1, x_2) = \|f(x_1) - f(x_2)\|_2^2$$

Y con la función de pérdida se buscará minimizar la distancia entre el vector de características de la imagen ancla y vector de características de la imagen positiva y maximizar la distancia entre el vector de características de la imagen ancla y el vector de características de imagen negativa de modo que:

$$d(a,p) \leq d(a,n)$$

$$d(a,p) - d(a,n) \leq 0$$

La pérdida en el entrenamiento de la red se calcula entonces de la siguiente manera:

$$L = max\,(d(a,p) - d(a,n) + \alpha, 0)$$

En donde α es un margen que tiene el propósito de que la red no satisfaga la ecuación de manera trivial haciendo 0 la distancia entre el ancla y la distancia positiva y negativa.

Una consideración importante en esta arquitectura es que el desempeño del modelo depende en gran medida de la selección de las tripletas de entrenamiento. Si a, p y n se seleccionan de manera aleatoria, la desigualdad $d(a,p) + \alpha \leq d(a,n)$ se puede satisfacer fácilmente debido a que la probabilidad de que a y n sean muy diferentes a a y p es alta.

2.3.4 Procesamiento de imágenes

El objetivo del preprocesamiento es mejorar los datos de una imagen suprimiendo información no deseada o mejorando algunas características relevantes de la imagen, esto facilita el aprendizaje de los modelos.

Los principales métodos de preprocesamiento utilizados en el presente proyecto son la *umbralización* y la *detección de contornos*:

2.3.4.1 Umbralización

La *umbralización* es una técnica para la obtención imágenes binarias, es decir, imágenes cuyos píxeles sólo tienen dos valores de intensidad posibles [30]. Habitualmente se utiliza para separar los objetos en primer plano del fondo y es un método preliminar que se usa antes de otras técnicas de preprocesamiento.

Existen múltiples algoritmos de *umbralización*, uno de los más usados es el método *Otsu* [31], el cual busca iterativamente el umbral que minimiza la varianza dentro de cada clase (fondo y primer plano), definida como una suma ponderada de las varianzas de las dos clases.

2.3.4.2 Detección de contornos

La *detección de contornos* es una técnica que permite segmentar partes de una imagen binaria. Los contornos consisten en curvas que unen el conjunto de puntos que encierran un área con color o intensidad homogénea. Unos de los algoritmos más usados es el propuesto por Suzuki [32], el cual fue uno de los primeros algoritmos en establecer relaciones jerárquicas entre los bordes y diferenciar bordes internos y externos.

2.3.5 Detección de texto

La detección de texto es una técnica en la que se buscan regiones de texto en una imagen y se trazan cuadros delimitadores a su alrededor.

EAST, acrónimo de "*Efficient and Accurate Scene Text Detection"* o "*Detección de texto de escena eficiente y precisa"* en español, es un modelo para detección de regiones de texto en imágenes introducido basado en redes neuronales profundas [33]. La salida del modelo consiste en las coordenadas de los recuadros que contienen las regiones de texto.

3 DATOS

3.1 Adquisición de los datos

Para la ejecución del proyecto se obtuvieron 3 fuentes de datos:

i. El conjunto de datos de marcas registradas *METU*, el cual consiste en un conjunto de imágenes de marcas comerciales reales construido en [19] por un grupo de investigación de la Universidad Técnica de Medio Oriente en Turquía, orientado al estudio de la búsqueda y recuperación de imágenes de logos. El conjunto de datos está disponible bajo solicitud y el acceso para los fines del presente proyecto fue autorizado por el Dr. Sinan Kalkan, profesor del Departamento de Ingeniería Informática de la Universidad mencionada. El dataset sólo podrá ser usado para fines investigativos y académicos.

ii. El dataset *Logos-2K* construido en [34], un conjunto de datos de logos del mundo real orientado al problema de clasificación de imágenes de logos. El acceso al conjunto de datos es público y de libre uso.

iii. Un conjunto de datos de elaboración propia que se compone de solicitudes de registro de marca con oposición de la Superintendencia de Industria y Comercio de Colombia. Estos datos fueron recolectados de forma manual de la Gaceta de Propiedad Industrial, el medio de información oficial de la Superintendencia de Industria y Comercios, mediante el cual se dan a conocer las solicitudes presentadas y los títulos otorgados en relación con marcas, patentes de invención y diseños industriales.

3.2 Descripción de los datos

3.2.1 Conjunto de imágenes para entrenamiento

El dataset *METU* está compuesto por 2 subconjuntos, un subconjunto llamado “de consulta”, compuesto por imágenes clasificadas en 45 categorías y un subconjunto no clasificado. Como en el presente trabajo el entrenamiento es supervisado sólo se tomaron de METU las imágenes del subconjunto clasificado. A este conjunto se le agregaron 104 categorías del dataset *Logos-2K*, para consolidar un conjunto de entrenamiento que consta de 1352 imágenes en 149 categorías. Las imágenes dentro de una misma categoría corresponden a imágenes identificadas por un experto como similares.

Figura 6 – muestra de imágenes del conjunto para entrenamiento

3.2.2 Conjunto de oposiciones para validación

Este conjunto de datos se construyó a partir de solicitudes de oposición de la Oficina Virtual de Propiedad Industrial de la Superintendencia de Industria y Comercio de Colombia. Se recolectaron 480 marcas, correspondientes a 240 casos de oposición. A continuación se describen las variables del conjunto:

Variable	Descripción	Tipo de dato
Número de oposición	Número identificador de la solicitud de oposición	Texto
Naturaleza del signo solicitante	Tipo del signo solicitante: nominativo (sólo texto), figurativo (sólo imagen), mixto (imagen y texto)	Categórico
Denominación del signo solicitante	Nombre de la marca solicitante	Texto
Componente figurativo del signo solicitante	Logo de la marca solicitante	Imagen
Naturaleza del signo opositor	Tipo del signo opositor: nominativo (sólo texto), figurativo (sólo imagen), mixto (imagen y texto)	Categórico
Denominación del signo opositor	Nombre de la marca opositora	Texto
Componente figurativo del signo opositor	Logo de la marca opositora	Imagen
Resolución	Decisión de la SIC sobre la similitud de las marcas: similares o diferentes	Categórico

Tabla 1 – descripción del dataset oposiciones

	Signo solicitante	Signo opositor	Resolución
Naturaleza	mixta	mixta	SIMILARES
Denominación	FALLEN	FF Studio F	
Componente figurativo			

Tabla 2 – muestra de un caso de oposición del dataset

De los casos de oposición recolectados 187 fueron resueltos como *similares* y 53 como *diferentes*.

3.3 Preprocesamiento de los datos

Las imágenes de los 3 conjuntos de datos se encontraban en diferentes representaciones, con el fin hacerlas comparables todas se unificaron a formato JPG, relación de aspecto 1:1 y tamaño 128x128 pixeles.

Para normalizar la relación de aspecto se tomó el valor máximo de las 2, $\max(x, y)$ de cada imagen y la diferencia con respecto al dimensión menor se repartió en los 2 extremos del eje menor, rellenándola con pixeles blancos (RGB 255,255,255), de manera que la imagen original quedara centrada.

