PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE MÁSTER EN SISTEMAS DE INFORMACIÓN, MENCIÓN DATA SCIENCE.

TEMA:

“SISTEMA DE RECUPERACIÓN DE IMÁGENES DE SIGNOS DISTINTIVOS, BASADO EN MACHINE LEARNING.”

AUTOR:

ING. MICHAEL JAMIL YANANGÓMEZ SUÁREZ

TUTOR:

EDISON FERNANDO LOZA AGUIRRE, PHD.

QUITO, AGOSTO DE 2024

DEDICATORIA

Quiero dedicar este trabajo a Dios, por haberme permitido hacer este sueño realidad a través del mi dedicación y empeño y a mi abnegado Padre, por ser mi guía de vida, forjador de valores y virtudes, fuente de inspiración y fortaleza en todo momento.

Michael Jamil

AGRADECIMIENTO

A Dios por la vida, a mi padre por su guía constante, a mi esposa y a mi pequeña hija, por entregarme todo su amor y cariño a lo largo de mi vida, por su gran ejemplo, compañía, y el apoyo incondicional en todas las circunstancias.

A la Pontificia Universidad Católica del Ecuador, por haberme brindado la oportunidad de poder seguir formándome como profesional.

A mi tutor, por todo su apoyo y su tiempo.

Michael Jamil

AUTORÍA DE LA INVESTIGACIÓN

La responsabilidad de las opiniones, comentarios y críticas emitidas en el trabajo de investigación con el tema "**SISTEMA DE RECUPERACIÓN DE IMÁGENES DE SIGNOS DISTINTIVOS, BASADO EN MACHINE LEARNING**", nos corresponde exclusivamente a Michael Jamil Yanangómez Suárez, Autor y Ms. Edison Loza, Director del Trabajo de Investigación; y el patrimonio intelectual del mismo a la Pontificia Universidad Católica del Ecuador.

Firmado electrónicamente por:
MICHAEL JAMIL
YANANGOMEZ SUAREZ

Firmado electrónicamente por:
EDISON FERNANDO
LOZA AGUIRRE

……………………………………

Ing. Michael J. Yanangómez S.

Autor

……………………………………

Edison Loza Aguirre, PhD

Director

DERECHOS DE AUTOR

Autorizo a la Pontificia Universidad Católica del Ecuador, para que haga de este trabajo de investigación o parte de él, un documento para su lectura, consulta y procesos de investigación, según las normas de la institución.

Cedo los Derechos de mi trabajo de investigación, con fines de difusión pública, además apruebo la reproducción de este, dentro de las regulaciones de la Universidad.

Firmado electrónicamente por:
MICHAEL JAMIL
YANANGOMEZ SUAREZ

…………………………………………

Ing. Michael Jamil Yanangómez Suárez

Tema: “Sistema de Recuperación de Imágenes de Signos Distintivos, Basado en Machine Learning.”

Autor: Ing. Michael Jamil Yanangómez Suárez

RESUMEN

La inteligencia artificial y el aprendizaje automático están revolucionando la manera en que procesamos y analizamos datos en diversas áreas, desde la medicina hasta la ingeniería y, más recientemente, en el campo de la propiedad intelectual. El aprendizaje automático, permite a los sistemas aprender y mejorar a partir de la experiencia y permiten gestionar grandes volúmenes de datos con una precisión y velocidad que superan las capacidades humanas.

La propiedad intelectual en Ecuador enfrenta desafíos únicos, especialmente en la gestión y protección de derechos intelectuales asociados a signos distintivos como los logos; tareas que están a cargo del SENADI. Tradicionalmente, la revisión de logos para garantizar su originalidad y evitar infracciones ha sido una tarea manual, llevada a cabo por expertos que pueden fácilmente verse abrumados por la creciente cantidad de datos visuales. Esta metodología no solo es propensa a errores, sino también insostenible en el largo plazo dada la alta frecuencia de solicitudes de registro.

Este proyecto busca abordar esta problemática mediante la creación de un sistema diseñado para mejorar significativamente la precisión y eficiencia en la identificación de logos similares. El proceso comenzó con la creación de un extenso repositorio de imágenes, las cuales fueron normalizadas para asegurar la uniformidad, ajustando su tamaño sin alterar las proporciones originales, mediante el relleno de espacios para mantener la integridad visual.

Utilizando la red neuronal convolucional VGG19, seleccionada por su eficacia sobre algoritmos similares, el modelo fue entrenado para reconocer y comparar patrones gráficos eficazmente. Este entrenamiento se llevó a cabo tanto de forma independiente como mediante el uso de la librería DeepImageSearch, que facilita la integración de algoritmos de aprendizaje profundo en aplicaciones de búsqueda de imágenes.

Para evaluar la efectividad del sistema, se realizaron pruebas con repositorios de menor tamaño, donde las imágenes fueron clasificadas manualmente para luego comparar estos resultados con las predicciones del modelo.

Se desarrolló también un sistema web en el lenguaje Java, con login institucional para usuarios autorizados. Esta plataforma permite seleccionar una imagen cualquiera del computador, la cual se la normaliza para su análisis mediante el modelo entrenado en Python, y posteriormente presentará logos similares encontrados. Al posicionarse sobre cualquier logo resultante, el sistema muestra detalles completos del signo distintivo asociado, como la denominación, el trámite, fecha de presentación entre otras cosas. Esta funcionalidad proporciona a los analistas del SENADI todas las herramientas necesarias para decidir sobre el registro de nuevos logos, mejorando así la gestión de los derechos intelectuales y la protección contra posibles infracciones.

Palabras clave: Recuperación de imágenes, Signos Distintivos, Machine Learning, Propiedad Intelectual, Red Neuronal Convolucional (CNN), modelo VGG19.

Theme: "Image Retrieval System for Distinctive Signs, Based on Machine Learning."

Author: Ing. Michael Jamil Yanangómez Suárez

ABSTRACT

Artificial intelligence and machine learning are revolutionizing the way we process and analyze data in diverse areas, ranging from medicine to engineering and, more recently, in the field of intellectual property. Machine learning allows systems to learn and improve from experience, enabling them to manage large volumes of data with accuracy and speed that surpass human capabilities.

Intellectual property in Ecuador faces unique challenges, particularly in the management and protection of intellectual rights associated with distinctive signs such as logos, tasks that are under the responsibility of SENADI. Traditionally, logo review to ensure originality and prevent infringements has been a manual task, carried out by experts who can easily be overwhelmed by the growing amount of visual data. This methodology is not only prone to errors but also unsustainable in the long term given the high frequency of registration requests.

This project aims to address this issue by creating a system designed to significantly improve accuracy and efficiency in the identification of similar logos. The process began with the creation of an extensive image repository, which were normalized to ensure uniformity, adjusting their size without altering the original proportions, by filling in spaces to maintain visual integrity.

Using the VGG19 convolutional neural network, selected for its effectiveness over similar algorithms, the model was trained to recognize and compare graphic patterns efficiently. This training was carried out both independently and through the use of the DeepImageSearch library, which facilitates the integration of deep learning algorithms in image search applications.

To evaluate the effectiveness of the system, tests were conducted with smaller repositories, where images were manually classified and then compared with the model's predictions.

A web system was also developed in the Java language, with institutional login for authorized users. This platform allows the user to select any image from their computer, which is normalized for analysis using the model trained in Python, and will then present similar logos found. When hovering over any resulting logo, the system displays complete details of the associated distinctive sign, such as the name, process, filing date, among other things. This functionality provides SENADI analysts with all the necessary tools to decide on the registration of new logos, thus improving the management of intellectual rights and protection against potential infringements.

Keywords: Image retrieval, Distinctive Signs, Machine Learning, Intellectual Property, Convolutional Neural Network (CNN), VGG19 model.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE ANEXOS

CAPÍTULO I

1. DESCRIPCIÓN EL PROBLEMA

La falta de una herramienta efectiva y automatizada para identificar logos similares en el repositorio y registro de signos distintivos en el SENADI es un problema importante. Actualmente, la detección de posibles infracciones de derechos intelectuales depende en gran medida de procesos manuales, lo que no solo es propenso a errores, sino que también consume recursos significativos y retrasa la respuesta a posibles infracciones.

1.1. Introducción

En la era digital actual, la capacidad de acceder y procesar información de manera eficiente se ha convertido en un aspecto fundamental en diversos campos, desde la seguridad y la publicidad hasta la investigación forense. Uno de los desafíos más relevantes en este contexto es la recuperación de imágenes, especialmente cuando se trata de signos distintivos, como logotipos, marcas y símbolos. Estos elementos visuales son esenciales para la identificación de productos y servicios, y su reconocimiento preciso puede tener un impacto significativo en las decisiones comerciales y legales. (BizMetriks, 2013)

El presente trabajo se centra en el desarrollo de un Sistema de Recuperación de Imágenes de Signos Distintivos, utilizando técnicas de Machine Learning. Este enfoque se basa en la capacidad de los algoritmos de aprendizaje automático para reconocer patrones y aprender de grandes volúmenes de datos, lo que permite mejorar la precisión y la eficacia del proceso de recuperación de imágenes. A través de la implementación de modelos avanzados, como redes neuronales convolucionales (CNN), se busca optimizar la identificación y clasificación de signos distintivos en bases de datos extensas. (Celina, 2006)

Este sistema no solo representa un avance tecnológico en la búsqueda visual, sino que también aborda importantes cuestiones relacionadas con la propiedad intelectual y la protección de marcas. A medida que la digitalización y el comercio en línea continúan en expansión, la necesidad de herramientas robustas para la identificación de signos distintivos se vuelve cada vez más crítica. En este sentido, el presente trabajo contribuye a la creación de un entorno más seguro y confiable para las empresas y los consumidores, permitiendo proteger de mejor manera los derechos intelectuales. (Chace, 2015)

1.2. Justificación

Un sistema de recuperación de imágenes de signos distintivos basado en Machine Learning es fundamental en un mundo donde la información visual se encuentra en constante crecimiento. La capacidad de identificar y clasificar imágenes de signos distintivos es vital para diversas aplicaciones, incluyendo la protección de derechos de propiedad intelectual, marcas registradas y la prevención de fraudes. Utilizar técnicas de Machine Learning para este propósito no solo permite una atención más precisa y rápida en el reconocimiento de imágenes, sino que también ofrece un enfoque escalable que puede adaptarse a diferentes tipos de señales y características visuales. (Apd, 2019)

Este tipo de sistema puede revolucionar la forma en que las empresas y las organizaciones gestionan sus activos visuales, garantizando una mayor eficiencia y seguridad en el tratamiento de la información. (B, 2015)

La justificación de la propuesta radica en la importancia de solucionar un problema específico que enfrenta el Servicio Nacional de Derechos Intelectuales (SENADI) en Ecuador, que es la identificación efectiva de logos de signos distintivos similares para salvaguardar la originalidad y proteger los derechos intelectuales asociados. En el presente trabajo de titulación se propone el desarrollo de un sistema.

La propuesta para el desarrollo de un sistema de identificación de logos similares para el Servicio Nacional de Derechos Intelectuales (SENADI) en Ecuador se justifica por la imperiosa necesidad de solucionar un problema crucial: la detección efectiva de logos que puedan violar la originalidad y los derechos intelectuales asociados. La digitalización y la reproducción constante de logos en diversos contextos han incrementado el riesgo de infracciones, por lo que la identificación temprana y precisa de logos similares es fundamental para prevenirlas y garantizar la protección legal de la propiedad intelectual.

Esta propuesta destaca por su innovación al implementar tecnologías de machine learning para la comparación de imágenes de signos distintivos, proporcionando una solución avanzada y pionera en el ámbito de la propiedad intelectual. La integración de esta tecnología aporta un valor teórico significativo al campo, explorando nuevas fronteras en la aplicación de tecnologías complejas para resolver problemas específicos.

La utilidad metodológica del sistema radica en la eficiencia que aporta a los servicios del SENADI. Al automatizar la recuperación de imágenes, se simplifican los procesos internos, se aumenta la eficacia en la detección de posibles infracciones y se permite una respuesta más rápida y efectiva. La aplicabilidad directa de la propuesta a la realidad del SENADI se evidencia en su capacidad de abordar los desafíos específicos que enfrenta la institución en la gestión y protección de signos distintivos.

Los beneficios esperados son múltiples. La implementación exitosa del sistema proporcionará al SENADI una herramienta automatizada para identificar logos similares, reduciendo el riesgo de infracciones y fortaleciendo la defensa de los derechos intelectuales. Esto contribuirá a la modernización de las prácticas de la institución, posicionándola a la vanguardia en la protección de la propiedad intelectual en un entorno digital en constante evolución. Además, la propuesta fomenta el progreso del conocimiento y las prácticas de propiedad intelectual al explorar la aplicación de tecnologías emergentes en este ámbito, resolviendo un problema específico y contribuyendo al avance del conocimiento.

En resumen, la propuesta no solo se centra en solucionar un problema real del SENADI, sino que también ofrece una solución innovadora, con un alto valor teórico y práctico, que aporta a la modernización y eficiencia de la institución, al mismo tiempo que impulsa el progreso del conocimiento en el campo de la propiedad intelectual.

Los siguientes elementos resaltan la relevancia y originalidad de la propuesta:

a. Necesidad Urgente

- La digitalización y reproducción de logos en una variedad de contextos ha aumentado el peligro de violaciones de derechos intelectuales
- La identificación temprana y precisa de logos similares es crucial para prevenir posibles violaciones y garantizar la protección legal de los derechos intelectuales

b. Innovación en la implementación de tecnologías de machine learning

- La propuesta se destaca por la aplicación específica de tecnologías de aprendizaje automático para la comparación de imágenes de signos distintivos, lo que proporcionaría una solución de propiedad intelectual innovadora y avanzada

c. Valor teórico en la Integración de tecnologías avanzadas

- La integración de machine learning aporta un valor teórico significativo al campo de la protección de derechos intelectuales, explorando nuevas fronteras en la aplicación de estas tecnologías para resolver problemas específicos y complejos.

d. Utilidad Metodológica en la eficiencia de los servicios del SENADI

- La implementación de un sistema de recuperación de imágenes no solo simplificará los procedimientos internos del SENADI, sino que también aumentará la eficacia en la detección de infracciones potenciales, lo que permitirá una respuesta más rápida y efectiva.

e. Aplicabilidad directa a la realidad del SENADI

- La propuesta responde directamente a las necesidades del SENADI al utilizar tecnologías avanzadas para abordar los desafíos particulares que enfrenta en la gestión y protección de signos distintivos.

f. Beneficios esperados

- La implementación exitosa de este sistema brindará al SENADI, una herramienta útil y automatizada para identificar logos similares, reduciendo el riesgo de infracciones y fortaleciendo la defensa de los derechos intelectuales.
- Contribuirá a la modernización de las prácticas de la Institución, posicionándola a la vanguardia en la protección de los derechos intelectuales en un entorno digital en constante cambio.

g. Contribución al Progreso del conocimiento y prácticas de propiedad intelectual

- Al explorar la aplicación de tecnologías emergentes en el ámbito de la propiedad intelectual, la propuesta no solo resuelve un problema específico, sino que también contribuye al avance del conocimiento.

La propuesta como tal, no solo aborda los problemas reales del SENADI, sino que también ofrece un enfoque innovador para la protección de los derechos intelectuales, lo que beneficia significativamente a la institución y población en general, estableciendo un nuevo estándar en la aplicación de tecnologías avanzadas en el campo de la propiedad intelectual.