Figura 7 – muestra de estandarización de relación de aspecto

3.4 Construcción de pares y tripletas

Como los modelos de redes siamesas se entrenan con pares y tripletas se realizaron combinaciones de las imágenes del conjunto de entrenamiento, algo que además sirvió como método de aumentación de datos.

3.4.1 Pares para red siamesa con pérdida contrastiva

Para crear el conjunto de pares se realizaron todas las posibles combinaciones de imágenes de una misma categoría y adicionalmente para cada imagen del conjunto se seleccionó aleatoriamente una imagen diferente, es decir de otra categoría, obteniendo 11846 pares de imágenes, 50% similares y 50% diferentes. Se agregó además una etiqueta $Y = 0$ para los pares similares y $Y = 1$ para los pares diferentes.

Figura 8 – muestra de pares de imágenes para red siamesa con pérdida contrastiva

3.4.2 Tripletas para red siamesa con pérdida triple

Según el artículo de *Facenet* [29] la forma de muestrear las imágenes al implementar la pérdida triple tiene un gran impacto en el desempeño del modelo. La recomendación es seleccionar tripletas conocidas como *semi-difíciles*, estas se definen como tripletas donde la imagen negativa está más lejos de la imagen ancla que la imagen positiva, pero aún produce una pérdida L positiva porque se encuentra dentro del margen α. Si la imagen negativa es muy diferente a la imagen ancla, la pérdida será 0 y la red no logrará aprender.

$$L = max\ (d(a,p)\ -\ d(a,n)\ +\ \alpha, 0)$$

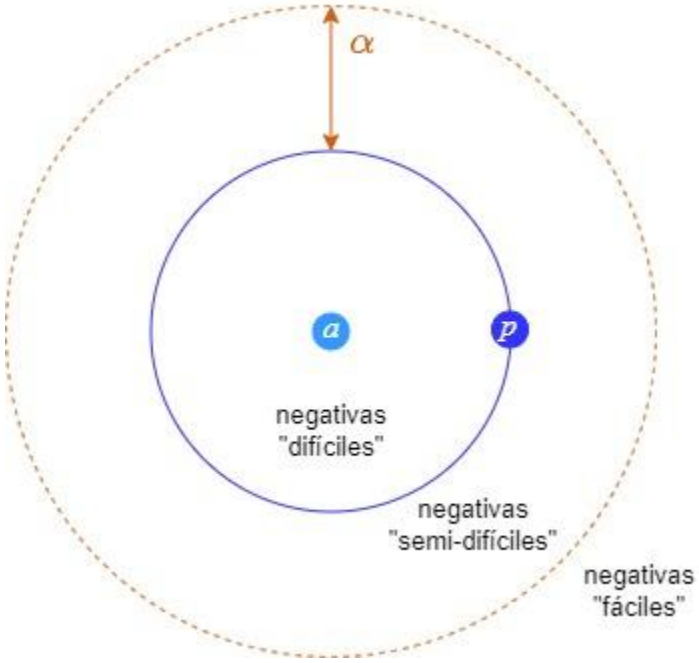


Figura 9 – tipo de tripletas según la pérdida

Para crear el conjunto de tripletas se realizaron todas las posibles combinaciones de imágenes de una misma categoría, creando de esta forma pares de imagen ancla e imagen positiva. Posteriormente para cada uno de estos pares se obtuvieron los vectores de atributos y se iteró aleatoriamente entre las imágenes de las demás categorías del dataset hasta encontrar una imagen cuyo vector de características satisficiera la siguiente condición para obtener una tripleta negativa “semi-dificil”:

$$d(a,p) < d(a,n) < d(a,p) + \alpha$$

ancla **positiva** **negativa**

Figura 10 – muestra de muestra de tripleta de imágenes para red siamesa con pérdida triple

4. METODOLOGÍA

4.1 Metodología de trabajo

La investigación se realizó bajo la metodología *CRISP-DM* (*Cross-Industrie Standard Process for Data Mining*), la cual consiste en un enfoque para responder problemas mediante modelos de datos, independiente del caso de uso o la industria en la que se realice la investigación.

A continuación se describen las actividades realizadas en las fases del ciclo *CRISP-DM* exceptuando la fase de despliegue debido a que el alcance del presente proyecto se limitó a un prototipo y la medición de su desempeño.

I - Comprensión del negocio	II - Compresión de los datos	III - Preparación de los datos
• Se definieron los objetivos del modelo. • Se investigaron los criterios en la legislación colombiana bajo los cuales puede existir un conflicto por similitud entre 2 marcas.	• Se realizó un análisis descriptivo de los datos en el que se determinó el preprocesamiento y la preparación requerida, como se describe en el capítulo 3.2.	• Se realizó una estandarización de los datos como se describe en el capítulo 3.3. • Se construyeron pares y tripletas para el modelado como se describe en el capítulo 3.4.

IV-Modelado	V-Evaluación
• Se dividió el conjunto de imágenes en 3 subconjuntos para entrenamiento, validación y prueba. • Se entrenaron diferentes modelos de similitud figurativa y se probó su respectivo ajuste con el subconjunto de prueba. • Se seleccionó el modelo con mejor desempeño para probarlo junto con las técnicas de similitud fonética y ortográfica en el conjunto de oposiciones de la SIC.	• Se probaron diferentes técnicas de similitud fonética y ortográfica en conjunto con el modelo seleccionado de similitud figurativa sobre el conjunto de oposiciones de la SIC. • Con base en los resultados se generaron etiquetas de predicción *similares* o *diferentes*, las cuales se compararon con las etiquetas reales del conjunto de oposiciones para medir el desempeño de la metodología.

4.2 Modelado y evaluación

4.2.1 Entrenamiento y prueba de modelos de similitud figurativa

Primero se realizó el entrenamiento de varios modelos de similitud figurativa con el dataset de imágenes construido a partir de *METU* y *Logos-2K*. Las técnicas de similitud ortográfica y fonética no requieren entrenamiento y por lo tanto se probaron directamente en la fase de evaluación sobre dataset de la SIC.

Como línea base se partió de los modelos pre-entrenados con *ImageNet* de las arquitecturas *VGG16*, *ResNet50V2* y *Xception*, omitiendo sus capas de clasificación para utilizarlos como generadores de atributos para después estimar la similitud entre imágenes. Es decir, conocimiento por transferencia sin ningún ajuste.

Después de esto se probaron los mismos modelos reentrenando los pesos de las últimas capas mediante las arquitecturas siamesas (triple y contrastiva). Para esto los conjuntos de tripletas y pares se dividieron en 3 subconjuntos: entrenamiento 64%, validación 16% y prueba 20%, procurando conservar una proporción de observaciones similares a diferentes de aproximadamente 1:1 en cada subconjunto.

Todos los entrenamientos de las redes neuronales se realizaron con 15 épocas y lotes de 30 observaciones y para reducir los tiempos de entrenamiento se utilizaron entornos de ejecución de Google Colab aprovisionados con 27GB de memoria RAM y aceleración con GPU.

4.2.2 Métricas de evaluación

Los resultados se evaluaron desde la perspectiva de un problema de clasificación binaria en el que las clases corresponden a las posibles resoluciones de un proceso de oposición de registro, es decir, las marcas son similares o diferentes.

El desempeño de los modelos se determinó mediante el *Valor-F*, una métrica para medir calidad de la clasificación que pondera las métricas de *precisión* y *exhaustividad*. Estas métricas parten del concepto de la *matriz de confusión*, que en el caso binario consiste en una tabla con las 4 combinaciones entre los valores predichos y reales del conjunto de pruebas.

		Predicción	
		Positivo (similares)	Negativo (diferentes)
Realidad	Positivo (similares)	*VP* Verdadero Positivo	*FP* Falso Positivo
	Negativo (diferentes)	*FN* Falso Negativo	*VN* Verdadero Negativo

Figura 11 – matriz de confusión

Precisión: indica la porción de pares de marcas que realmente son similares, del total de pares que el modelo predijo como similares.

$$precisión = \frac{VP}{VP + FP}$$

Exhaustividad: indica del total de marcas que realmente son similares, el porcentaje que el modelo es capaz de detectar como similares.

$$exhaustividad = \frac{VP}{VP + FN}$$

Valor-F: es una métrica que permite medir precisión y exhaustividad simultáneamente.

$$Valor\ F = \frac{2 * precisión * exhaustividad}{precisión + exhaustividad}$$

4.2.3 Evaluación de los 3 tipos de similitud en el conjunto de la SIC

La evaluación sobre el conjunto de la SIC, se realizó igualmente como un problema de clasificación binaria. Para orquestar el resultado final de los 3 tipos de similitud se consideró que un par de marcas son similares si cualquiera de los 3 criterios de evaluación (figurativo, ortográfico o fonético) es positivo. Esto debido a que la norma establece que la semejanza entre 2 signos puede derivarse de alguno o de los 3 criterios de similitud.

Es posible que una marca sólo tenga uno de los componentes, nominativo o gráfico. En ese caso sólo se comparó el componente común. Es decir, si una marca es mixta (tiene logo y nombre) y se compara con una marca figurativa (sólo tiene logo), sólo se consideró la similitud figurativa o gráfica.

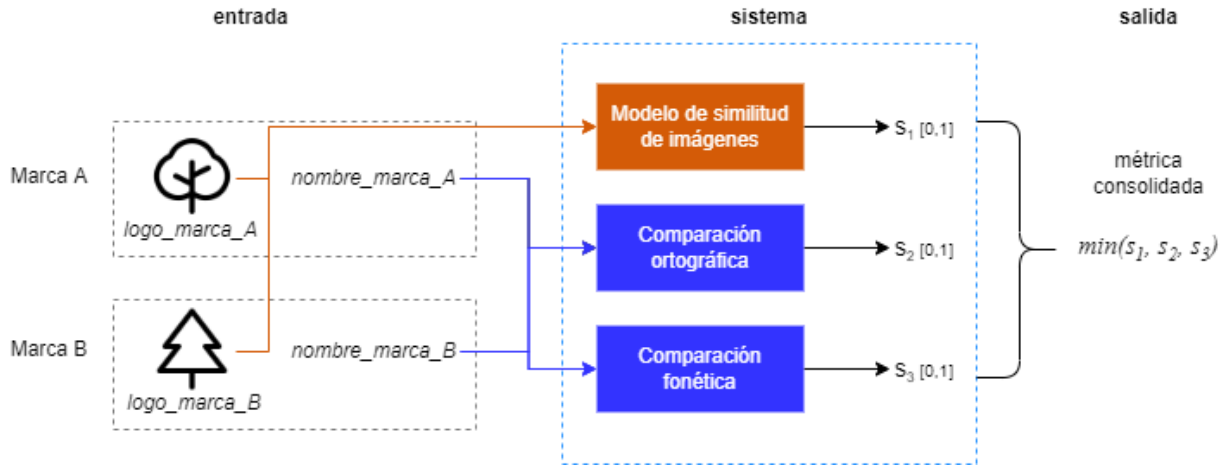


Figura 12 – diagrama de flujo de una evaluación de similitud

5 DESARROLLO DE LOS MÉTODOS Y RESULTADOS

5.1 Entrenamiento de métodos para similitud figurativa

5.1.1 Transfer learning

Como línea base se partió de los modelos pre-entrenados con *ImageNet*, omitiendo sus capas de clasificación y comparando los vectores resultantes mediante la distancia coseno definida como:

$$distancia\cos = 1 - \frac{u \cdot v}{\|u\|_2 \|v\|_2}$$

Si bien esta distancia no es estrictamente una métrica, pues no satisface la desigualdad triangular, para este caso es útil como indicador normalizado de la similitud entre los vectores

resultantes de los modelos, pues todos los elementos de los vectores de atributos que se obtienen de los modelos son positivos, de manera que el resultado estará entre 0 y 1.

El umbral de decisión se definió en 0.5, así los pares con distancia mayor o igual que el umbral se clasificaron como “diferentes” y los pares con valor de similaridad menor que el umbral se clasificaron como “similares”.

A continuación se presentan los resultados de la comparación de las primeros 12 pares del conjunto de prueba con el modelo de línea base VGG16 de transfer learning:

Figura 13 – muestra de resultados del conjunto de pruebas con transfer learning VGG16

En la figura 13 se puede observar que el modelo clasificó incorrectamente el primer par pues la distancia coseno entre los vectores de atributos de las imágenes es de 0.091. Con un umbral de decisión de 0.5 la etiqueta predicha es 0 (similares), cuando la etiqueta real es 1 (diferentes) y se puede observar que los logos en cuestión no se parecen. Por el contrario, el modelo clasificó correctamente el segundo par de logos cuya distancia coseno fue 0.100, es decir, el modelo clasificó las imágenes como similares y en realidad lo son.

A continuación se presentan los resultados con el conjunto de prueba de los 3 modelos:

Modelo	Valor-F transfer learning
ResNet50v2	0.73
Xception	0.71
VGG16	0.68

Tabla 3 – resultados transfer learning

Los 3 modelos tuvieron un desempeño similar con un valor-F entre 0.68 y 0.73, que en este caso es un desempeño medio. Estos valores se utilizaron como línea base para comparar y el desempeño podía mejorarse mediante otras técnicas o modelos.

5.1.2 Transfer learning con procesamiento de imágenes

Después de inspeccionar manualmente una muestra aleatoria de 50 pares de imágenes clasificados incorrectamente por los modelos de transfer learning (principalmente falsos negativos), se encontró que aproximadamente el 70% de los casos cumplía con alguna de las siguientes condiciones:

i. Había texto en la imagen.
ii. Había formas relevantes para la diferenciación contenidas de otras formas.