1.3. Planteamiento el problema

La creciente cantidad de signos distintivos, como marcas y logotipos, plantea un desafío significativo en su gestión y protección. Las empresas y ciudadanos enfrentan dificultades para identificar infracciones de propiedad intelectual y para monitorear el uso de sus signos distintivos en el mercado. Además, la clasificación y recuperación manual de estas imágenes es un proceso laborioso que puede llevar a errores humanos. La falta de suficientes herramientas automatizadas y precisas para la recuperación de imágenes y la identificación de similitudes entre diferentes signos distintivos hace que sea fundamental desarrollar un sistema que integre Machine Learning para mejorar la eficiencia y la efectividad en la gestión de estos recursos visuales. (Correa et al., 2017)

1.4. Contextualización del tema u objeto

En el contexto actual, la digitalización y el acceso a grandes volúmenes de datos han transformado la manera en que las empresas y organizaciones operan. Los signos distintivos, que incluyen logotipos, marcas y otros elementos visuales, se han vuelto omnipresentes en el comercio y la publicidad. A medida que más empresas entran al mercado global, la probabilidad de confusión entre signos se incrementa. Existen diversas técnicas de recuperación de imágenes, pero muchas no utilizan inteligencia artificial, lo que limita su eficacia. Así, el desarrollo de un sistema basado en Machine Learning para la recuperación de imágenes de signos distintivos se presenta como una solución innovadora que podría cambiar las reglas del juego en este ámbito.

1.5. Objetivos

1.5.1. Objetivo general

Desarrollar un sistema de recuperación de imágenes de signos distintivos que utilice tecnologías de machine learning, que permita garantizar la originalidad de los logos de signos distintivos en el SENADI

1.5.2. Objetivo específico

1. Construcción de un repositorio de logos
2. Identificación y selección de tecnologías de machine learning
3. Desarrollo de algoritmos para la comparación de imágenes
4. Desarrollo de un sistema amigable, seguro y confiable.
5. Evaluación de la efectividad del sistema.

CAPÍTULO II

2. MARCO TEÓRICO

2.1. Antecedentes

El desarrollo de sistemas automáticos de identificación de imágenes en dispositivos móviles es un problema que está abordándose de forma global, Por ejemplo, la empresa NEC con su servicio Gaziru (Fukuzawa, 2014). Más aún, dispositivos como Google Glass forman alianzas con empresas como Catchoom (Catchoom, 2015) para implementar reconocimiento de imágenes en aplicaciones de compras. Asimismo, la empresa Microsoft libera su dispositivo Hololens cuya tecnología involucra el reconocimiento de imágenes y realidad aumentada (Microsoft, 2015).

Estos hechos establecen tendencias en el desarrollo de aplicaciones para dispositivos portátiles que se enfoquen en el reconocimiento de imágenes y que hacen uso de algoritmos de aprendizaje computacional. Esto se soporta al observar los resultados actuales tanto en investigación de visión por computadora como en herramientas de código abierto desarrolladas para este fin.

El sistema de reconocimiento automático de logotipos comerciales en imágenes pretende seguir estas tendencias y enfocar su aplicación en centros comerciales. Con base en la idea de (Ferrell & Hartline, 2012) que establece que la publicidad se basa en la idea o noción de que, mientras más llegada al público tenga un producto, más conocido se hará y por lo tanto, tendrá más posibilidades de ser consumido.

2.2. Inteligencia artificial

Para entender la inteligencia artificial (IA), primero debemos comprender el concepto de inteligencia. La palabra "inteligencia" proviene del latín "intelligere", que significa "elegir entre diferentes opciones". Diversos autores han definido la inteligencia como la capacidad de abordar tareas de forma creativa, alcanzar objetivos en diferentes entornos o lograr metas complejas que requieren comprensión, autocrítica y aprendizaje.

Existen múltiples perspectivas sobre la inteligencia, incluyendo la propuesta de Howard Gartner, quien identifica nueve tipos de inteligencia, desde la lingüística hasta la naturalista. Sin embargo, no existe un consenso universal sobre qué define la inteligencia. (Qigang, 2012) Por otro lado, en la obra Surviving IA (Chace, 2015), se presenta la inteligencia a modo que la capacidad para alcanzar objetivos en diversos entornos. Algo similar se expresa en Vida 3.0, donde se define como la aptitud para cumplir metas complejas, que incluyen la agudeza, la autocrítica, la resolución de problemas y el autoaprendizaje (Tegmark, 2017).

Existen múltiples perspectivas sobre la inteligencia, incluyendo la propuesta de Howard Gartner, quien identifica nueve tipos de inteligencia, desde la lingüística hasta la naturalista. Sin embargo, no existe un consenso universal sobre qué define la inteligencia.(Gartner, 2018).

El concepto de IA surge en el siglo XX con John McCarthy, quien la define como la creación de máquinas inteligentes a través de programas informáticos avanzados. Marvin Minsky la describe como la capacidad de las máquinas para realizar tareas que requieren inteligencia humana. (McCarthy, 2007). Marvin Minsky, también de EE. UU., la define como la disciplina que permite que las máquinas ejecuten tareas que requerirían inteligencia si las realizaran humanos (Guinovart, 1998).

Según (Romero, 2007) Existen dos enfoques principales para definir la IA: uno ingenieril, que se centra en la creación de sistemas informáticos capaces de realizar tareas inteligentes, y otro científico, que busca comprender el comportamiento inteligente y desarrollar una teoría que explique la inteligencia tanto en seres vivos como en máquinas.

A pesar de la falta de consenso sobre qué es la IA, existe un acuerdo sobre su potencial transformador. Andrew Ng la describe como la "nueva electricidad", capaz de revolucionar industrias enteras. (Schwab, 2016).

La definición de IA continúa en constante evolución, debido a que su desarrollo es un proceso en curso. La ambigüedad en su definición depende de los objetivos de los investigadores y los métodos utilizados para crear modelos de IA. (Ng, 2019).

En última instancia, la pregunta sobre qué es la IA puede tener diversas respuestas. Estas dependen significativamente los métodos utilizados para desarrollar modelos de IA. La ambigüedad en su definición surge del hecho de que el desarrollo de la IA todavía está en curso (Schwab, 2016).

A continuación, revisaremos los hitos más importantes en la evolución de la IA.

2.2.1. Historia

En principio, el ser humano buscaba formas de reducir el esfuerzo en su trabajo. Hace aproximadamente dos millones de años, con la creación tales como herramientas, comenzó este proceso. Posteriormente, hace varios milenios, aprendió a dominar el fuego. Se descubrió la agricultura y la ganadería, utilizando esta última para facilitar el cultivo de la tierra. Un cuarto avance significativo se produjo hace miles de años con la escritura, lo que permitió al ser humano almacenar información en papel, pergaminos y luego en libros (Alfonseca, 2019). Más recientemente, hace doscientos años, inició la Revolución Industrial, donde entre los siglos XVIII y XX, se sustituyeron los animales de cultivo por maquinaria que utilizaba, surgieron las computadoras, que vinieron a complementar y reemplazar algunas funciones mentales humanas, delegando en ellas cálculos complicados y tareas repetitivas. Sin embargo, su limitación es que son rígidas; no pueden adaptarse a situaciones no anticipadas por los humanos (Alfonseca, 2019).

2.2.2.Principios Éticos

En su artículo "Las éticas de la inteligencia artificial", Bostrom y Yudkowsky (2011) identifican cinco características esenciales que deben poseer los sistemas de inteligencia artificial:

- **Responsabilidad.** En caso de fallo de un sistema, es importante determinar quién es el responsable.
- **Transparencia.** Es fundamental desarrollar algoritmos de IA que sean claros y accesibles para su revisión.
- **Auditabilidad.** La complejidad de algoritmos basados en Redes Neuronales hace difícil entender las decisiones que toman. En cambio, los algoritmos más simples, como los árboles de decisión, son más fáciles de analizar por los desarrolladores.
- **Incorruptibilidad.** Los sistemas deben ser resistentes a manipulaciones externas.
- **Predictibilidad.** Los programas tienen que ser diseñados con una comprensión clara de los comportamientos esperados.

Problemas de investigación

- **Metas de investigación.** Los objetivos deben ser específicos y orientados al bienestar social.
- **Alteración en investigación.** La introducción de proyectos de IA debe centrarse en asegurar su uso provechoso.
- **Conexión entre ciencia y política.** Para el avance de la IA, es crucial un diálogo fluido y constructivo entre la comunidad científica y los responsables políticos.
- **Cultura de investigación.** La investigación en IA debe cultivar una cultura de colaboración, confianza y transparencia entre todos los involucrados.
- **Evitar la carrera.** La competencia excesiva en el desarrollo de IA debe evitarse para que se prioricen cuestiones esenciales como la seguridad.

2.2.3. Ética y Valores

- **Seguridad.** Los sistemas de IA deben ser seguros y verificables a lo largo de su vida útil.
- **Transparencia en los fallos.** Es crucial comprender las razones detrás de cualquier daño causado por un sistema de IA.

- **Transparencia judicial.** Las decisiones tomadas por sistemas autónomos deben poder explicarse de forma auditada por autoridades humanas.
- **Responsabilidad.** Las empresas responsables de los sistemas de IA deben asumir la responsabilidad moral de su uso y actuar para alinear sus soluciones con dicha responsabilidad.
- **Alineación de valores.** Los sistemas de IA deben ser diseñados para que sus objetivos y comportamientos se alineen con los valores humanos.
- **Valores humanos.** Estos sistemas deben operar de modo que sean compatibles con la dignidad, derechos y diversidad cultural de las personas.
- **Privacidad individual.** Es esencial que los usuarios consientan el acceso y manejo de sus datos por parte de sistemas de IA.
- **Beneficio compartido.** Las tecnologías de IA deben beneficiar a la mayor cantidad posible de personas.
- **Prosperidad compartida.** La riqueza generada por la IA debe distribuirse equitativamente para el bien de toda la humanidad.
- **Control humano.** Los individuos deben decidir cómo y cuándo delegar decisiones a sistemas de IA para lograr sus objetivos.
- **No subversión.** El poder delegado a sistemas avanzados de IA debe respetar y fortalecer los procesos cívicos y sociales, en lugar de socavarlos.

2.2.4. Definición de Machine Learning

En pocas palabras, el aprendizaje automático (ML) es una rama de la inteligencia artificial que dota a las máquinas de la capacidad de aprender a partir de datos, sin necesidad de ser programadas explícitamente. Esto se logra mediante un proceso de entrenamiento en el que las máquinas "observan" grandes conjuntos de datos (datasets) y extraen patrones o relaciones que les permiten realizar tareas o tomar decisiones. A diferencia de la programación tradicional, donde las instrucciones son fijas y se ejecutan paso a paso, el ML permite a las máquinas adaptarse y mejorar su rendimiento con la experiencia. (The Royal Society, 2017).

Los algoritmos de ML son capaces de prever nuevos casos basándose en la experiencia adquirida a partir de los datos con los que fueron entrenados (Gisela & García, 2014).

2.3. Tipos de Aprendizaje

Broussard (2018) clasifica el aprendizaje:

- **Supervisado**: En este tipo, el sistema aprende a partir de ejemplos etiquetados, donde se proporciona una salida esperada para cada entrada. El objetivo es descubrir una regla general que relacione las entradas con las salidas:
  - **Clasificación**: Se utilizan para categorizar las entradas en grupos distintos, como "gato" o "perro", "verdadero" o "falso".
  - **Regresión**: Se utiliza para predecir valores continuos, como la temperatura o el precio de un producto.
- **No supervisado**: A diferencia del aprendizaje supervisado, aquí el sistema no recibe etiquetas para las entradas. El objetivo es encontrar patrones y estructuras en los datos sin ninguna guía previa.
- **Reforzamiento**: El sistema podría agrupar automáticamente los clientes de una tienda online en categorías según sus hábitos de compra, sin tener que definir previamente estas categorías.

El aprendizaje supervisado es actualmente el método más maduro, mostrando mayor efectividad en una variedad de aplicaciones. El aprendizaje no supervisado, aunque prometedor, aún se encuentra en desarrollo y su éxito depende del problema específico, del conocimiento previo y de los objetivos. (Benitez Ruíz, 2015)

Existen dos enfoques principales para entrenar algoritmos: entrenamiento offline y entrenamiento online. El entrenamiento offline implica entrenar y probar el algoritmo fuera del entorno real de producción. Una vez completado este proceso, el modelo se "congela" y se implementa en producción. Cualquier cambio o modificación posterior requiere un nuevo entrenamiento offline, prueba y reimplementación. Este método es el más utilizado en el aprendizaje automático porque permite evaluar el comportamiento del modelo antes de que sea utilizado por los usuarios finales. (The Royal Society, 2017).

En contraste, el entrenamiento online comienza con una fase inicial de entrenamiento y prueba offline, pero una vez que el algoritmo se implementa en producción, sigue aprendiendo y ajustándose continuamente a partir de los datos reales que recopila. Esto significa que el modelo se adapta constantemente a nuevas circunstancias y patrones emergentes, lo que puede ser particularmente beneficioso en entornos dinámicos donde los datos cambian constantemente. (The Royal Society, 2017).

2.3.1. Entrenamiento offline

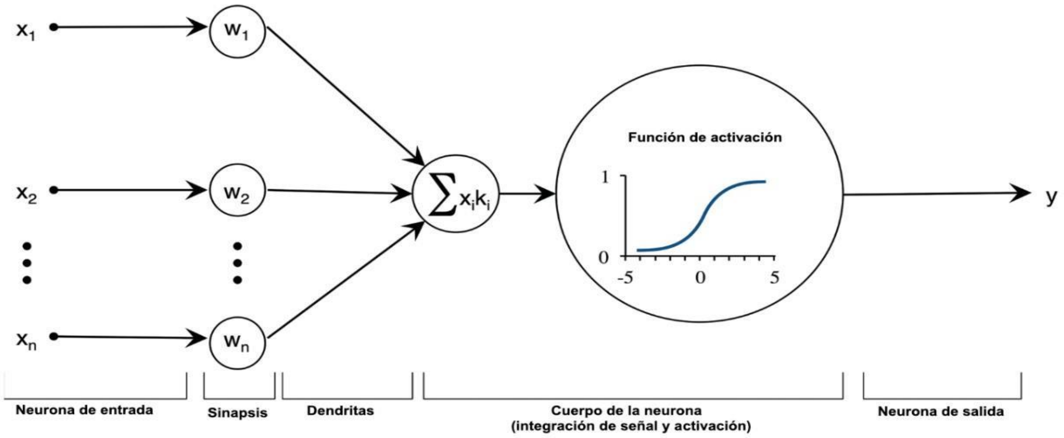
Para entrenar un modelo de aprendizaje automático, es crucial dividir el conjunto de datos en tres subconjuntos: entrenamiento, validación y prueba, cada uno con un rol específico.

- **Conjunto de entrenamiento**: Se utiliza para ejercitar y mejorar los parámetros de las Redes Neuronales (descritas en el punto 2.2.2). Es el corazón del proceso de aprendizaje. Es usado para ajustar los parámetros de las Redes Neuronales, un tipo de modelo de aprendizaje automático que emula la estructura neuronal del cerebro. Este ajuste se realiza mediante un proceso iterativo: el modelo procesa el conjunto de entrenamiento repetidamente, modificando los "pesos" de las neuronas para minimizar el error de predicción. Cada pasada completa sobre el conjunto de entrenamiento se llama "época". Para mejorar la precisión, el conjunto de entrenamiento se suele dividir en "batches" más pequeños, cada uno de los cuales se procesa individualmente en una "iteración". (Sharma, 2017).
- **Conjunto de validación**: Se usa para evaluar el modelo durante el entrenamiento. Después de cada época, se procesa este conjunto para ajustar los parámetros del modelo, buscando un equilibrio entre precisión y generalización. Es importante destacar que el modelo no aprende en esta etapa, solo lo hace en el conjunto de entrenamiento. (Shah, 2017).
- **Conjunto de prueba**: Se utiliza una vez que el modelo ha sido completamente entrenado y configurado. Este conjunto sirve para evaluar la precisión del modelo final y generar un informe detallado de su rendimiento en datos nunca antes vistos. (Chartrand et al., 2017).

2.4. Redes Neuronales Artificiales

Son modelos de aprendizaje automático que utilizan una serie de unidades de procesamiento llamadas neuronas, interconectadas entre sí en capas. Esta estructura simple emula la forma en que funciona un cerebro. La unidad básica de una RN es el "Perceptrón", el cual toma como entrada un conjunto de valores que representan características. Cada valor es multiplicado por un "peso" específico, y los resultados son sumados y procesados por una función de activación, que genera un valor de salida. La combinación de muchos Perceptrones crea una red neuronal completa. (Mazurowski, Buda, Saha, & Bashir, 2018).

Figura 1. *Perceptron.*



Fuente: *Estructura de un Perceptron. - Fuente: Elaboración propia a partir de (Chartrand, 2017).*

Las Redes Neuronales (RN) llamadas Perceptrones Multicapa se construyen a partir de la unión de neuronas, organizadas en capas. La salida de una capa N se convierte en la entrada de la siguiente capa (N+1). La primera capa se denomina "capa de entrada" y recibe los datos de entrada, mientras que la última capa se conoce como "capa de salida" y representa el resultado. Las capas intermedias se llaman "capas ocultas" porque no generan salidas visibles (Chartrand et al., 2017).

Para obtener la predicción de una observación, como una imagen, se activa cada nodo de cada capa, desde la capa de entrada hasta la capa de salida. Este proceso se llama "propagación hacia adelante" (forward propagation). El resultado de la capa de salida se procesa mediante una Función de Costo o Error, generalmente la función softmax, que lo transforma en un número que representa una probabilidad.

El aprendizaje de la red implica ajustar los valores de peso y características de los nodos. El algoritmo utilizado para esta modificación se denomina "descenso de gradiente" (gradient descent), que busca la combinación de parámetros que minimice el resultado de la Función de Costo para el conjunto de datos utilizado. En cada predicción, la red se evalúa según su salida, determinando el error. Los parámetros se ajustan para reducir este error, un proceso conocido como "propagación hacia atrás" (back propagation) (Chartrand et al., 2017).

Una limitación importante de los Perceptrones Multicapa es su bajo rendimiento en el procesamiento de imágenes, que requiere una gran complejidad computacional. Por esta razón, se suele recurrir al aprendizaje profundo (Deep Learning) para este tipo de tareas (O'Shea & Nash, 2015). El desafío principal en el uso de RN es asegurar que funcionen correctamente no solo con los datos de entrenamiento, sino también con datos nuevos. La capacidad de funcionar con datos desconocidos se denomina generalización.

Al entrenar un modelo de aprendizaje automático (ML), se utiliza un conjunto de datos de entrenamiento. Los errores de medición con este conjunto se denominan error de entrenamiento. Por otro lado, los errores de medición con nuevos datos se conocen como errores de generalización o de prueba (Goodfellow, 2015).

La eficacia de una RN está determinada por los conceptos de subadaptación (underfitting) y sobreadaptación (overfitting). El subadaptado ocurre cuando el modelo no puede obtener un error suficientemente pequeño con los datos de entrenamiento. La sobreadaptación surge cuando la diferencia entre el error de entrenamiento y el error de prueba es demasiado grande (Goodfellow, 2015).

Si un modelo presenta sobreadaptación, se debe considerar reducir su capacidad, por ejemplo, disminuyendo el número de parámetros, o agregar más datos de entrenamiento (Chartrand et al., 2017). La reducción de la complejidad de las RN, con menor cantidad de parámetros, disminuye la posibilidad de sobreadaptación y mejora los resultados de predicción del modelo (O'Shea & Nash, 2015).

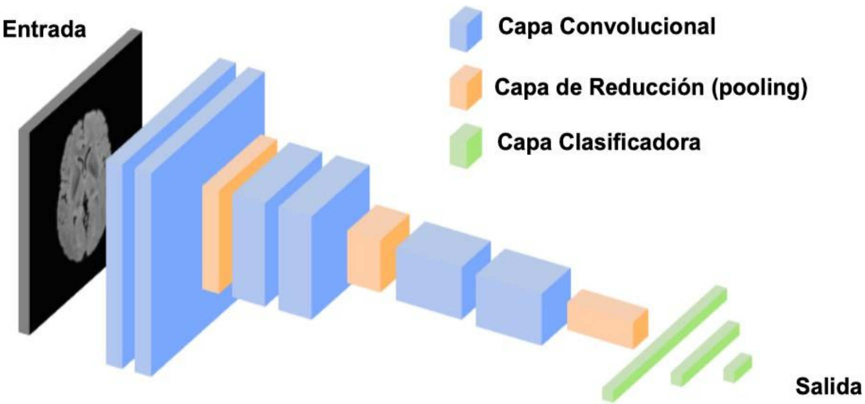
2.4.1. Deep Learning

El Deep Learning (DL) se caracteriza por emplear algoritmos que aprenden a través de una composición de "features" (Chartrand et al., 2017). Estas características reflejan la estructura jerárquica de los datos, descomponiendo representaciones complejas en términos de representaciones simples. Esta arquitectura las convierte en herramientas particularmente efectivas para la clasificación de imágenes (Chartrand et al., 2017).

Si bien existen diversas arquitecturas de aprendizaje profundo, las Redes Neuronales Convolucionales (RNC) son las más ampliamente utilizadas en este campo (Lopez Briega, 2016). Estas redes, aunque comparten las propiedades generales de las redes neuronales, poseen características distintivas que las hacen idóneas para el análisis de imágenes (Lopez Briega, 2016).

Las RNC poseen una estructura tripartita, compuesta por tres tipos de capas: convolucional, de reducción (también llamada pooling) y clasificadora. A continuación, analizaremos en detalle cada una de estas capas.

Figura 2. *Representación de una RNC.*

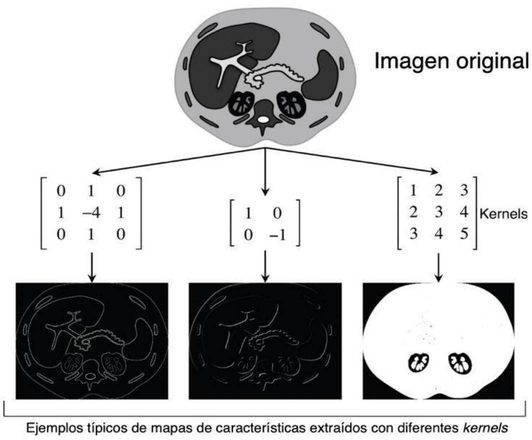


Fuente: Representación de de una RNC. En la misma se aplica una serie de capas convolucionales y de pooling para luego pasar por la capa clasificadora. - Fuente: Elaboración propia a partir de (Mazurowksi, 2018).

2.4.2. Capa Convolucional

La arquitectura de las redes neuronales convolucionales se caracteriza por la capa convolucional, la cual es fundamental en su funcionamiento. Esta capa recibe la imagen a clasificar como entrada y aplica un filtro, también conocido como kernel, sobre ella. El resultado de esta operación es un mapa de características de la imagen original, lo que permite reducir la cantidad de parámetros necesarios para el procesamiento (López Briega, 2016). En otras palabras, la capa convolucional simplifica la información de la imagen de entrada, eliminando la complejidad innecesaria y extrayendo las características más relevantes para la clasificación.

Figura 3. *Aplicación de Kernels sobre una imagen*



Fuente: Ejemplo de aplicación de un kernel sobre una imagen. A partir de los mismos se logra extraer distintas características como, por ejemplo, contornos. - Fuente: Elaboración propia a partir de (Chartrand, 2017).

En La convolución y la correlación, en el contexto del procesamiento de señales, son conceptos estrechamente relacionados. Según Benitez Ruíz (2015), la correlación puede interpretarse como la búsqueda de la presencia de una forma de onda específica dentro de una señal más amplia, ya sea en una dimensión o en dos.

En términos prácticos, la convolución se traduce en deslizar un pequeño patrón, denominado "kernel", sobre una imagen. Este kernel representa un "feature" o característica particular que se busca, como por ejemplo, bordes o cambios bruscos en la intensidad de la luz. Durante este deslizamiento, se calcula la similitud entre el kernel y la región de la imagen sobre la que se encuentra en cada posición.

Este proceso de comparación se realiza a través de un algoritmo de backpropagation, que ajusta los valores del kernel para minimizar la función de costo. Esto significa que el kernel se adapta con el objetivo de detectar con mayor precisión el feature deseado en la imagen (C. Selmo, comunicación personal, 17 de junio de 2019).

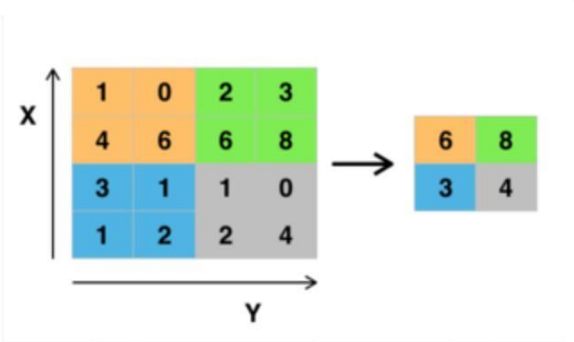
2.4.3. Capa de Reducción (pooling)

La capa de pooling, ubicada después de la capa convolucional, desempeña un papel crucial en la arquitectura de las redes neuronales convolucionales (CNNs). Su función principal es reducir las dimensiones espaciales (ancho y alto) del volumen de entrada, preparando así la información para la siguiente capa convolucional. Este proceso, conocido también como reducción de muestreo, implica una pérdida de información, lo cual podría parecer un inconveniente a primera vista. Sin embargo, esta reducción tiene dos beneficios importantes.

En primer lugar, la reducción del tamaño del volumen de entrada disminuye la carga computacional de las capas subsecuentes, lo que optimiza la eficiencia del proceso de aprendizaje. Por otro lado, la capa de pooling también ayuda a prevenir el sobreajuste de la red. La sobreadaptación ocurre cuando la red aprende a reconocer patrones específicos del conjunto de entrenamiento, pero no generaliza bien a nuevos datos.

Para realizar la reducción de dimensiones, la capa de pooling utiliza la operación de Max-pooling. Esta operación divide la imagen de entrada en un conjunto de rectángulos, y de cada subregión, selecciona el valor máximo. Como se ilustra en la Figura 4, este proceso reduce la cantidad de información, pero conserva las características más importantes de la imagen original. (Lopez Briega, 2016).

__Figura 4.__ Aplicación de la función Max-pooling.



Fuente: Ejemplo de cómo, de una subregión, la función max-pooling obtiene el valor máximo. - Fuente: (Briega, 2016).

Según (González, 2018) el motivo por el que se guarda el valor máximo de cada subregión es que el mismo hace referencia a una mayor presencia del feature que se estaba buscando con un kernel determinado"

2.4.4. Capa Clasificadora

Según (González, 2018) "luego de que se itere una N cantidad de veces sobre la Capa Convolucional y la Capa de Reducción, se utiliza una capa completamente conectada. Generalmente se trata de un Perceptrón Multicapa, en la que cada pixel es considerado como una neurona separada, tal cual las RN convencionales. El objetivo de esta capa es realizar la clasificación de las extracciones que se han realizado en las capas convolucionales y de reducción".

2.5. Visión Artificial

La visión artificial, como rama de la informática, se encarga de traducir imágenes del mundo físico a información digital. Este proceso, como describe Szeliski (2010), implica la reconstrucción de las imágenes tomando en cuenta sus propiedades intrínsecas como la forma, la iluminación y la distribución de colores.

La visión artificial se configura como un campo multidisciplinario que se nutre de la inteligencia artificial y el aprendizaje automático (Brownlee, 2019). Esta interconexión permite el desarrollo de sistemas complejos capaces de analizar y entender imágenes de manera similar a los humanos.

Dentro de la visión artificial, existen diversas tareas que se han convertido en pilares de este campo:

1. Clasificación: La clasificación de imágenes implica la asignación de una etiqueta a una imagen, identificando la clase dominante dentro de ella. Este proceso se aplica en situaciones donde la imagen presenta un solo objeto de interés en una forma claramente visible. Un ejemplo clásico es la distinción entre imágenes de gatos y perros.

2. Detección: La detección, a diferencia de la clasificación, permite la identificación de múltiples objetos de diferentes clases dentro de una misma imagen. Además de determinar la clase probable de cada objeto, la detección utiliza un "cuadro delimitador" (bounding box) para marcar la ubicación precisa de cada objeto en la imagen. Un ejemplo sería la detección de un gato y un perro dentro de una misma imagen, con un cuadro delimitador alrededor de cada uno.

3. Segmentación: La segmentación es una extensión de la detección de objetos que permite identificar diferentes clases dentro de una imagen con un nivel de detalle mucho mayor. En lugar de utilizar un cuadro delimitador, la segmentación define la ubicación de cada objeto a nivel de píxel, marcando el contorno de la clase píxel por píxel (Deep Learning Analytics, 2018). Esto permite una mayor precisión en la identificación y delimitación de objetos complejos.

2.6. VGG19: Una Red Convolucional Preentrenada

Una estrategia común para abordar problemas en redes neuronales profundas es utilizar redes previamente preentrenadas con grandes bases de datos y adaptarlas al problema específico que queremos resolver.

Para este propósito, es fundamental que la red preentrenada haya sido entrenada para resolver un problema de carácter más general, del cual nuestro problema pueda considerarse un caso particular. Por ejemplo, en lugar de entrenar una red desde cero para clasificar diferentes tipos de animales, podemos aprovechar una red preentrenada como la VGG19, que ha sido entrenada en el extenso conjunto de datos de ImageNet (Simonyan and Zisserman, 2015). Las razones para usar VGG19 incluyen:

- Ofrece una arquitectura comprensible y bien documentada, lo que facilita su implementación y personalización.
- Ha logrado excelentes resultados en la competencia ImageNet (ILSVRC), alcanzando una precisión superior al 90%.
- Contiene 16 capas convolucionales y 3 capas densas, lo que la hace más profunda que su predecesora VGG16, lo que puede ofrecer una mejor captura de características más complejas.

- El modelo, junto con los pesos preentrenados, está disponible en Keras, facilitando su uso inmediato en aplicaciones específicas.

CAPÍTULO III

3. METODOLOGÍA

La metodología empleada en este trabajo se organiza en varias etapas clave, cada una de las cuales es fundamental para garantizar la eficacia y funcionalidad del sistema propuesto. Para abordar este enfoque, hemos optado por utilizar la metodología CRISP-DM (Cross-Industry Standard Process for Data Mining) la cual proporciona un marco estructurado y sistemático para el desarrollo de proyectos de minería de datos. A continuación, ofrecemos una breve introducción a esta metodología y a sus componentes esenciales.