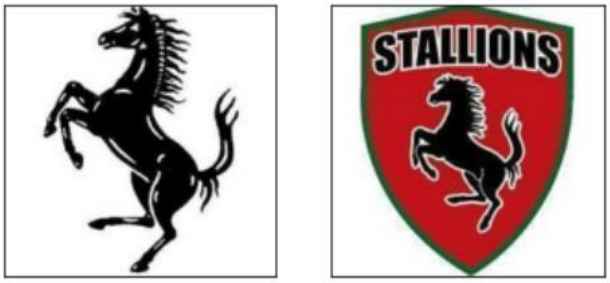


Figura 14 – ejemplo imagen con texto y el caballo contenido dentro de un escudo

Con el fin de reducir los falsos negativos se probó una estrategia utilizando métodos de procesamiento antes de entregar las imágenes a los modelos. Para esto se definieron las siguientes instancias de preprocesamiento, buscando hacer énfasis en diferentes aspectos diferenciadores de la imagen:

- **Instancia1 - imagen original**: imagen original en RGB.
- **Instancia2 - imagen binaria:** imagen umbralizada en blanco y negro mediante el método *Otsu* [31].
- **Instancia3 - contornos de la imagen:** contornos detectados en la imagen mediante el método *Suzuki* [32], dibujados en color blanco sobre un fondo negro del mismo tamaño de la imagen original.
- **Instancia4 - contornos con altura significativa:** contornos detectados en la imagen mediante el método *Suzuki,* cuya dimensión en el eje Y fuera mayor que 23 pixeles.
- **Instancia5 - contornos con área significativa:** contornos detectados en la imagen mediante el método *Suzuki,* cuya área sea al menos de 86 pixeles cuadrados, esto equivales a contornos que ocupen como mínimo 5.3% de la imagen.
- **Instancia6 - contornos sin texto:** contornos resultantes después de un proceso de remoción de texto. La remoción se realizó detectando regiones de texto mediante el modelo *EAST* [33], cuya salida consiste en las coordenadas de los recuadros que contienen las regiones de texto. Para eliminar los contornos que correspondían a texto o letras se iteraron todos los contornos de la imagen, contando la cantidad de puntos contenidos dentro de una región de texto, los contornos con al menos 60% de sus puntos contenidos dentro de una región de texto se excluyeron de la imagen.
- **Instancia7 - contornos individuales:** contornos detectados en la imagen mediante el método *Suzuki* [32], separados y dibujados individualmente sobre canvas negros del mismo tamaño de la imagen original y escalados conservando su relación de aspecto para abarcar al menos el 90% de una de las dimensiones X ó Y. El propósito con estas imágenes fue hacer un análisis de la similitud de las combinaciones posibles de los contornos individuales del par de imágenes.

Para medir la similitud de 2 imágenes se realizó el procesamiento de cada una ellas hasta las instancias descritas anteriormente. Luego se obtuvieron los vectores de atributos de cada instancia utilizando los modelos de transfer learning y posteriormente se compararon los vectores de las instancias correspondientes con la distancia coseno.

Figura 15 – ejemplo de las instancias resultantes de una imagen

Finalmente, para consolidar las 7 distancias obtenidas se realizó una suma ponderada:

$$\sum_{i=1}^{7} x_i * w_i$$

A las instancias 1 a 6 se les dio un peso de 0.1 y la instancia 7 un peso de 0.4. Se le otorgó un mayor peso a la instancia 7, ya que en esta se hace la comparación más exhaustiva pues analiza la similitud entre todas las combinaciones de contornos. El resultado final es un valor entre 0 y 1 con la misma interpretación anterior donde 0 es “similar”, 1 es “diferente” y el umbral de decisión es de 0.5.

Los resultados con el conjunto de pruebas fueron los siguientes:

Modelo	Valor-F transfer learning	Valor-F transfer learning sobre instancias
ResNet50v2	0.73	0.65 ↓
Xception	0.71	0.56 ↓
VGG16	0.68	0.77 ↑

Tabla 4 – resultados línea base vs transfer learning sobre instancias

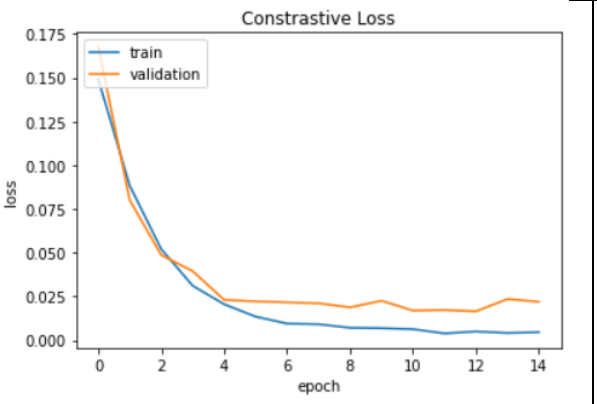
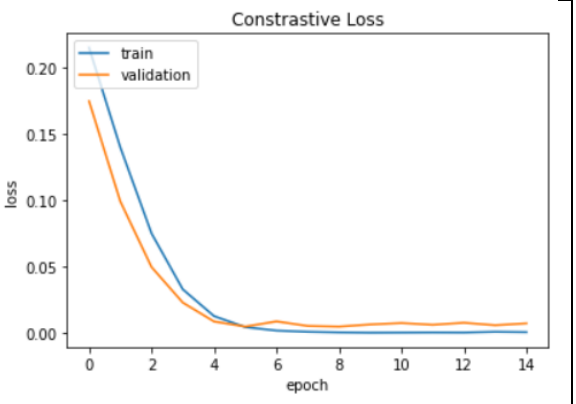
En los resultados se observa que sólo el modelo *VGG16* mejoró con las instancias de preprocesamiento con respecto a los modelos de línea base. Los modelos *ResNet50v2* y *Xception* tuvieron un desempeño inferior al obtenido con las imágenes sin el preprocesamiento propuesto

Al analizar los resultados se encontró que el motivo por el que los modelos con arquitecturas *ResNet50v2* y *Xception* tuvieron un desempeño menor es porque en la instancia 7 encontraron similitudes en formas básicas (como círculos y cuadrados), que al estar aisladas del resto de los contornos no tenían elementos de diferenciación y por ende los modelos clasificaron la mayoría de las imágenes como similares, generando una alta tasa de falsos negativos.

5.1.3 One-shot learning

5.1.3.1 Red siamesa con pérdida contrastiva

Las últimas capas de los 3 modelos objetivo se re-entrenaron con el dataset de pares bajo una arquitectura siamesa de pérdida contrastiva. A continuación, se presentan las curvas de aprendizaje (comportamiento de la pérdida) de los modelos medida en los conjuntos de entrenamiento y validación.

VGG16	ResNet50v2
	

Xception

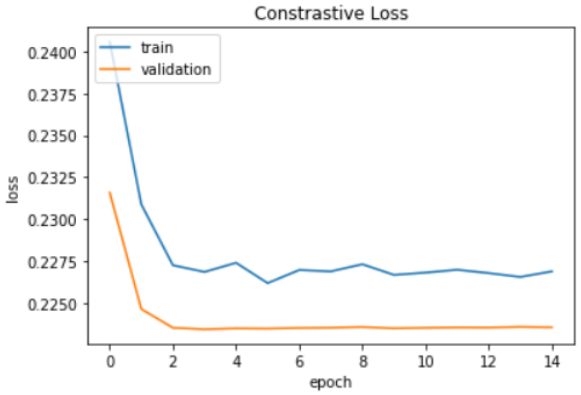


Figura 16 – resultado entrenamiento modelos con pérdida contrastiva

En las 3 curvas se observa que en los modelos la pérdida disminuyó en la medida que fueron avanzando los pasos de entrenamiento y se estabilizaron entre 4 y 6 épocas. En los modelos *VGG16* y *ResNet50v2* la pérdida logró reducirse a niveles cercanos a 0, mientras que *Xception* sólo alcanzó un nivel de ~0.22.

El desempeño de cada modelo sobre el conjunto de pruebas fue el siguiente:

Modelo	Valor-F transfer learning	Valor-F Red siamesa pérdida contrastiva
ResNet50v2	0.73	**0.98** ↑
Xception	0.71	**0.63** ↓
VGG16	0.68	**0.97** ↑

Tabla 5 – resultados línea base vs redes siamesas con pérdida contrastiva

Se puede observar que en los modelos *ResNet50v2* y *VGG16* incrementó significativamente el desempeño con respecto a los modelos base de transfer learning sin re-entrenamiento, alcanzando valores-F cercanos a 1, lo que significa que los modelos fueron capaces de clasificar los pares de logos como similares o diferentes con un alto grado de precisión y exhaustividad. Por el contrario, en el caso de *Xception* el desempeño disminuyó con respecto al modelo base de transfer learning.

A continuación se presentan los resultados de la comparación de los primeros 12 pares del conjunto de prueba con el modelo *VGG16* después de ser re-entrenado con la arquitectura siamesa de pérdida contrastiva:

Figura 17 – muestra de resultados del conjunto de pruebas con pérdida contrastiva en VGG16

En los 12 pares de la muestra el modelo logró clasificar correctamente si los logos eran similares o diferentes.

5.1.3.2 Red siamesa con pérdida triple

Los 3 modelos objetivo también se re-entrenaron en sus últimas capas con el dataset de tripletas mediante una arquitectura siamesa de pérdida triple. A continuación, se presentan las curvas de aprendizaje (comportamiento de la pérdida) de los modelos medida en los conjuntos de entrenamiento y validación.

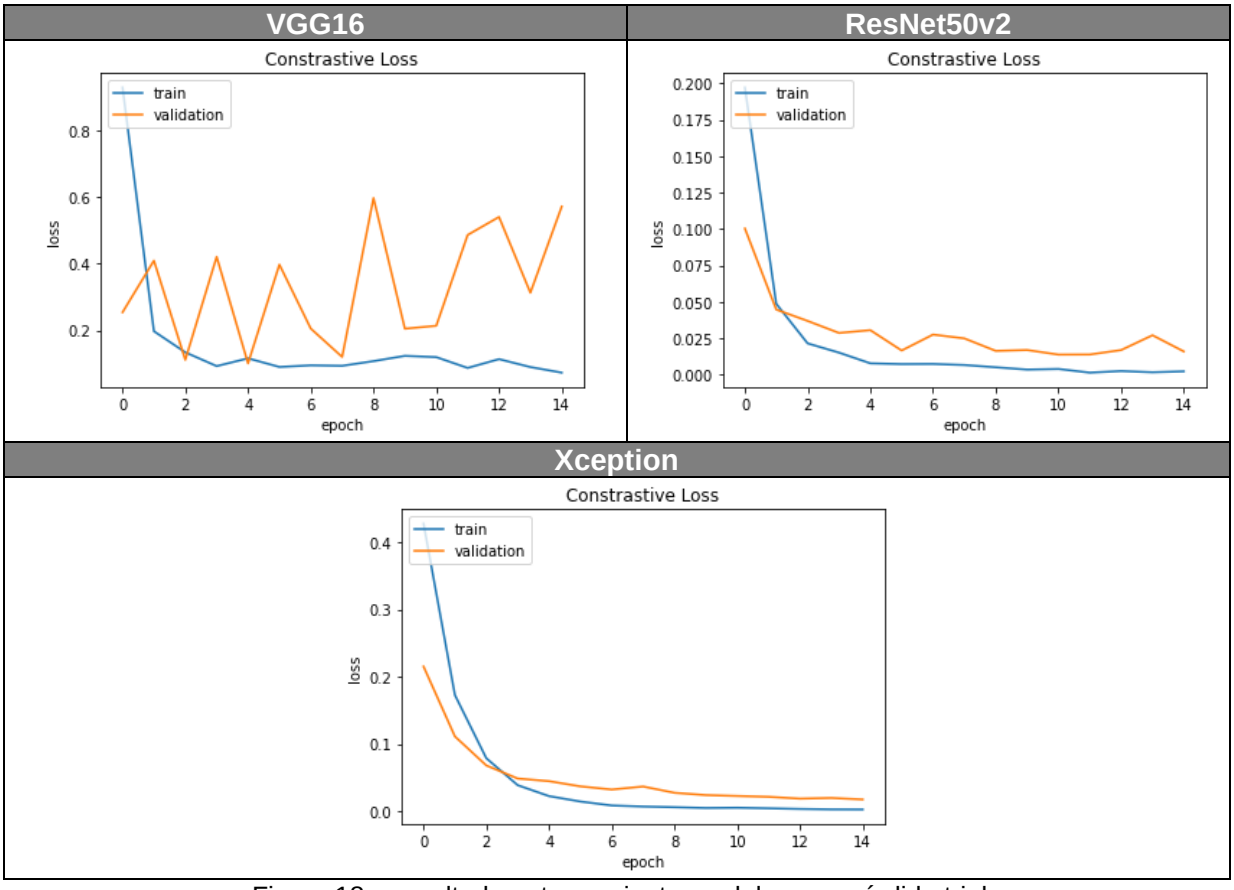


Figura 18 – resultado entrenamiento modelos con pérdida triple

En las curvas se observa cómo disminuyó y se estabilizó la pérdida de los modelos *ResNet50v2* y *Xception* tanto para el conjunto de entrenamiento como para el de validación. En el caso de *VGG16* se observa que a pesar de que la pérdida de entrenamiento se estabilizó un valor cercano a 0, la pérdida para el conjunto de validación nunca se estabilizó y fue incrementando a medida que el entrenamiento avanzaba en épocas.

Los resultados de los modelos sobre el conjunto de pruebas fueron los siguientes:

Modelo	Valor-F transfer learning	Valor-F Red siamesa pérdida triple
ResNet50v2	0.73	**0.94** ↑
Xception	0.71	**0.96** ↑
VGG16	0.68	**0.82** ↑

Tabla 6 – resultados línea base vs redes siamesas con pérdida triple

En los 3 modelos se observa una mejora significativa con respecto a los modelos de línea base. *ResNet50v2* y *Xception* tuvieron un muy buen desempeño alcanzando valores-F cercanos a 1, lo que significa que los modelos fueron capaces de clasificar los pares de logos como similares o diferentes con un alto grado de precisión y exhaustividad.