1. **Comprensión del Negocio**: En esta etapa se busca entender los objetivos y requisitos del proyecto desde la perspectiva del negocio. Se definen claramente las metas y se establecen criterios para el éxito. Esto permite al equipo alinear el desarrollo del modelo con las necesidades y expectativas de los interesados.
2. **Comprensión de los Datos**: Una vez que se comprenden los objetivos del negocio, se realiza una exploración inicial de los datos disponibles. Esto incluye la recopilación de datos, la descripción y la inspección de su calidad. Se identifican las variables relevantes y se comienza a formular hipótesis sobre cómo los datos pueden ayudar a resolver el problema planteado.
3. **Preparación de los Datos**: Esta fase implica todas las tareas necesarias para preparar los datos para el modelado. Incluye la limpieza de los datos, la transformación, la selección de atributos y la creación de conjuntos de entrenamiento y prueba. Un conjunto de datos bien preparado es esencial para el éxito del modelo de minería de datos.
4. **Modelado**: Durante esta etapa, se seleccionan y aplican técnicas de modelado adecuadas según los objetivos del proyecto. Se experimenta con diferentes algoritmos y parámetros, y se evalúa la calidad de los modelos generados. Es común realizar ajustes iterativos para optimizar el rendimiento del modelo.
5. **Evaluación**: Tras el modelado, se lleva a cabo una evaluación crítica de los resultados obtenidos. Se comprueba si el modelo cumple con los objetivos establecidos en la primera fase. En caso necesario, se pueden realizar ajustes adicionales o incluso regresar a etapas anteriores para mejorar los resultados.
6. **Despliegue**: Finalmente, se ponen en práctica los resultados del modelo. Esto puede incluir la implementación de un sistema que utilice el modelo para hacer predicciones o la presentación de los resultados a los interesados. También es importante documentar el proceso y establecer un plan para el mantenimiento y la monitorización del modelo a lo largo del tiempo.

3.1. Desarrollo del Sistema

Paralelamente a la fase de entrenamiento del modelo, se desarrollará la interfaz del sistema, que debe ser:

- **Amigable:** Facilitar el acceso y la interacción del usuario final con el sistema.
- **Seguro:** Incorporar medidas de protección de datos y autenticación de usuarios.
- **Confiable:** Asegurar que el sistema funcione de manera estable y precisa.

3.2. Etapas de la Metodología CRISP-DM:

3.2.1. Comprensión del Negocio:

Objetivo: Entender las necesidades del SENADI, en cuanto a la protección de los derechos intelectuales referentes a logos de signos distintivos.

Análisis: Identificar las necesidades urgentes y requerimientos específicos, incluyendo la precisión requerida, la facilidad de uso y la integración con los sistemas existentes del SENADI.

Recopilación de Información: Realizar entrevistas con los stakeholders del SENADI para obtener una comprensión profunda de sus procesos, necesidades y expectativas.

3.2.2. Fase de Investigación y Revisión Bibliográfica

En esta fase se llevará a cabo una revisión exhaustiva de la literatura existente relacionada con tecnologías de Machine Learning, recuperación de imágenes, y manejo de signos distintivos. Se podrán usar bases de datos académicas como Google Scholar, IEEE Xplore, y otros repositorios relevantes. El objetivo es identificar:

- Las metodologías más efectivas para el reconocimiento y clasificación de imágenes.
- Algoritmos existentes que sean aplicables a la identificación de logos de signos distintivos.
- Casos de estudio relevantes y proyectos previos que sirvan de referencia para el presente proyecto.

3.2.3. Definición del Problema

Con la información recabada, se establecerá un marco delimitado del problema que busca resolver el sistema. Se definirá claramente:

- La especificidad del manejo y clasificación de signos distintivos.

- Las limitantes de los sistemas existentes y la necesidad de un enfoque basado en Machine Learning.

3.2.4. Recolección de Datos

Una base indispensable para el modelo de Machine Learning será la recopilación de un conjunto de datos robusto. Para ello:

1. Fuentes de Datos: Se identificarán fuentes confiables para la recolección de imágenes de logos de signos distintivos. Estas fuentes incluyen los logos publicados en la gaceta mensual que genera el SENADI, nuevos registros de signos distintivos y datasets existentes de recuperación de imágenes.

2. Preprocesamiento: Se realizará la limpieza de los datos, que incluye la normalización de tamaños de imagen, rellenado de logos verticales y horizontales, y la eliminación de imágenes duplicadas o irrelevantes para asegurar la calidad del conjunto de datos.

3.2.5. Comprensión de los Datos

Recopilación de Datos: Se recopilarán datos de diferentes fuentes, como la gaceta mensual del SENADI, registros de nuevos signos distintivos y datasets existentes de recuperación de imágenes.

Análisis Exploratorio: Se analizarán los datos recolectados para identificar patrones, variables relevantes y potenciales problemas con la calidad de los datos.

Preparación de los Datos: Se transformarán y normalizarán los datos para prepararlos para el modelado.

3.2.6. Preparación de los Datos

Selección de Características: Se intenta identificar los patrones más representativos de los logos, características visuales, colores, formas, información textual asociada a los signos distintivos.

Ingeniería de Características: Se crearán nuevas variables a partir de las existentes para mejorar la precisión del modelo.

Transformación de Datos: Se aplicarán transformaciones a los datos para que sean adecuados para el modelo de Machine Learning.

3.3. Modelado

Selección del Algoritmo: Se seleccionará el algoritmo de Machine Learning más adecuado para la tarea de recuperación de imágenes, considerando las características de los datos, el tipo de modelo deseado y la precisión requerida. Las opciones más apropiadas serían:

Redes Neuronales Convolucionales (CNNs): Para el reconocimiento de patrones visuales en las imágenes.

Transfer Learning: Para aprovechar modelos pre-entrenados y acelerar el entrenamiento.

Entrenamiento del Modelo: Se entrenará el modelo de Machine Learning utilizando los datos preparados.

Optimización del Modelo: Se ajustarán los parámetros del modelo y se optimizarán sus características para lograr un rendimiento óptimo (cuyos detalles me mencionaran en la construcción del modelo).

3.3.1. Selección de Técnicas de Machine Learning

En esta etapa, se explorarán diversas técnicas de Machine Learning que son adecuadas para la implementación del sistema. Las técnicas a considerar serán:

- **Redes Neuronales Convolucionales (CNN):** Por su alto rendimiento en el reconocimiento de patrones visuales.
- **Transfer Learning:** Para aplicar modelos preentrenados y acelerar el proceso de entrenamiento.
- **Técnicas de Aumento de Datos:** Para aumentar artificialmente el tamaño del conjunto de datos mediante métodos como rotaciones, recortes y escalados.

3.3.2. Desarrollo de Algoritmos

Una vez seleccionadas las técnicas, se procederá al desarrollo de algoritmos específicos que se encargarán de la comparación de imágenes:

- **Entrenamiento del Modelo:** Utilizando el conjunto de datos previamente preparado, se entrenará el modelo de Machine Learning. Esta fase incluirá el ajuste de hiperparámetros y la selección de la arquitectura más adecuada de red neuronal.
- **Validación Cruzada:** Se implementará una validación cruzada para asegurar que el modelo generalice bien a datos no vistos, evitando el sobreentrenamiento.

3.4. Evaluación

Evaluación del Modelo: Se evaluará el rendimiento del modelo utilizando diferentes métricas como precisión, recall y F1-score.

Comparación con otras Soluciones: Se comparará el rendimiento del modelo VGG19 con otras soluciones existentes como la librería DeepSeearchImage, para determinar su competitividad.

Análisis de Resultados: Se analizarán los resultados de la evaluación para identificar las áreas de mejora y tomar decisiones sobre la implementación del sistema.

3.4.1. Evaluación del Sistema

Una vez finalizado el desarrollo, se realizará una evaluación exhaustiva del sistema:

- **Pruebas de Funcionamiento:** Verificar que todas las funcionalidades del sistema operen correctamente.
- **Evaluación de Precisión:** Medir la precisión del sistema en la recuperación y clasificación de imágenes, utilizando métricas como precisión, recall, y F1-score.
- **Análisis de Resultados:** Comparar el rendimiento del sistema con resultados previos y estándares del sector.

3.5. Despliegue:

Diseño de la Interfaz: Se diseñará una interfaz amigable, segura y confiable para el sistema, permitiendo la interacción del usuario con el modelo de Machine Learning.

Integración con Sistemas Existentes: Se integrará el sistema con las tecnologías existentes del SENADI para facilitar su uso y garantizar la coherencia de los datos.

Capacitación de Usuarios: Se capacitará a los usuarios del SENADI sobre el funcionamiento del sistema y sus beneficios.

3.5.1. Implementación y Capacitación

Después de la validación, se procederá a implementar el sistema en el ambiente real del SENADI. Esto incluirá:

- Capacitación a los funcionarios sobre el uso del sistema.
- Provisión de manuales y documentos de apoyo para asegurar el entendimiento y buen uso del sistema.

3.6. Monitoreo y Mantenimiento

Seguimiento del Rendimiento: Se monitoreará el rendimiento del sistema para identificar posibles problemas y realizar ajustes necesarios.

Actualización del Modelo: Se actualizará el modelo de Machine Learning con nuevos datos para mejorar su precisión y adaptarlo a las nuevas necesidades del SENADI.

Mantenimiento y Soporte: Se proporcionará un servicio de mantenimiento y soporte técnico para garantizar el correcto funcionamiento del sistema.

La metodología CRISP-DM permite un enfoque estructurado y flexible para el desarrollo del sistema de recuperación de imágenes de signos distintivos, asegurando la calidad del modelo, su integración con las necesidades del SENADI y la optimización de su rendimiento.

3.6.1. Evaluación Continua y Mantenimiento

Finalmente, se establecerá un plan para la evaluación continua del sistema post-implementación, que incluirá:

- Recopilación de feedback de los usuarios.

Mantenimiento y actualización del sistema para incorporar nuevas funcionalidades y optimizaciones basadas en la retroalimentación recogida.

3.7. Construcción del modelo

3.7.1. Primera Sección - Descarga de Signos Distintivos

Figura 5. *Descarga de Signos Distintivos*

```
public void getSignos(String ruta) {
    ...
    Controlador c = new Controlador();
    List<PpdiSolicitudSignoDistintivo> signos = new ArrayList<>();
    while (n <= f) {
        List<PpdiSolicitudSignoDistintivo> aux = c.getSignosByGaceta(n);
        ...
        signos.addAll(aux);
        n++;
    }
    ...
    for (int i = 0; i < signos.size(); i++) {
        try {
            PpdiSolicitudSignoDistintivo signo = signos.get(i);
            ...
            String downloadedlogo = c.descargaLogo(ruta, signo.getNumeroExpediente());
            ...
        } catch (IOException ex) {
            ...
        }
    }
    ...
}
```

Fuente: *Yanangómez (2024)*

- **Propósito:** Este método se ocupa de descargar los logos de los signos distintivos que se encuentra en el repositorio web del sistema Alfresco, en un rango especificado de gacetas (660 a 737).
- **Estructura:**
    - Usa un ciclo while para iterar sobre cada gaceta.
    - Para cada gaceta, invoca un método en un objeto Controlador para obtener una lista de signos distintivos.
    - Descarga el logo asociado para cada signo distintivo y maneja excepciones en caso de fallos.
- **Consideraciones:**
    - Se registra el número de logos descargados, así como los errores en caso de haber ocurrido un problema en la descarga de algún logo.
    - Se realizó la descarga de los logos hasta la gaceta 737, porque esa era la última publicada por el SENADI, con fecha 31 de Julio de 2024.

3.7.2. Segunda Sección - Carga y Procesamiento de Imágenes

Figura 6. *Carga y Procesamiento de Imágenes*

```
public void loadImage() {
    ...
    List<String> errores = new ArrayList<>();
    for (int i = 0; i < list.length; i++) {
        ...
        if (!pathImage.contains(".pdf")) {
            ...
            BufferedImage image = ImageIO.read(file);
            ...
            BufferedImage resizedImage = createDefinedImage(image, width, height);
            ...
        }
    }
    ...
}
```

Fuente: *Yanangómez (2024)*

- **Propósito:** Cargar imágenes desde un repositorio de logos, realizar validaciones, y redimensionar las imágenes a dimensiones definidas (224x224).
- **Estructura:**
    - Recorre todos los archivos en un directorio y procesa solo aquellos que son imágenes.
    - Redimensiona las imágenes usando métodos no mostrados (increaseSize, createDefinedImage).

- **Consideraciones:**
    - Se mantiene un registro de errores para las imágenes que no se pueden procesar.
    - Lo ideal es manejar excepciones específicas en lugar de una general para mejor depuración.

3.7.3. Tercera Sección - Características de Imágenes

Figura 7. *Características de Imágenes*

```
def calcular_y_guardar_caracteristicas(dir_imagenes, nombre_archivo):
    ...
    np.save(nombre_archivo, np.array(features_list))
```

Fuente: *Yanangómez (2024)*

- **Propósito:** Calcular características de imágenes usando un modelo de red profunda (como VGG19) y guardar esos datos.
- **Estructura:**
    - Se obtiene una lista de imágenes y luego se extraen características usando el modelo preentrenado VGG19.
    - Las características de las imágenes son almacenadas en archivos '.npy', para luego ser utilizados al momento de obtener las imágenes similares.
- **Consideraciones:**
    - La estructura de la tecnología utilizada es relevante tareas de aprendizaje y clasificación de imágenes.
    - El uso de la librería numpy permite crear archivos que guardan las características más relevantes de los logos, y posteriormente utilizarlos sin tener que nuevamente hacer el proceso de entrenamiento.

3.7.4. Cuarta Sección - Búsqueda de Imágenes Similares

Figura 8. *Búsqueda de Imágenes Similares*

```
def buscar_con_vgg19(image_path, numer_of_images):
        ...
        return logos
```

Fuente: *Yanangómez (2024)*

- **Propósito:** Encontrar imágenes similares a una dada utilizando un método de machine learning, para este caso se utilizó el modelo preentrenado VGG19.
- **Estructura:**
    - Utiliza una mezcla de modelos de búsqueda que almacenan características previamente calculadas.
- **Consideraciones:**
    - Se debe gestionar eficientemente el acceso a la memoria o almacenamiento debido a la gran cantidad de datos que debe manejar.

3.7.5. Quinta Sección - Servidor HTTP

Figura 9. *Servidor HTTP*

```
class SimpleHTTPRequestHandler(BaseHTTPRequestHandler):
    ...
    def do_POST(self):
        ...
        self.send_response(200)
        ...
```

Fuente: *Yanangómez (2024)*

- **Propósito:** Este bloque configura un servidor HTTP simple para manejar solicitudes POST que contienen imágenes para buscar similitudes.
- **Estructura:**
    - Se utilizan solicitudes POST, que envían la imagen y el número deseado de logos más similares a obtener.
- **Consideraciones:**
    - La integración de tecnologías divide el monitoreo de ejecución del sistema, por lo que hay que estar atento en identificar posibles errores y saber en qué parte del flujo de esta están ocurriendo.

El código analiza y procesa imágenes para tratar tareas de búsqueda de imágenes similares a través de un flujo que incluye la recolección de datos, el análisis de imágenes, y la construcción de un sistema web basado en un servidor HTTP. Se benefician de frameworks de Java, así como de bibliotecas de Python para tareas de machine learning. Algunas mejoras que se podrían implementar incluyen optimización de manejo de excepciones, mejora de las funciones de logging, y la consideración de prácticas de seguridad en la configuración del servidor.