A continuación se presentan los distancias entre imagen ancla-positiva e imagen ancla-negativa de una muestra de las primeras 12 tripletas del conjunto de prueba con el modelo *ResNet50v2* después de ser re-entrenado con la arquitectura siamesa de pérdida triple:

Figura 19 – muestra de resultados del conjunto de pruebas con pérdida triple en ResNet50v2

En todas las tripletas de la muestra se observa que el modelo clasifico correctamente las imágenes similares y clasificó correctamente 10 de los 12 pares diferentes.

5.1.3.3 Resultados generales de los modelos de similitud figurativa

En la siguiente tabla se presentan los resultados sobre el conjunto de pruebas de todos los modelos que se entrenaron y probaron con su respectiva métrica de evaluación (Valor-F):

Modelo	Valor-F
ResNet50v2 - siamés pérdida contrastiva	0.98
VGG16 - siamés pérdida contrastiva	0.97
Xception - siamés pérdida triple	0.96
ResNet50v2 - siamés pérdida triple	0.94
VGG16 - siamés pérdida triple	0.82
VGG16 - transfer learning + instancias	0.77
ResNet50v2 - transfer learning (línea base)	0.73
Xception - transfer learning (línea base)	0.71
VGG16 - transfer learning (línea base)	0.68
ResNet50v2 - transfer learning + instancias	0.65
Xception - siamés pérdida contrastiva	0.63
Xception - transfer learning + instancias	0.56

Tabla 7 – resultados comparativos de los modelos de similitud figurativa

- En los resultados se observa que en las arquitecturas siamesas 4 de los 6 modelos tuvieron desempeños excelentes, valores-F por encima de 0.94.
- Salvo el modelo *Xception* con pérdida contrastiva, todos los demás modelos re-entrenados con arquitecturas siamesas mejoraron significativamente con respecto a los modelos de transfer learning de línea base.
- Ninguno de los modelos que utilizó las instancias tuvo un buen desempeño.
- El modelo de mejor desempeño *ResNet50v2* re-entrenado en la arquitectura siamesa con pérdida contrastiva (valor-F 0.98) será el usado para las pruebas con el conjunto de la SIC.

5.2 Métodos para similitud ortográfica y fonética

La similitud ortográfica y fonética de las marcas se midió utilizando algoritmos de procesamiento del lenguaje natural que no requieren entrenamiento, por lo tanto, se aplicaron directamente sobre el conjunto de datos SIC.

Para el caso de la similitud ortográfica se probaron 3 algoritmos:

i. Un método basado en caracteres, la *distancia de Levenshtein* [35].
ii. Un método basado en términos, el *coeficiente de superposición*, también conocido como *coeficiente de Szymkiewicz-Simpson* [36].
iii. Un método híbrido que combina los beneficios de los métodos basados en caracteres y los métodos basados en texto, la *similitud de Monge-Elkan* [37].

Como el conjunto de datos está etiquetado con 0 para marcas "similares" y 1 para marcas "diferentes", los valores obtenidos de los algoritmos se adaptaron de la siguiente forma para que denoten la similitud de la misma forma que el conjunto:

- La *distancia de Levenshtein* se normalizó dividiendo la distancia entre la longitud máxima de caracteres entre las 2 cadenas de texto a comparar.

$$d(A,B) = \frac{d_{Levenshtein}\,(A,B)}{\max\,(longitud(A), longitud(B))}$$

- El *coeficiente de superposición* se expresó como:

$$d(A,B) = 1 - \ coeficiente\ de\ superposición\ (\mathrm{A},\mathrm{B})$$

- La *similitud de Monge-Elkan* se expresó como:

$$d(A,B) = 1 - \ sim_{MongeElkan}(\mathrm{A},\mathrm{B})$$

El componente nominativo de una marca puede estar compuesto por varias palabras, así que para la comparación con la *distancia de Levenshtein* se tokenizaron los textos a nivel de palabras (separación por espacios) y se calculó la *distancia de Levenshtein* entre las posibles combinaciones de pares de tokens entre las 2 marcas. De los valores obtenidos se seleccionó el menor valor como medida de similaridad entre las 2 marcas. Para la medición con el *coeficiente de superposición* y la *similitud de Monge-Elkan* se utilizaron los textos completos de las marcas.

Para el caso de la similitud fonética se probaron 3 algoritmos:

i. *PhoneticSpanish* [13]
ii. *Spanish Metaphone* [14]
iii. *MetaSoundex* [15]

Los 3 métodos fonéticos se utilizaron sobre los nombres de las marcas tokenizados. Para esto se calculó la *distancia de Levenshtein* entre las posibles combinaciones de pares de codificaciones fonéticas de tokens.

	Signo solicitante A	Signo opositor B
Denominación	King Andina	Vidrio Andino
Codificación de los tokens con Spanish Metaphone	['KNG', 'ANDN']	['VDR', 'ANDN']
Combinaciones de pares resultantes	[['KNG', 'VDR'], ['KNG', 'ANDN'], ['ANDN', 'VDR'], ['ANDN', 'ANDN']]	
Distancia de Levenshtein normalizada entre pares	[1.0, 0.75, 0.75, 0.0]	
$\mathbf{min}\ (\boldsymbol{d_{Levenshtein}}\ (\boldsymbol{V}))$	0	

Tabla 8 – ejemplo comparación fonética

En todos los casos el umbral de decisión se definió en 0.5, así los pares con valores de similaridad mayor o igual que el umbral se clasificaron como “diferentes” y los pares con similaridad menor que el umbral se clasificaron como “similares”.

5.3 Resultados con el conjunto SIC

Sobre el conjunto de la SIC se probó el sistema de medición de 3 componentes (figurativo, ortográfico y fonético) ilustrado anteriormente en la figura 12, utilizando los 3 algoritmos de similaridad ortográfica y los 3 algoritmos de similaridad fonética anteriormente mencionados, más el modelo de mejor desempeño de similitud visual (*ResNet50v2* re-entrenado en la arquitectura siamesa con pérdida contrastiva) y el modelo de similitud visual de mejor desempeño entrenado con la arquitectura siamesa de pérdida triple (*Xception*).

Se probaron todas las combinaciones entre algoritmos fonéticos y ortográficos con el modelo visual *ResNet50v2* y todas las combinaciones con los 2 modelos visuales *ResNet50v2* y *Xception* en donde si alguno de los 2 modelos indicaba similitud se emitía la predicción de similitud para el componente figurativo (es decir, siempre se tomaba la predicción mínima).

A continuación se presentan los resultados de todas las posibles combinaciones:

Métodos de similitud	Valor-F
Overlap, SpanishMetaphone, ResNet50v2	0.810
Levenshtein, SpanishMetaphone, ResNet50v2	0.806
MongeElkan, SpanishMetaphone, ResNet50v2	0.801
Overlap, PhoneticSpanish, ResNet50v2 + Xception	0.801
Overlap, SpanishMetaphone, ResNet50v2 + Xception	0.801
MongeElkan, PhoneticSpanish, ResNet50v2 + Xception	0.801
MongeElkan, SpanishMetaphone, ResNet50v2 + Xception	0.801
Overlap, PhoneticSpanish, ResNet50v2	0.799
MongeElkan, MetaSoundex, ResNet50v2 + Xception	0.798
Levenshtein, PhoneticSpanish, ResNet50v2	0.797
Levenshtein, MetaSoundex, ResNet50v2	0.796
Levenshtein, PhoneticSpanish, ResNet50v2 + Xception	0.794
Levenshtein, SpanishMetaphone, ResNet50v2 + Xception	0.794
MongeElkan, MetaSoundex, ResNet50v2	0.793
MongeElkan, PhoneticSpanish, ResNet50v2	0.792
Levenshtein, MetaSoundex, ResNet50v2 + Xception	0.791
Overlap, MetaSoundex, ResNet50v2 + Xception	0.791
Overlap, MetaSoundex, ResNet50v2	0.790

Tabla 9 – resultados comparación conjunto de datos SIC

- Se observa un desempeño similar entre todas las combinaciones de técnicas con un valor-F ente 0.79 y 0.81.
- En los algoritmos de similitud fonética orientados únicamente a español (*PhoneticSpanish* y *SpanishMetaphone*) se observa un mejor desempeño con respecto a *MetaSoundex* que soporta español e inglés.

Complementariamente es interesante analizar las matrices de confusión, a continuación, se presentan las matrices de la combinación de mejor desempeño (*Overlap, SpanishMetaphone, ResNet50v2*) y de la misma combinación pero con los 2 modelos de similitud visual (*Overlap - SpanishMetaphone - ResNet50v2 + Xception*):

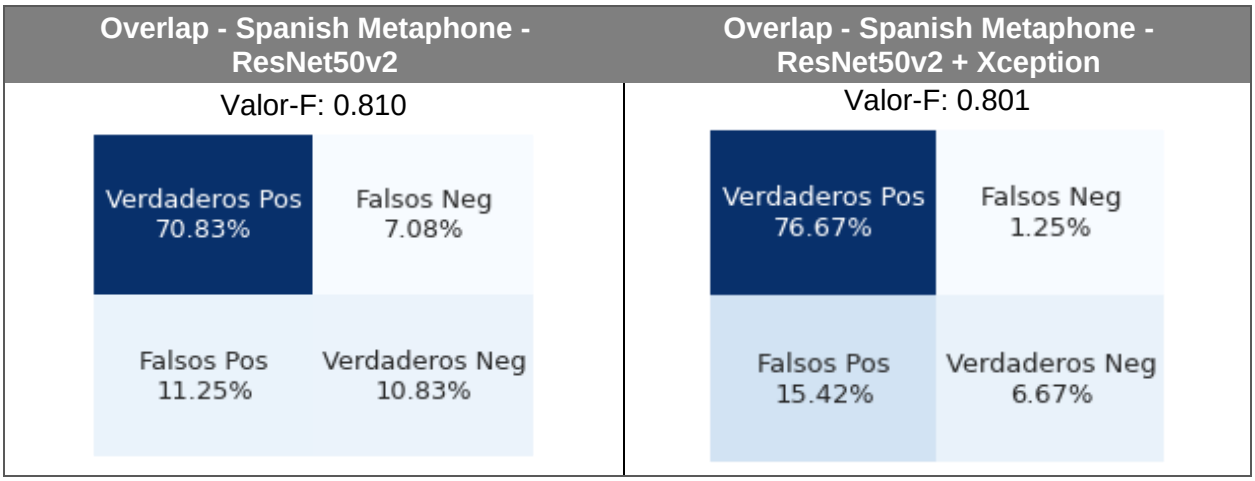


Figura 20 – matrices de confusión de los modelos relevantes

- Ambos sistemas tienen valores-F muy similares, sin embargo, el sistema que combina los 2 modelos visuales (*ResNet50v2 + Xception*) detecta mejor la similitud visual reduciendo la tasa de Falsos Negativos de 7.08% a 1.25% sin incrementar significativamente la tasa de Falos Positivos. En un escenario de automatización de vigilancia marcaria esta reducción de Falsos Negativos es muy relevante.
- La tasa de Falsos Positivos está explicada en parte por términos similares o iguales en los nombres de las marcas, por los que el sistema clasifica las marcas como parecidas pero que para la SIC no son causales de similitud debido a que son palabras genéricas dentro de la categoría de los productos o servicios que identifican.

6 CONCLUSIONES

6.1 Discusión de los resultados

En la exploración de modelos de similitud figurativa se puede observar que el aprendizaje por transferencia proporcionó un buen punto de partida, arrojando valores-F entre 0.68 y 0.73 entre los 3 modelos probados, la cual permitió desarrollar rápidamente modelos de línea base eficaces sin la necesidad de contar grandes recursos, datos y tiempo de entrenamiento.

La estrategia que se buscó con las instancias de preprocesamiento para reducir los falsos negativos derivó en una mayor cantidad de falsos positivos. Esto debido a que la instancia 7, en la que se independizaban y escalaban los contornos para compararlos, funcionaba muy bien con contornos complejos, pero encontraba una similitud alta entre formas básicas (como por ejemplo círculos y triángulos). Esto resulto en un mal desempeño predictivo (valores-F entre 0.56 y 0.77), por debajo incluso de los modelos de línea base. Aunque el rendimiento no hacía parte del alcance de este trabajo, vale la pena mencionar que se observó que esta técnica consumía más tiempo para analizar la misma cantidad de datos que los modelos de línea base.

Para finalizar en lo referente a la similitud figurativa, se observó que 5 de los 6 modelos probados mejoraron significativamente en desempeño gracias al re-entrenamiento de las últimas capas con las arquitecturas siamesas. La combinación de aprendizaje por transferencia con arquitecturas siamesas resultó en modelos con un alto poder predictivo (valores-F entre 0.94 y 0.98).

En la arquitectura siamesa con pérdida triple se observó más consistencia que en la de pérdida contrastiva, ya que los 3 de los 3 modelos con pérdida triple mejoraron con respecto a los modelos de línea base, a diferencia de los de pérdida contrastiva donde 2 de 3 mejoraron y 1 estuvo por debajo de su respectivo modelo de línea base.

En cuanto a las métricas de similitud ortográfica se observa un desempeño homogéneo entre las técnicas probadas, pues dejando constantes los modelos de similitud visual y fonética y probando las 3 métricas ortográficas el desempeño del sistema completo es muy parecido.

En las métricas de similitud fonética se observa un mejor desempeño del algoritmo *Spanish Metaphone*, pues las combinaciones 3 con este algoritmo resultaron en los 3 sistemas con mejor desempeño (valores-F entre 0.80 y 0.81).