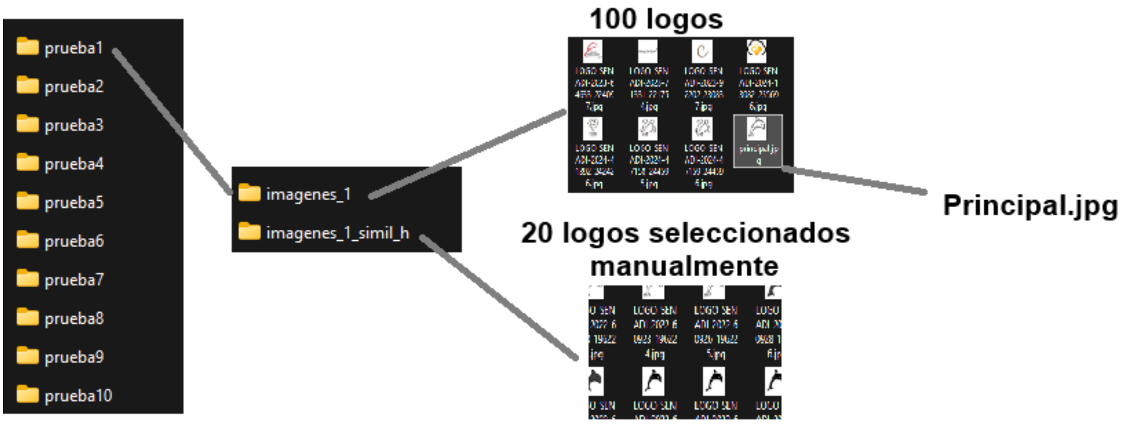
CAPÍTULO IV

4. Desarrollo de la Metodología CRISP-DM

La metodología presentada para la evaluación del modelo se apoya en un enfoque estructurado para medir la eficacia de un sistema de recuperación de imágenes. A continuación, se extiende y analiza cada componente de la metodología, así como los datos y sus implicaciones.

En el proceso de evaluación del modelo, se crearon 10 conjuntos de pruebas, denominados 'prueba_n' donde 'n' es el número de prueba. Cada uno de estos conjuntos se estructuró en carpetas que, a su vez, contenían dos subcarpetas: **imagenes_n** y **imagenes_n_simil_h**. La carpeta **imagenes_n** incluye 100 logos, entre los cuales se destaca un archivo llamado **principal.jpg**, representando el logo de referencia. En la subcarpeta **imagenes_n_simil_h**, se agrupan 20 logos seleccionados manualmente del primer repositorio; esta selección se basó en una evaluación subjetiva en la que se determinó que estos logos eran los más similares al logo destacado **principal.jpg**.

Figura 10. *imagenes_n y imagenes_n_simil_h*



Fuente: *Yanangómez (2024)*

4.1. Compresión del Negocio

Esta etapa es crucial, ya que permite entender el contexto del negocio y alinear los objetivos del proyecto con las metas estratégicas de la organización.

4.1.1. Determinar objetivos de negocio

Los objetivos de negocio son embriones fundamentales que guían todo el proceso de minería de datos. En el contexto de la recuperación de imágenes, el principal objetivo podría ser mejorar la precisión y eficiencia de un sistema de recuperación de logotipos. Esto se traduce en alcanzar un nivel de satisfacción más alto para los usuarios, optimizando continuamente el modelo según el feedback recibido. Además, se apuntaría a la integración de estas capacidades en el registro de signos distintivos, facilitando la identificación de logos asociados a marcas registradas y protegiendo los derechos de propiedad intelectual.

4.1.2. Evaluar la Situación

En esta fase, es fundamental realizar un análisis profundo de la situación actual. Se debe considerar el contexto del negocio, los recursos disponibles, las limitaciones tecnológicas y las expectativas de los stakeholders. En el caso específico de los sistemas de recuperación de imágenes, la evaluación incluye un análisis de los modelos existentes y la calidad de los datos disponibles. Un inventario de logos y sus características se organiza en un repositorio, consistente con la recuperación esperada en diversas plataformas y su intersección con el SENADI (Servicio Nacional de Derechos Intelectuales).

4.1.3. Producir el Plan de Proyecto

El plan de proyecto define los pasos necesarios para cumplir con los objetivos establecidos. Involucra la planificación de las etapas sucesivas del proyecto, asignación de roles y tareas, así como el cronograma de trabajo para garantizar que todas las actividades se completen a un ritmo adecuado. Este plan debe ser flexible para adaptarse a cualquier cambio inesperado en el proyecto.

4.2. Compresión de los Datos

La compresión de los datos es la fase que permite al equipo entender los datos necesarios para la minería. Se deben recolectar, describir y verificar los datos para asegurar su calidad e idoneidad.

4.2.1. Recoger Datos Iniciales

En el contexto del proyecto de recuperación de imágenes, se llevaron a cabo varias etapas de recolección de datos. Esto incluyó la adquisición de 56,000 logos, extraídos de una base de datos existente. Los logos se organizaron en repositorios y bases de datos para garantizar que toda la información relevante estuviese disponible para el análisis.

4.2.2. Describir Datos

Una descripción exhaustiva de los datos permite comprender su naturaleza y estructura. Cada logo se clasifica tanto por su diseño visual como por su representación, lo cual es relevante para la posterior selección de imágenes similares. La descripción incluye aspectos como el formato de los archivos, sus dimensiones y la calidad visual.

4.2.3. Verificar la Calidad de los Datos

La verificación de calidad es una etapa fundamental para asegurar que los datos sean confiables y utilizables. Esto implica la identificación de errores, duplicados y datos incompletos. En el caso del proyecto, esto se tradujo en la normalización de las imágenes, asegurando que todas cumplían con un formato y tamaño específico de 224x224 píxeles antes de proceder al análisis.

4.3. Preparación de los Datos

Esta fase involucra la limpieza y transformación de los datos para que estén en un formato adecuado para el modelado.

4.3.1. Seleccionar Datos

Basándose en las descripciones y la calidad de los datos, se eligen los subconjuntos más relevantes que se utilizarán para el modelado. Aquí se decide cuáles logos incluir en las pruebas, basándose en su similitud y relevancia respecto a la imagen de referencia.

4.3.2. Limpiar Datos

En la limpieza de datos, se eliminan o corrigen las inconsistencias detectadas en la etapa anterior. En este proyecto, esto involucró manejar los errores que surgirían de la extracción de imágenes o problemas en el archivo, asegurando que todos los logos fuesen legibles y correctamente etiquetados.

4.3.3. Construir Datos

Se debe construir un conjunto de datos final que sintetice todos los procesos anteriores y esté listo para ser utilizado en el modelado. Esto puede incluir la creación de vectores de características a partir de las imágenes, que capturan sus atributos visuales para su posterior análisis.

4.4. Modelado

El modelado es la etapa donde se eligen técnicas específicas para analizar los datos y cumplir con los objetivos establecidos.

4.4.1. Selecciona Técnica de Modelado

En esta fase, se decide el enfoque de modelado más adecuado. Para el caso de recuperación de imágenes, se consideró utilizar técnicas de aprendizaje profundo, específicamente el modelo VGG19 que se adapta bien a la tarea de clasificación y recuperación de imágenes. Esto no solo se basa en su popularidad, sino también en su eficacia comprobada en tareas similares.

4.4.2. Generar Diseño de Prueba

Se establecieron pruebas sistemáticas para evaluar el desempeño del modelo en distintas condiciones. Esto incluyó la creación de conjuntos de prueba y control, donde se incluyeron una variedad de logos y sus respectivos pares de similitud.

4.4.3. Construir Modelo

En la construcción del modelo, se implementó el algoritmo elegido utilizando Python y sus librerías especializadas. Se llevaron a cabo entrenamientos usando los datos preprocesados, ajustando los parámetros a medida que se avanzaba en el proceso para optimizar el rendimiento del modelo.

4.4.4. Modelo de Evaluación

Una vez que el modelo fue construido y entrenado, se procedió a evaluar su rendimiento mediante métricas definidas, tales como precisión, recall y F1-score. Esto permite entender no solo la efectividad del modelo, sino también qué áreas requieren mejora.

4.4.5. Estructura de los Conjuntos de Prueba

Conjuntos de Prueba:

La definición de 10 conjuntos de prueba, denominados prueba_n respectivamente, garantiza que se evalúe el modelo de manera sistemática y controlada. Al estructurar cada prueba con un logo principal (principal.jpg) y un conjunto de 100 logos, se asegura consistencia en el tipo de comparación realizada.

Subcarpetas:

La inclusión de dos carpetas imagenes_n y imagenes_n_simil_h permite distinguir entre las imágenes a evaluar y las imágenes que se consideran como relevantes. Esta organización es crucial para la realización de comparaciones objetivas y la posterior validación de los resultados.

Selección Manual:

La selección manual de logos similares para cada carpeta se basa en una evaluación subjetiva. Esto destaca la importancia de la calibración humana en tareas de recuperación, donde a menudo el contexto puede influir en la percepción de similitud. Esto también puede introducir sesgos que deben ser analizados y considerados al interpretar los resultados

4.4.6. Procesamiento y Extracción de Características

Modelo Utilizado:

Procesamiento en Python

Para extraer las características de cada logo del repositorio, se aplicó el mismo modelo utilizado para procesar un conjunto más grande de imágenes, que contaba con un total de 56,000 logos. Entonces, para la evaluación se generó 10 archivos con las características obtenidas por el modelo vgg19, denominados **caracteristicas_imagenes_n.npy**. Posteriormente, se procedió a recuperar imágenes similares al logo principal, utilizando los modelos generados.

Una vez completada la recuperación, se llevaron a cabo comparaciones entre los resultados obtenidos automáticamente y las clasificaciones manuales realizadas previamente. Esto permitió calcular una medida aproximada de precisión del modelo en su capacidad para recuperar imágenes similares.

4.5. Evaluación

La evaluación es una etapa clave para determinar si el modelo cumple con los objetivos planteados en la fase de compresión del negocio.

4.5.1. Evaluar Resultados

Se llevaron a cabo comparaciones entre los resultados automáticos generados por el modelo y las clasificaciones manuales de los logos. Esto permitió calcular métricas de rendimiento y establecer la precisión del modelo.

4.5.2. Proceso de Revisión

En esta fase se revisan los resultados y se identifican patrones o errores. La revisión implica también solicitar feedback de los usuarios y stakeholders del proceso, para garantizar que el modelo sea intuitivo y satisfaga las expectativas.

4.5.3. Comparaciones y Resultados

- **Comparación Automática vs. Manual:**

La comparación realizada entre los resultados automáticos del modelo y las clasificaciones manuales permite identificar la precisión del modelo. Al evaluar la precisión como la proporción de imágenes correctamente recuperadas, se establece un estándar que refleja el desempeño del modelo en funciones de recuperación.

- **Tabla de Resultados:**

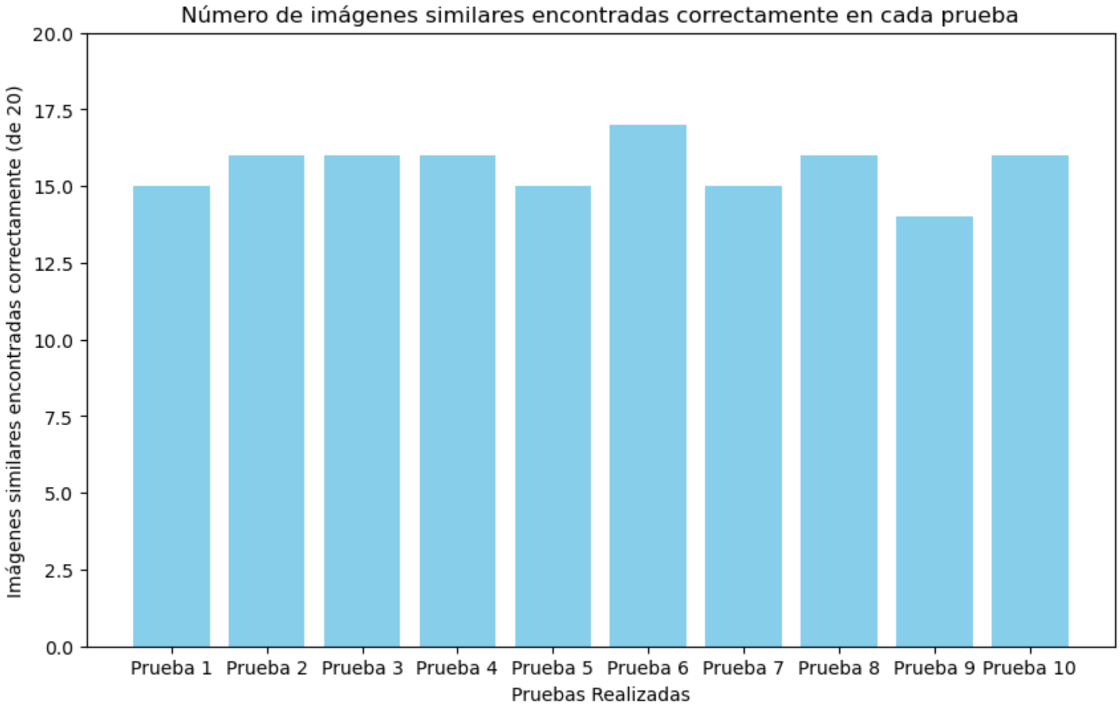
La tabla de resultados muestra un rendimiento relativo notable, con una precisión promedio del 78%. Sin embargo, el rendimiento variable en diferentes pruebas (como se observa en el caso de la Prueba 9) debe ser analizado más a fondo para entender las condiciones específicas que pueden haber influido en los resultados.

Tabla 1. *Tabla de Resultados*

Prueba	Imágenes Similares Correctas	Total de Imágenes Evaluadas	Porcentaje de Precisión
1	15	20	75%
2	16	20	80%
3	16	20	80%
4	16	20	80%
5	15	20	75%
6	17	20	85%
7	15	20	75%
8	16	20	80%
9	14	20	70%
10	16	20	80%
Total	**156**	**200**	**78% (Promedio)**

Fuente: *Yanangómez (2024)*

__Figura 11.__ Numero de imágenes similares encontradas correctamente en cada prueba



***Fuente:** Yanangómez (2024)*

- **Análisis de Precisión:**

El análisis de la precisión resalta que, aunque el modelo es efectivo en general, hay características que pueden ser optimizadas. Por ejemplo, una mayor atención a los casos donde la precisión cae por debajo del promedio puede revelar oportunidades para el ajuste del modelo o la mejora en la selección de imágenes de referencia.

- **Resultados de las Pruebas:**

En promedio, el modelo fue capaz de identificar correctamente entre 14 y 17 imágenes de un total de 20 en cada prueba, lo que indica una alta precisión. Sin embargo, hay espacio para mejorar, especialmente en la prueba 9, donde la cantidad de aciertos fue más baja (14).

- **Precisión Promedio:**

Al evaluar un total de 10 pruebas, cada una con 20 imágenes, el número total de imágenes correctamente identificadas fue de 156. Esta cifra deriva en una precisión promedio del modelo de 156/200 = 0.78, lo que significa que, en general, el modelo obtuvo un 78% de precisión en la tarea de encontrar imágenes similares, proporcionando una buena indicación del rendimiento del modelo entrenado con 56,000 imágenes.