Es relevante mencionar que en la similitud de texto, tanto ortográfica como fonética, hay marcas que contienen palabras iguales, por lo cual el sistema predijo las marcas como similares, pero que en el contexto de signos distintivos las marcas son diferentes debido a que la palabra en común es considerada un término genérico que no aporta distintividad. La genericidad de un término depende de la clasificación de productos que identifique la marca, por ejemplo, en la clase 45 de servicios jurídicos, la palabra “legal” es un término genérico. En casos como este el sistema clasificó incorrectamente las marcas pues no considera estas condiciones:

Marca A	Marca B	Etiqueta real	Predicción
L COMPLEMENTO **LEGAL**	CONTEXTO **LEGAL**	1 - diferentes	0 - similares

Esto resulto en una mayor tasa de falsos positivos y por ende en un menor valor-F.

6.2 Conclusiones

En el presente trabajo se ha planteado y evaluado una metodología para establecer cuantitativamente la similitud entre marcas, la cual combina redes neuronales convolucionales y medidas de similitud del procesamiento del lenguaje natural. La metodología considera los 3 criterios de semejanza principales considerados en la legislación colombiana y los tratados internacionales como son la similitud figurativa, ortográfica y fonética. Como resultado se obtuvo un sistema que puede medir la similitud entre marcas y contribuir en la detección de conflictos de signos distintivos de propiedad industrial.

Independientemente de los algoritmos y modelos subyacentes usados en el trabajo se observa que la metodología explorada es útil para resolver el problema planteado. Esta metodología consiste en medir por separado los 3 criterios de semejanza, emitir para cada uno una clase binaria (0-similares o 1-diferentes) y si en alguno de los criterios se encuentra similitud el par de marcas se clasifica como similar.

En el caso de la semejanza figurativa las redes siamesas demostraron ser de gran utilidad, pues no requieren aprender con anterioridad el logo de una marca para determinar su similitud con otra. Este tipo de redes combinada con el apalancamiento en modelos pre-entrenados de clasificación de imágenes permitieron obtener buenos resultados con un conjunto de entrenamiento pequeño.

Si bien el sistema planteado ha mostrado un buen desempeño (valor-F de 0.81), la valoración de marcas es un proceso complejo con particularidades subjetivas que finalmente necesita de la valoración de un experto. La metodología planteada no considera por ejemplo la conexidad competitiva de los productos o servicios que identifican las marcas, ni la capacidad de diferenciación o genericidad de los términos que componen el nombre. Sin embargo, el sistema planteado puede tener aplicación asistiendo la valoración experta reduciendo significativamente el volumen de marcas a revisar en procesos de registro marcario.

7 REFERENCIAS

1. Colombia, Superintendencia de Industria y Comercio (2020). Propiedad Industrial 2020.
2. Colombia, Superintendencia de Industria y Comercio (2017). ABC de Propiedad Industrial.
3. Comunidad Andina, Tribunal de Justicia (2000). Régimen Común de Propiedad Industrial.
4. Gomaa, W. H., & Fahmy, A. A. (2013). A survey of text similarity approaches. *International Journal of Computer Applications*, *68*(13), 13-18.
5. Hall, P. A., & Dowling, G. R. (1980). Approximate string matching. ACM computing surveys (CSUR), 12(4), 381-402.
6. Winkler, W. E. (1990). String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage.
7. Barrón-Cedeno, A., Rosso, P., Agirre, E., & Labaka, G. (2010, August). Plagiarism detection across distant language pairs. In Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010) (pp. 37-45).
8. Arnold, M., & Ohlebusch, E. (2011). Linear time algorithms for generalizations of the longest common substring problem. *Algorithmica*, *60*(4), 806-818.
9. Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bull Soc Vaudoise Sci Nat, 37, 547-579.
10. Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26(3), 297-302.
11. Shah, Rima & Singh, Dheeraj. (2014). Analysis and Comparative Study on Phonetic Matching Techniques. *International Journal of Computer Applications*. 87. 10.5120/15236-3771.
12. Philips, L. (2000). The double metaphone search algorithm. *C/C++ users journal*, 18(6), 38-43.
13. Amón, I., Moreno, F., & Echeverri, J. (1). Algoritmo fonético para detección de cadenas de texto duplicadas en el idioma español. Revista Ingenierías Universidad De Medellín, 11(20), 127-138.
14. Mosquera, A., Lloret, E., & Moreda, P. (2012). Towards facilitating the accessibility of web 2.0 texts through text normalization. In *Proceedings of the LREC workshop: Natural Language Processing for Improving Textual Accessibility (NLP4ITA)* (pp. 9-14).
15. Koneru, K., & Varol, C. (2018). MetaSoundex Phonetic Matching for English and Spanish. Global Journal of Enterprise Information System, 10(1), 1-13.

16. ping Tian, D. (2013). A review on image feature extraction and representation techniques. *International Journal of Multimedia and Ubiquitous Engineering*, *8*(4), 385-396.
17. Wang, W., Yang, Y., Wang, X., Wang, W., & Li, J. (2019). Development of convolutional neural network and its application in image classification: a survey. *Optical Engineering*, *58*(4), 040901.
18. O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.
19. Tursun, O., Aker, C., & Kalkan, S. (2017). A large-scale dataset and benchmark for similar trademark retrieval. *arXiv preprint arXiv:1701.05766*.
20. Trappey, C. V., Trappey, A. J., & Lin, S. C. C. (2020). Intelligent trademark similarity analysis of image, spelling, and phonetic features using machine learning methodologies. *Advanced Engineering Informatics*, *45*, 101120.
21. Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., ... & He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, *109*(1), 43-76.
22. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3), 211-252.
23. Koch, G., Zemel, R., & Salakhutdinov, R. (2015, July). Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop* (Vol. 2).
24. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
25. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
26. He, K., Zhang, X., Ren, S., & Sun, J. (2016, October). Identity mappings in deep residual networks. In *European conference on computer vision* (pp. 630-645). Springer, Cham.
27. Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1251-1258).
28. Hadsell, R., Chopra, S., & LeCun, Y. (2006, June). Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)* (Vol. 2, pp. 1735-1742). IEEE.

29. Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 815-823).
30. Abak, A. T., Baris, U., & Sankur, B. (1997, August). The performance evaluation of thresholding algorithms for optical character recognition. In *Proceedings of the fourth international conference on document analysis and recognition* (Vol. 2, pp. 697-700). IEEE.
31. Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, *9*(1), 62-66.
32. Suzuki, S. (1985). Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, *30*(1), 32-46.
33. Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., & Liang, J. (2017). East: an efficient and accurate scene text detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition* (pp. 5551-5560).
34. Wang, J., Min, W., Hou, S., Ma, S., Zheng, Y., Wang, H., & Jiang, S. (2020, April). Logo-2K+: A large-scale logo dataset for scalable logo classification. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, No. 04, pp. 6194-6201).
35. Levenshtein, V. I. (1966, February). Binary codes capable of correcting deletions, insertions, and reversals. In Soviet physics doklady (Vol. 10, No. 8, pp. 707-710).
36. Simpson, E. H. (1949). Measurement of diversity. nature, 163(4148), 688-688.
37. Jimenez, S., Becerra, C., Gelbukh, A., & Gonzalez, F. (2009, March). Generalized mongue-elkan method for approximate text string comparison. In *International conference on intelligent text processing and computational linguistics* (pp. 559-570). Springer, Berlin, Heidelberg.