4.2.2. Código Utilizado para la Evaluación

Figura 12. *Código Utilizado para la Evaluación*

```
# Iterar sobre cada prueba
for i in range(1, 11):
        # Directorio de la prueba actual
        dir_prueba = os.path.join(base_dir, f'prueba{i}')

        # Directorio de imágenes
        dir_imagenes = os.path.join(dir_prueba, f'imagenes_{i}')

        # Ruta del archivo .npy donde se guardarán las características
        features_path = os.path.join(dir_prueba, f'caracteristicas_imagenes_{i}.npy')

        # Calcular y guardar las características (comentar esta línea cuando ya se tenga los .npy)
        #calcular_y_guardar_caracteristicas(dir_imagenes, features_path)
        print(f'Características de la prueba {i} calculadas y guardadas en {features_path}')

        # Ruta de la imagen principal
        imagen_principal = os.path.join(dir_imagenes, 'principal.jpg')

        # Extraer características de la imagen principal
        features_principal = extraer_caracteristicas(imagen_principal)

        # Encontrar las 20 imágenes más similares
        nombres_resultados, resultados = encontrar_imagenes_similares(features_path, features_principal, dir_imagenes, n_similares=21)

        print(f'Resultados para prueba {i}:')
        print(resultados)

        # Crear un diagrama de 10x2 con las imágenes similares
        fig, axes = plt.subplots(4, 5, figsize=(6, 4))
        fig.suptitle(f'Resultados para prueba {i}', fontsize=16)
        for ax, img_name in zip(axes.ravel(), nombres_resultados):
            img_path = os.path.join(dir_imagenes, img_name)
            img = mpimg.imread(img_path)
            ax.imshow(img, aspect='auto')  # 'aspect' controla la relación de aspecto de la imagen
            #ax.imshow(img)
            ax.set_title(img_name.replace("LOGO_","").split('_')[0], fontsize=8)
            ax.axis('off')

        plt.tight_layout()
        plt.subplots_adjust(top=0.85)
        plt.show()

        # Directorio de imágenes similares seleccionadas manualmente
        dir_simil_h = os.path.join(dir_prueba, f'imagenes_{i}_simil_h')

        # Lista de imágenes seleccionadas manualmente
        imagenes_manuales = [os.path.basename(nombre) for nombre in os.listdir(dir_simil_h) if nombre.endswith('.jpg')]

        # Comparación de los resultados
        aciertos = len(set(nombres_resultados).intersection(imagenes_manuales))
        print(f'Número de imágenes similares encontradas correctamente: {aciertos} de 20')
        print('---' * 10)

        # Agregar el resultado de la prueba actual a la lista de resultados
        resultados_pruebas.append(aciertos)


# Mostrar los resultados finales
print("Resultados finales de las pruebas:", resultados_pruebas)
14.4s
```

Fuente: *Yanangómez (2024)*

4.2.3. Métricas Utilizadas para la Evaluación

Tabla 2. *Métricas Utilizadas para la Evaluación*

Métrica	Valor	Explicación
MAP@20	0.7726	**Mean Average Precision at 20** mide la precisión promedio en las primeras 20 posiciones para cada prueba. Un valor de 0.7726 indica que, en promedio, el modelo tiene una precisión de aproximadamente 77.26% en las primeras 20 imágenes recuperadas. Este es un buen indicador de que las imágenes relevantes suelen estar entre las primeras posiciones.
Matriz de Confusión	[[0 4] [0 16]]	La **matriz de confusión** muestra que hubo 16 verdaderos positivos (imágenes correctamente identificadas) y 4 falsos negativos (imágenes relevantes que no fueron identificadas). No hubo falsos positivos ni verdaderos negativos en este caso porque solo se está considerando ai las imágenes recuperadas son relevantes o no.
Rango de Precisión	3	El **rango de precisión** es la diferencia entre la precisión máxima y mínima en las diferentes pruebas. Un rango de 3 indica que hay cierta variabilidad en el rendimiento del modelo entre las diferentes pruebas, pero esta variabilidad no es extrema, lo que sugiere que el modelo es razonablemente consistente.
Posición Promedio de Imágenes Similares	9.125	Esta métrica mide la posición promedio de las imágenes relevantes dentro del conjunto de las 20 imágenes recuperadas por el modelo. Un valor de 9.125 sugiere que las imágenes correctas suelen estar alrededor de la posición 9 de las 20 imágenes devueltas.
F1-Score	0.8889	El **F1-Score** es la media armónica de la precisión y el recall, y en este caso es 0.8889, lo que indica un equilibrio bastante bueno entre la precisión y el recall del modelo. Un valor cercano a 1 indica un buen rendimiento en términos de encontrar imágenes relevantes y evitar falsos positivos.

Fuente: *Yanangómez (2024)*

- **MAP@20:**

Una métrica clave fue la Precisión Promedio en las 20 primeras posiciones (MAP@20), que indica cuán bien el modelo se desempeña en recuperar imágenes relevantes en las mejores posiciones. La puntuación de 0.7726 es un indicador alentador, pero se podrían considerar métricas adicionales que exploren el rendimiento en posiciones más bajas.

- **Matriz de Confusión:**

La matriz de confusión proporciona información valiosa sobre los tipos de errores que comete el modelo. Con 4 falsos negativos y 16 verdaderos positivos, es crucial investigar los casos perdidos para ajustar el modelo. Este análisis puede ofrecer pistas sobre qué características o clases de logos no se están reconociendo adecuadamente.

- **Evaluaciones F1-Score:**

El F1-Score de 0.8889 sugiere un buen equilibrio entre precisión y recuperación. Sin embargo, se debe explorar cómo se podría ajustar el umbral del modelo para mejorar este índice en contextos donde la minimización de falsos negativos es crucial.

- **Rango de Precisión:**

3. Refleja una variabilidad razonable en el rendimiento del modelo.

- **Posición Promedio de Imágenes Similares**

9.125. Indica que las imágenes correctas suelen estar alrededor de la posición 9 de las 20 imágenes recuperadas.

4.6. Despliegue

La fase de despliegue involucra la implementación del modelo en un ambiente operativo real.

4.6.1. Casos de Utilización

Incluir visualizaciones efectivas de los resultados tales como curvas de precisión-recall, gráficos de la matriz de confusión, y ejemplos visuales de imágenes recuperadas—sería beneficioso para interpretar y comunicar los resultados de la metodología. Las visualizaciones no solo facilitan la comprensión de los hallazgos, sino que también pueden servir para identificar patrones que no son evidentes en las tablas numéricas.

4.6.2. Conclusiones y Futuras Mejora

Este análisis proporciona una perspectiva integral sobre la evaluación del modelo y sus resultados. A partir de aquí, el enfoque debe dirigirse hacia:

- La mejora continua del modelo, basándose en los errores identificados y la posibilidad de realizar ajustes en la extracción de características.

- La integración de una evaluación más amplia mediante métricas adicionales que aborden específicamente el rendimiento en diferentes contextos y conjuntos de datos.
- La consideración de ciclos de retroalimentación en los que se incorporen evaluaciones subjetivas continuas para ajustar y adaptar el modelo a los criterios cambiantes de similitud de logos.

El enfoque sistemático presentado aquí no solo evidencia el estado actual del modelo, sino que también sienta las bases para futuras investigaciones y optimizaciones que podrían llevar a un rendimiento aún más vigoroso en la recuperación de imágenes similares.

4.6.1. Implementación del Plan

Se ejecuta el plan de despliegue, en el que se integra el modelo en una aplicación que permite la recuperación de imágenes de manera accesible para el usuario final.

4.6.2. Seguimiento y Mantenimiento del Plan

Al implementar el sistema, es esencial establecer procedimientos de seguimiento, donde se recopila información sobre el uso del modelo y se realizan ajustes según sea necesario para optimizar su rendimiento.

4.6.3. Análisis y Mejora del Proceso de Registro de Signos Distintivos ante el SENADI

A continuación, se detallará y analizará de manera exhaustiva el proceso hasta la obtención del producto final en lo que respecta al registro de signos distintivos ante el SENADI (Servicio Nacional de Derechos Intelectuales).

4.6.4. Tipos de Signos Distintivos

El SENADI permite el registro de distintos tipos de signos distintivos, cada uno con características y requisitos específicos:

1. **Denominativo**: Este tipo de signo se refiere a la identificación verbal, compuesto únicamente por palabras, letras o números sin elementos gráficos que lo acompañen.
2. **Figurativo**: Incluye cualquier forma gráfica o diseño que represente una marca.
3. **Mixto**: Combinación de elementos denominativos y figurativos.
4. **Tridimensional**: Se refiere a la forma tridimensional de un producto o embalaje.
5. **Sonoro/Auditiva**: Protege sonidos únicos que pueden servir para identificar productos o servicios.

6. **Olfativo**: Señala la innovación de proteger aromas específicos asociados a productos o servicios.
7. **Táctil**: Protección de texturas o características que se pueden percibir al tacto.

Exceptuando los signos de naturaleza "Denominativa", todos los demás requieren la presentación de un logo, que es una representación visual clave para el reconocimiento de la marca.

Figura 13. *Formato único de registro de signos distintivos*

Fuente: *Yanangómez (2024)*

4.6.5. Proceso de Registro

Para iniciar el proceso de registro, el usuario debe llenar un formulario en línea. Una vez completado, el trámite es publicado mensualmente en "La Gaceta", una publicación digital que incluye detalles del registro y los logos asociados. Este acceso es fundamental para garantizar la transparencia del proceso y permite al público verificar y revisar los registros en curso.

__Figura 14.__ Signo de solicitud

Signo de solicitud: Mixto
Royal Stark
Nombre comercial
País: Ecuador
(511) 91
(220) 2019-06-18
(210) SENADI-2019-43538
(730) FEMAR S.A.
(740) ERIKA MARICRUZ GUZMÁN ORTEGA
ACTIVIDADES: Restringida
Actividades comerciales relacionadas con la comercialización, importación, exportación, compra, venta, permuta, almacenaje, alquiler de aparatos, equipos y accesorios eléctricos de uso doméstico.

Signo de solicitud: Denominativo
HAGAMOS BACA
Marca de servicios
País: Ecuador
(511) 35
(220) 2019-07-09
(210) SENADI-2019-43954
(730) COMPRASIGMA S.A.
(740) PABLO DAVID ORDOÑEZ BACA
SERVICIOS: Restringida
servicios de abastecimiento para terceros [compra de productos y servicios para otras empresas], asesoramiento sobre dirección de empresas, suministro de información y asesoramiento comerciales al consumidor en la selección de productos y servicios, asistencia en la dirección de negocios, investigación de marketing, consultoría sobre dirección de negocios, consultoría sobre organización de

Signo de solicitud: Mixto
SUMESA Farma - Bella
Marca de producto
País: Ecuador
(511) 3
(220) 2019-10-01
(210) SENADI-2019-51651
(730) SUMESA S.A.
(740) JOSÉ ALFREDO GONZÁLEZ GARAVITO
PRODUCTOS: Restringido
cosméticos, cremas cosméticas, productos cosméticos para el cuidado de la piel, mascarillas de belleza, pomadas para uso cosmético, preparaciones de tocador*

Signo de solicitud: Mixto
MURANO MÁS LOGOTIPO
Marca de servicios
País: Ecuador
(511) 44
(220) 2019-09-26
(210) SENADI-2019-70197
(730) CRISTIAN CAMILO CARMONA
(740) ---
SERVICIOS: Restringida
servicios de salones de belleza, servicios de manicura, salones de belleza
Servicios de lavado corte, recorte, peinado, teñido, coloración, ondulación, alisado del cabello, manicura, pedicura para hombres y mujeres.

Fuente: *Yanangómez (2024)*

4.6.6. Repositorio de Logos

El SENADI no cuenta con un repositorio directo de acceso a los logos registrados, sino que utiliza un sistema externo, **Alfresco Share – Alfresco Content Services**, como un repositorio de documentos. Este sistema facilita la gestión y almacenamiento de los datos y archivos adjuntos de los diferentes tipos de productos que la Institución permite registrar.

Figura 15. *Alfresco Content Services*

Fuente: *Yanangómez (2024)*

4.6.7. Obtención de Logos

Para construir un repositorio de logos, se realizó una descarga masiva desde Alfresco mediante un script en Java. El resultado fue un conjunto de imágenes de dimensiones variadas y posiciones irregulares. Esta variedad en las características de los logos presentó un desafío para la normalización posterior.

Figura 16. *Logo en alfresco*

Fuente: *Yanangómez (2024)*

Figura 17. Script de java

```java
public void getSignos(String ruta) {
    int n = 660;
    int f = 737;

    System.out.println("Se obtendrá datos desde la gaceta " + n + " hasta la gaceta " + f);

    Controlador c = new Controlador();
    List<PpdiSolicitudSignoDistintivo> signos = new ArrayList<>();
    while (n <= f) {
        List<PpdiSolicitudSignoDistintivo> aux = c.getSignosByGaceta(n);
        System.out.println("Gaceta " + n + ": " + aux.size());
        signos.addAll(aux);
        n++;
    }
    System.out.println("tamaño total de todas las gacetas: " + signos.size());
    int total = signos.size();
    int contador = 0;
    String timein = new Date().toGMTString();
    System.out.println("*************** EMPEZANDO DESCARGA ***************");
    for (int i = 0; i < signos.size(); i++) {
        try {
            PpdiSolicitudSignoDistintivo signo = signos.get(i);
            System.out.println("------------------" + signo.getNumeroTramite() + "------");
            String downloadedlogo = c.descargaLogo(ruta, signo.getNumeroExpediente());
            System.out.print("registro " + (i + 1) + ": ");
            if (!downloadedlogo.trim().isEmpty()) {
                System.out.println("Logo descargado: " + downloadedlogo+"\n");
                contador++;
            } else {
                System.err.println("No se descargó el logo del expediente: " + signo.getNumeroExpediente()+
                        ", sugerencia: LOGO_"+signo.getNumeroTramite()+"_"+signo.getCodigoSolicitudSigno());
            }
        } catch (IOException ex) {
            Logger.getLogger(Getalfrescoimages.class.getName()).log(Level.SEVERE, null, ex);
            System.err.println("\nHubo un problema al descargar el logo del expediente " +
                    signos.get(i).getNumeroExpediente()+", sugerencia: LOGO_"+signos.get(i).getNumeroTramite()+
                    "_"+signos.get(i).getCodigoSolicitudSigno());
        }
    }
    String timeout =  new Date().toGMTString();
    System.out.println("Terminado.  Se han descargado "+ contador+" logos de "+total+" posibles");
    System.out.println("Empezó: "+timein+", Terminó: "+timeout);
}
```

Fuente: *Yanangómez (2024)*

4.6.8. Normalización de Logos

El siguiente paso fue la normalización de los logos para garantizar que todos fueran cuadráticos y de la misma extensión (.jpg). Para ello, se desarrolló un segundo script en Java que ajustó cada imagen. Este proceso consistió en rellenar las imágenes hasta lograr que todas tuvieran dimensiones de 224x224 píxeles.

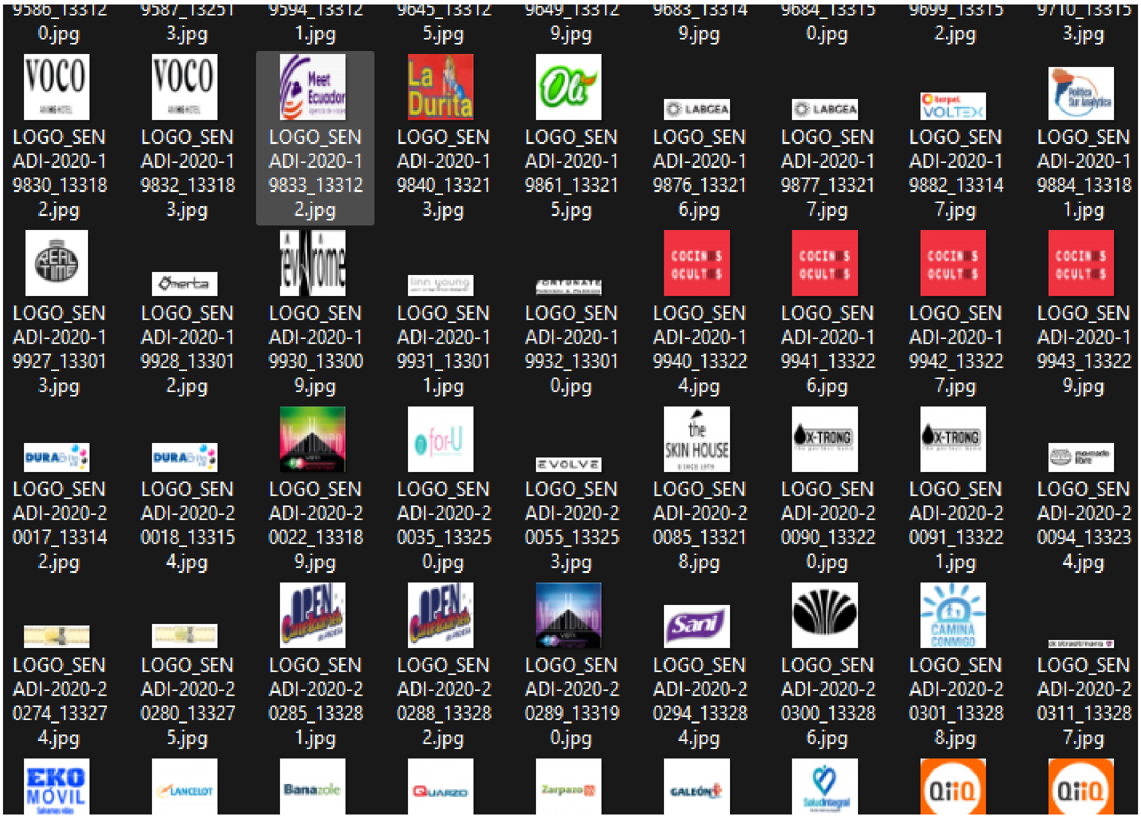
Figura 18. *Script corriendo*

```
registro 593: Logo descargado: LOGO_SENADI-2021-68673_171397.jpg

--------------------SENADI-2021-56777------
registro 594: Logo descargado: LOGO_SENADI-2021-56777_167760.jpg

--------------------SENADI-2021-63861------
registro 595: Logo descargado: LOGO_SENADI-2021-63861_169978.jpg

--------------------SENADI-2021-58946------
registro 596: Logo descargado: LOGO_SENADI-2021-58946_168221.jpg

--------------------SENADI-2021-59661------
registro 597: Logo descargado: LOGO_SENADI-2021-59661_168346.jpg

--------------------SENADI-2021-68850------
registro 598: Logo descargado: LOGO_SENADI-2021-68850_170226.jpg

--------------------SENADI-2021-62710------
registro 599: Logo descargado: LOGO_SENADI-2021-62710_169555.jpg

--------------------SENADI-2021-70006------
registro 600: Logo descargado: LOGO_SENADI-2021-70006_171941.jpg

--------------------SENADI-2021-55074------
registro 601: Logo descargado: LOGO_SENADI-2021-55074_166999.jpg

--------------------SENADI-2021-61897------
registro 602: Logo descargado: LOGO_SENADI-2021-61897_169398.jpg

--------------------SENADI-2021-67170------
registro 603: Logo descargado: LOGO_SENADI-2021-67170_170150.jpg

--------------------SENADI-2021-65525------
registro 604: Logo descargado: LOGO_SENADI-2021-65525_170260.jpg

--------------------SENADI-2021-67739------
registro 605: Logo descargado: LOGO_SENADI-2021-67739_171267.jpg

--------------------SENADI-2021-70314------
registro 606: Logo descargado: LOGO_SENADI-2021-70314_171839.jpg

--------------------SENADI-2021-34209------
registro 607: Logo descargado: LOGO_SENADI-2021-34209_161433.jpg
```

Fuente: *Yanangómez (2024)*

La normalización no solo asegura la uniformidad estética de los logos, sino también la compatibilidad para futuras aplicaciones, como la implementación de modelos de inteligencia artificial para la generación de la gaceta.

Figura 19. *Captura de los logos obtenidos a partir del script*



Fuente: *Yanangómez (2024)*

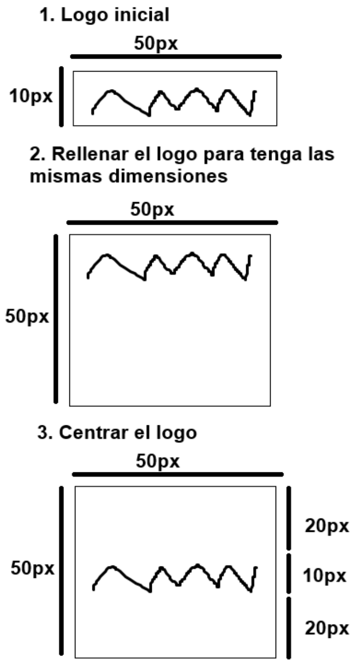
En la Figura 19 se pueden apreciar que las imágenes no siguen un patrón especifico en sus dimensiones y diseño, por ello se realizó un proceso de redimensionamiento de los logos de la siguiente manera:

De cada logo, partiendo de cuál es la medida más grande en su estructura, es decir el ancho o el alto, se rellenó de blanco el lado más pequeño hasta que sea igual al otro (de esta manera se logra hacer un cuadrado), desplazando el logo a la posición inicial de la nueva imagen creada.

Luego se procede a centrar el logo, calculando en base a su tamaño pequeño original y al tamaño nuevo generado, la nueva posición que tendrá el logo en la visualización.

Posteriormente se redimensiona el nuevo logo cuadrático a 224 x 224 px, para manejar los mismos tamaños en todas las imágenes.

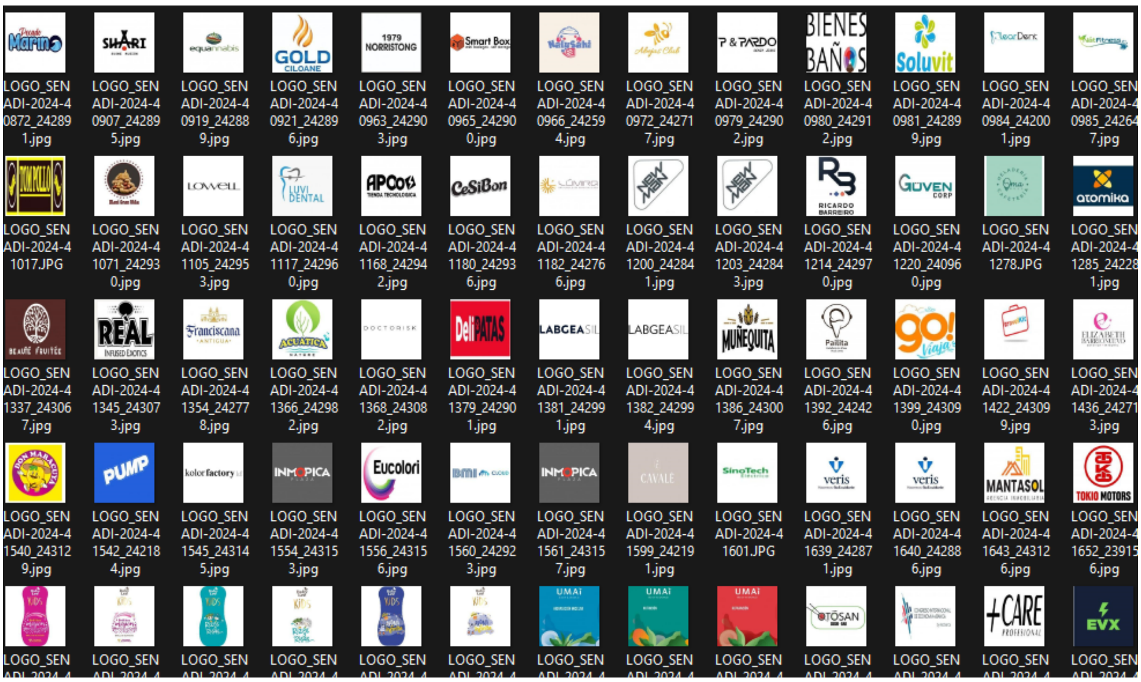
Figura 20. *Renderización y normalización del logo inicial*



Fuente: *Yanangómez (2024)*

Como resultado todos los logos tienen las mismas dimensiones y no han perdido sus características iniciales, es decir no se han perdido sus características:

Figura 21. *Logos normalizados*



Fuente: *Yanangómez (2024)*

4.6.9. Creación de Modelos de Predicción

Una vez normalizadas las imágenes, se procedió a la creación de modelos de predicción utilizando Python y sus librerías especializadas.

1. **Modelo VGG19**: Este modelo se entrenó empleando un conjunto de 56,278 logos, optimizando así la capacidad de reconocimiento visual.

Figura 22. *Entrenamiento del modelo VGG19*

```python
# Función para calcular y guardar características de todas las imágenes en un directorio
def calcular_y_guardar_caracteristicas(dir_imagenes, nombre_archivo):
    # Obtener lista de imágenes en el directorio
    imagenes = [os.path.join(dir_imagenes, nombre) for nombre in os.listdir(dir_imagenes) if nombre.endswith('.jpg')]

    # Calcular características de cada imagen y guardar en una lista
    features_list = [extraer_caracteristicas(img) for img in imagenes]

    # Guardar características en un archivo .npy
    np.save(nombre_archivo, np.array(features_list))
```

```python
# Cargar el modelo VGG19 preentrenado, se debe activar solo cuando se creen nuevas características
modelo_vgg19 = VGG19(weights='imagenet', include_top=False)

# Directorio de imágenes
dir_logos = 'D:\\maestria_micha\\tesis\\search_logo_1000\\logo_total'
# Ruta del archivo donde se guardarán las características
features_path = 'caracteristicas_imagenes_total.npy'
```

```python
# Calcular y guardar características de las imágenes (ya se ejecutó con anterioridad por ello se encuentra comentado)
# Se demora al rededor de unos 15 a 20 mins para 1000 y para 10mil al rededor de una hora y para 55mil 5 horas
calcular_y_guardar_caracteristicas(dir_logos, features_path)

print("Cálculos terminados")
```

Fuente: *Yanangómez (2024)*

2. **DeepImageSearch**: Este modelo alternativo que utiliza la librería DeepImageSearch, facilita la implementación de un sistema de búsqueda y recuperación de imágenes basado en similitud visual, se implementó como alternativa para la búsqueda de similitudes.

Figura 23. *DeepImageSearch*

```python
# Cargamos el modelo preentrenado vgg19, el cual es un modelo extraído de un entrenamiento anterior que utiliza imágenes
# de un proyecto ya realizado con millones de imágenes
st_rgb = Search_Setup(image_list=image_list_rgb, model_name='vgg19', pretrained=False, image_count=56278)
```

```
Please Wait Model Is Loading or Downloading From Server!
Model Loaded Successfully: vgg19
```

```python
#Entrenamiento (dura aproximadamente unos 5horas), como ya se hizo con anterioridad, no es necesario ponerlo de nuevo
st_rgb.run_index();
```

```python
# Se obtiene los paths y los features (características) de los logos
metadata_rgb = st_rgb.get_image_metadata_file()
```

Fuente: *Yanangómez (2024)*

4.6.10. Implementación de la Aplicación

Con los modelos entrenados, se creó un servidor HTTP en Python que habilita la interacción con los modelos de predicción. En paralelo, se desarrolló una aplicación en Java que permite a los usuarios enviar peticiones al servidor y recibir las respuestas, facilitando así la consulta y el reconocimiento de logos.

Figura 24. *Librerías Python utilizadas*

```
from DeepImageSearch import Load_Data, Search_Setup
from urllib import parse, request
from http.server import HTTPServer, BaseHTTPRequestHandler
import os
import tempfile
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity
from tensorflow.keras.applications import VGG19
from keras.applications.vgg19 import preprocess_input
from tensorflow.keras.preprocessing import image
```

Fuente: *Yanangómez (2024)*

El siguiente paso fue crear un aplicativo .py el cual permite arrancar el servidor HTTP que tendrá la funcionalidad de los dos modelos de predicción implementados:

Figura 25. *Función DeepImageSearch*

```
# Función para encontrar imágenes similares con DeepImageSearch
def buscar_con_deepimagesearch(image_path, number_of_images):
    result = st.get_similar_images(image_path=image_path, number_of_images=number_of_images)
    logos = ', '.join([valor.replace('data\\','') for valor in result.values()])
    logos = logos.replace("logo_total\\","")
    return logos
```

Fuente: *Yanangómez (2024)*

Figura 26. *Función VGG19*

```
# Función para encontrar imágenes similares con VGG19 directo
def buscar_con_vgg19(image_path, number_of_images):
    dir_logos = 'logo_total'
    features_path = 'caracteristicas_imagenes_total.npy'
    features_consulta = extraer_caracteristicas(image_path)
    logos = encontrar_imagenes_similares(features_path, features_consulta, dir_logos, n_similares=number_of_images)
    return logos
```

Fuente: *Yanangómez (2024)*

El siguiente código muestra la implementación del servidor y como se captura la petición de información enviada a través de un 'POST' para la recuperación de los logos similares, haciendo la búsqueda en el modelo seleccionado.

Figura 27. *Implementación del servidor HTTP*

```python
class SimpleHTTPRequestHandler(BaseHTTPRequestHandler):

    def do_POST(self):
        print("Petición recibida")
        content_length = int(self.headers['Content-Length'])
        data = self.rfile.read(content_length)
        data = data.decode().replace('imagen_web=', '')
        data = parse.unquote(data).replace('%20', ' ')
        print("Frase unquote recibida: " + data)
        data = data.split(",")
        image_path = data[0]
        number_of_images = int(data[1])
        modelo = int(data[2])

        # Verificar si la ruta es una URL
        if image_path.startswith('http://') or image_path.startswith('https://'):
            response = request.urlopen(image_path)
            with tempfile.NamedTemporaryFile(delete=False, suffix=".jpg") as tmp_file:
                tmp_file.write(response.read())
                tmp_file_path = tmp_file.name
            image_path = tmp_file_path

        # Ejecutar el modelo seleccionado
        if modelo == 1:
            logos = buscar_con_deepimagesearch(image_path, number_of_images)
        elif modelo == 2:
            logos = buscar_con_vgg19(image_path, number_of_images)
        else:
            logos = "Modelo no válido"

        print(logos)
        # Regresar respuesta a la petición HTTP
        self.send_response(200)
        self.send_header("Access-Control-Allow-Origin", "*")
        self.end_headers()
        self.wfile.write(str(logos).encode())

print("Iniciando el servidor...")
server = HTTPServer(('localhost', 8000), SimpleHTTPRequestHandler)
server.serve_forever()
```

Fuente: *Yanangómez (2024)*

El aplicativo que interactúa con el Usuario se lo hizo en Java y para la comunicación con el servidor Python se utilizó Javascript.

Script implemente en Jascript que envía la petición al servidor Python, y obtiene la respuesta de este para visualizarlo en Java:

__Figura 28.__ Script en javascript

```javascript
function searchLogoPython(urlima, urgeneral) {
    mostrarStatusD();
    frase = '_____________EMPEZANDO PETICIÓN_____________';
    console.log(frase);
    var div = document.getElementById("resultado");

    //console.log(image.src);
    ima = urlima;

    console.log("Imagen: " + ima);

    numero_similitudes = document.getElementById('allform:nsimil_input').value;
    //modelo = document.getElementById("allform:model_simil").value;
    modelo = document.querySelector("input[name='allform:model_simil']:checked").value;
    //urgeneral = urgeneral.replace("\\", "/");
    //console.log("Destination: " + urgeneral);
    console.log("numero_imagenes: " + numero_similitudes);
    console.log("modelo seleccionado: " + modelo);
    div.innerHTML = "";
    var model = 0;
    if (modelo === 'DEEPIMAGE') {
        model = 1;
    } else {
        model = 2;
    }
    $.post("http://localhost:8000", {imagen_web: ima + "," + numero_similitudes + "," + model},
            function (response) {
                console.log("Resultado: " + response);

                subs = response.split(",");
                if (model === 1) {
                    // Enviar la lista de imágenes locales al servidor Java para copiarlas
                    $.post("copyImages", {images: subs.join(",")}, function (copyResponse) {
                        console.log("Imágenes copiadas en el servidor: " + copyResponse);

                        // Mostrar las imágenes ahora desde la carpeta resources/images
                        for (var i = 0; i < subs.length; i++) {
                            if (subs[i] !== "undefined") {

                                var container = document.createElement("div");
                                container.className = "image-container";

                                imagen = document.createElement("img");
                                //imagen.src = urgeneral + "found_logos/" + subs[i].split("/").pop() + ".xhtml";
                                imagen.src = urgeneral + "found_logos/" + subs[i].trim().split("___")[0] + ".xhtml";
                                console.log(imagen.src + "\n");
                                // Asignar el evento onclick para mostrar el dialogo
                                nombretramite = subs[i].split("_")[1];  // Nombre del archivo;

                                (function (name) {
                                    imagen.onclick = function () {
                                        console.log(name);
                                        showImageInfo(name);
                                    };
                                })(nombretramite);


                                var imageName = document.createElement("div");
                                imageName.className = "image-label";
                                imageName.textContent = nombretramite;

                                container.appendChild(imagen);
                                container.appendChild(imageName);
                                div.appendChild(container);
                            }
                        }
                    });
                } else {
                    // Enviar la lista de imágenes locales al servidor Java para copiarlas
                    $.post("copyImages", {images: subs.map(sub => sub.split("___")[0]).join(",")}, function (copyResponse) {
                        console.log("Imágenes copiadas en el servidor: " + copyResponse);

                        // Mostrar las imágenes ahora desde la carpeta resources/images
                        for (var i = 0; i < subs.length; i++) {
                            if (subs[i] !== "undefined") {

                                var container = document.createElement("div");
                                container.className = "image-container";

                                imagen = document.createElement("img");
                                imagen.src = urgeneral + "found_logos/" + subs[i].trim().split("___")[0] + ".xhtml";
                                console.log(imagen.src + "\n");
                                // Asignar el evento onclick para mostrar el dialogo
                                //nombretramite = subs[i].split("_")[1];  // Nombre del archivo;

                                var partes = subs[i].split("___");
                                var nombretramite = partes[0].split("_")[1];
                                var similitud = partes[1];
                                (function (name) {
                                    imagen.onclick = function () {
                                        console.log(name);
                                        showImageInfo(name);
                                    };
                                })(nombretramite);


                                var imageName = document.createElement("div");
                                imageName.className = "image-label";
                                imageName.textContent = nombretramite + " (" + decimalToPercentage(similitud) + ")";

                                container.appendChild(imagen);
                                container.appendChild(imageName);
                                div.appendChild(container);
                            }
                        }
                    });
                }
                console.log("_________________TERMINADO______________________");
                ocultarStatusD();
            }
    );

}
```

__Fuente:__ Yanangómez (2024)

Al ejecutar el pedido desde la app web java de una imagen de entrada determinada, seleccionando el modelo VGG19 y solicitando la obtención de los 20 logos más similares, esta es la salida del log del servidor Python:

Figura 29. *Salida del log del servidor Python*

```
Petición recibida
Frase unquote recibida: http://localhost:8080/logosearch-web/javax.faces.resource/images/exito.png.xhtml?ts=1
724725204601,20,2
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 675ms/step
LOGO_SENADI-2021-31988_160691.jpg___0.999345, LOGO_SENADI-2023-92137_231232.jpg___0.714038, LOGO_SENADI-2023-
92135_231230.jpg___0.714038, LOGO_SENADI-2024-31408_240483.jpg___0.707454, LOGO_SENADI-2022-78704_201328.jpg_
__0.707136, LOGO_SENADI-2024-44513_243629.jpg___0.698513, LOGO_SENADI-2024-44523_243633.jpg___0.698513, LOGO_
SENADI-2023-83705_228829.jpg___0.692962, LOGO_IEPI-2017-71823_76596.jpg___0.690472, LOGO_SENADI-2021-74574_17
2761.jpg___0.688085, LOGO_SENADI-2023-40644_217621.jpg___0.685491, LOGO_SENADI-2023-85192_229304.jpg___0.6833
70, LOGO_SENADI-2020-20751_133353.jpg___0.682897, LOGO_SENADI-2022-45609_191609.jpg___0.678748, LOGO_SENADI-2
022-10699_181437.jpg___0.677224, LOGO_SENADI-2022-49528_192778.jpg___0.676589, LOGO_SENADI-2022-49533_192781.
jpg___0.676589, LOGO_SENADI-2022-49532_192780.jpg___0.676589, LOGO_SENADI-2022-49529_192779.jpg___0.676589, L
OGO_SENADI-2021-68364_170956.jpg___0.676180
127.0.0.1 - - [26/Aug/2024 21:20:16] "POST / HTTP/1.1" 200 -
```

Fuente: *Yanangómez (2024)*

4.6.11. Conclusiones Generales

El proceso descrito desde la recolección de logos hasta la creación de modelos de predicción refleja un enfoque metódico y sistemático para la recuperación de los logos más similares de signos distintivos, dada una imagen de entrada. Este método no solo optimiza la presentación visual de los logos, sino que también asegura su integración en sistemas de inteligencia artificial para el reconocimiento y la búsqueda eficiente. Además, la transparencia del proceso a través de "La Gaceta" y el acceso público a la información registrada son aspectos fundamentales que fortalecen la confianza en el sistema de propiedad intelectual del país.

Este esfuerzo por normalizar y digitalizar el manejo de signos distintivos no solo moderniza el proceso de registro y mejora la protección de los derechos intelectuales de los usuarios del SENADI, sino que también abre nuevos caminos para el uso de tecnología en la protección de signos distintivos y otros tipos de productos que permite el registro la Institución, lo que es crucial en un entorno empresarial cada vez más competitivo y globalizado.

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- **Implementación Exitosa**: El desarrollo de un Sistema de Recuperación de Imágenes de Signos Distintivos basado en Machine Learning se ha ejecutado de manera efectiva, logrando automatizar el proceso de identificación y clasificación de logos utilizando el modelo VGG19. Este avance representa un cambio significativo en la gestión de signos distintivos para el SENADI, permitiendo un trabajo más rápido y preciso.
- **Mejora en la Eficiencia**: La automatización proporcionada por el sistema ha demostrado mejorar la eficiencia en el proceso de detección de infracciones de derechos de propiedad intelectual. Esto no solo reduce el tiempo necesario para las revisiones manuales, sino que también disminuye la probabilidad de errores humanos.
- **Precisión del Modelo**: Los resultados de las pruebas muestran que el modelo tiene una precisión promedio del 78% en la identificación de imágenes similares. Este nivel de precisión es alentador; sin embargo, se identifican áreas de mejora, especialmente en la selección de los datos de entrenamiento y en la extracción de características.
- **Contribución a la Propiedad Intelectual**: Este sistema no solo aborda problemas prácticos en la protección de signos distintivos, sino que también representa un avance significativo en el uso de tecnología avanzada dentro del marco de la propiedad intelectual en Ecuador. Facilita la adaptación del SENADI a un panorama empresarial cada vez más digitalizado y complejo.
- **Futuras Oportunidades de Mejora**: Se identifican oportunidades para incrementar la precisión del modelo mediante ajustes en los algoritmos y la inclusión de técnicas de aumento de datos. Estas mejoras son cruciales para enfrentar la creciente complejidad de los signos distintivos en el mercado.

5.2. RECOMENDACIONES

- **Mantenimiento y Actualización del Sistema**: Se recomienda establecer un calendario de mantenimiento regular para el sistema, así como actualizar continuamente los modelos de Machine Learning con nuevos datos para mejorar la precisión y adaptabilidad del sistema ante inevitablemente cambiantes patrones de uso.
- **Capacitación Continua de Usuarios**: Es fundamental implementar programas de capacitación continua para el personal del SENADI sobre el uso y manejo del sistema. Esto asegurará que el personal esté bien equipado para utilizar la tecnología efectivamente y contribuirá a maximizar los beneficios del sistema.
- **Evaluación y Retroalimentación Regular**: Se sugiere implementar un mecanismo de evaluación y recopilación de feedback periódico por parte de los usuarios. Esto ayudará a identificar áreas de mejora en la funcionalidad del sistema y proporcionará información valiosa para futuros desarrollos.
- **Exploración de Nuevas Tecnologías**: Se recomienda investigar y evaluar tecnologías emergentes como el aprendizaje profundo (Deep Learning) y métodos de inteligencia artificial que puedan potencialmente mejorar aún más la eficiencia y efectividad del sistema en el futuro.
- **Colaboración Interinstitucional**: Fomentar la colaboración con otras áreas gubernamentales y entidades privadas para compartir avances y experiencias en la implementación de sistemas de inteligencia artificial en la protección de la propiedad intelectual. Esto podría llevar a un enfoque más holístico y coordinado en la gestión de signos distintivos en el ámbito nacional.

Bibliografía

Apd. (04 de 04 de 2019). *¿Cuáles son los tipos de algoritmos del machine learning?* Obtenido de https://www.apd.es/algoritmos-del-machine-learning/

AWS. (2022). *¿Qué es la ciencia de datos?* Obtenido de https://acortar.link/zLTe55

B, J. (2015). *Metodología Fundamental IBM.* Obtenido de https://www.ibm.com/downloads/cas/6RZMKDN8

Benitez Ruíz, A. M. (2015). *Animación de un modelo 3D del robot Darwin-OP utilizando Kinect. Pistas Educativas, Instituto Tecnológico de Celaya.*

BizMetriks. (2013). *Los datos se están convirtiendo en la nueva materia prima de los negocios.* Obtenido de http://www.bizmetriks.com/metodologia.html

Cecco, C. N. (02 de 09 de 2021). *¿Cómo puede la inteligencia artificial mejorar la salud de los latinoamericanos?* Obtenido de https://acortar.link/k70wKt

Celina. (2006). *Centro Médico Celina*. Obtenido de https://medicelina.com/nosotros#contenido

Chace, C. (2015). Surviving AI.

Correa et al., C. R. (2017). Correa et al., C. R., Pinzón Delgado, J. T., Aragón Barrero, M. A., Perdomo Santos, P. A., Alfonso Moreno, F. L., & Director. (2017).

Estrella, À. (2020). *Aplicaciones basadas en aprendizaje automatico (machine learnin) en plataforma de bajo consumo.* Obtenido de https://oa.upm.es/66520/1/TFG_ALVARO_ESTRELLA_OLIVA.pdf

Figueiras, S. (20 de 09 de 2021). *¿CONOCES JUPYTER NOTEBOOK?* Obtenido de https://www.ceupe.mx/blog/conoces-jupyter-notebook.html

Gartner. (2018). Hype Cycle for Emerging Technologies. Obtenido de https://www.gartner.com/technology/media-products/reprints/ntt_com/340159.html?mkt_tok=eyJpIjoiWlRFM05tSTVNelE1WVRVMSIsInQi OiJQaWZFZ1dEVUVkNWpmWHBtRjNYb3h4bWFqOFhRUnI2WkZjY3JYZ0

Gil, D. Á. (14 de 01 de 2021). *Metodología CRISP-DM.* Obtenido de https://www.adictosaltrabajo.com/2021/01/14/metodologia-crisp-dm/

González, R. (2018). Raspberry pi como plataforma de algoritmos de Machine Learning: reconocimiento de imágenes y datos financieros en streaming.

Guinovart, J. (1998). Fundamentos de Lingüística Computacional: bases teóricas, líneas de investigación y aplicaciones.

Heras, J. M. (10 de 09 de 2020). *Precision, Recall, F1, Accuracy en clasificación*. Obtenido de https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/

IBM. (18 de 08 de 2021). *CRISP-DM en IBM SPSS Modeler*. Obtenido de https://acortar.link/kZ9yl4

José, M. (21 de 09 de 2015). *La metodología de la investigación*. Obtenido de https://www.gestiopolis.com/la-metodologia-de-la-investigacion/

Luther, M. (16 de 01 de 2020). *4 Metodologías para proyectos de Data Science – INVESTIGACIÓN DATLAS.* . Obtenido de https://acortar.link/vrDVok

Martínez, C. G. (5 de 2018). *REGRESIÓN LOGÍSTICA (Simple y Múltiple)*. Obtenido de https://rstudio-pubs-static.s3.amazonaws.com/388799_ac1988bada0143d4a4cef0847e3605f8.html#regresi%C3%B3n_log%C3%ADstica_m%C3%BAltiple

Martinez, J. (10 de 10 de 2020). *Librerías de Python para Machine Learning*. Obtenido de 10: https://www.iartificial.net/librerias-de-python-para-machine-learning/

McCarthy, J. (2007). What is artificial intelligence.

Medina, B. y. (2015). Sistema Basado En La Detección Y Notificación De Somnolencia En Conductores De Autos.

Ng, A. (2019). How to Choose Your First AI Project. Harvard Business Review. Obtenido de https://hbr.org/2019/02/how-to-choose-your-first-ai-project

Pedrero, V. (1 de 2021). Generalidades del Machine Learning y su aplicación en la gestión sanitaria en Servicios de Urgencia. Obtenido de https://www.scielo.cl/scielo.php?pid=S0034-98872021000200248&script=sci_arttext

Qigang, B. (2012). *Logo Recognition Base on a Novel Pairwise Classification Approach. 2012 16th CSI Interational Symposium on Artificial Intelligence and Signal Processing (AISP) (págs. 316-321). Shiraz, Fars: IEEE.*

Rollins, I. J. (06 de 2015). *Metodología Fundamental* . Obtenido de https://www.ibm.com/downloads/cas/6RZMKDN8

Romero, J. (2007). Inteligencia Artificial y Computación Avanzada. (F. A. Brañas, Ed.). .

Ronquillo, M. y. (2020). Diseño de un prototipo para sistema de monitoreo del nivel de llenado en contenedores de basura por protocolo de comunicación inalámbrica IEEE 802.15.4 (ZIGBEE).

Saini, A. (29 de 08 de 2021). *Algoritmo de árbol de decisión: una guía completa*. Obtenido de https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/

Santiago. (02 de 2021). *Generalidades del Machine Learning y su aplicación en la gestión sanitaria en Servicios de Urgencia*. Obtenido de https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0034-98872021000200248

School, T. (11 de 10 de 2022). *¿Qué es Python? Te contamos todo sobre este popular lenguaje*. Obtenido de https://www.tokioschool.com/noticias/que-es-python/

Schwab, K. C. (2016). The Fourth Industrial Revolution: what it means, how to respond.

Tegmark, M. (2017). Vida 3.0.

unesco. (14 de 03 de 2023). *Recomendación sobre la ética de la inteligencia artificial*. Obtenido de https://eduteka.icesi.edu.co/articulos/unesco-etica-de-la-inteligencia-artificial

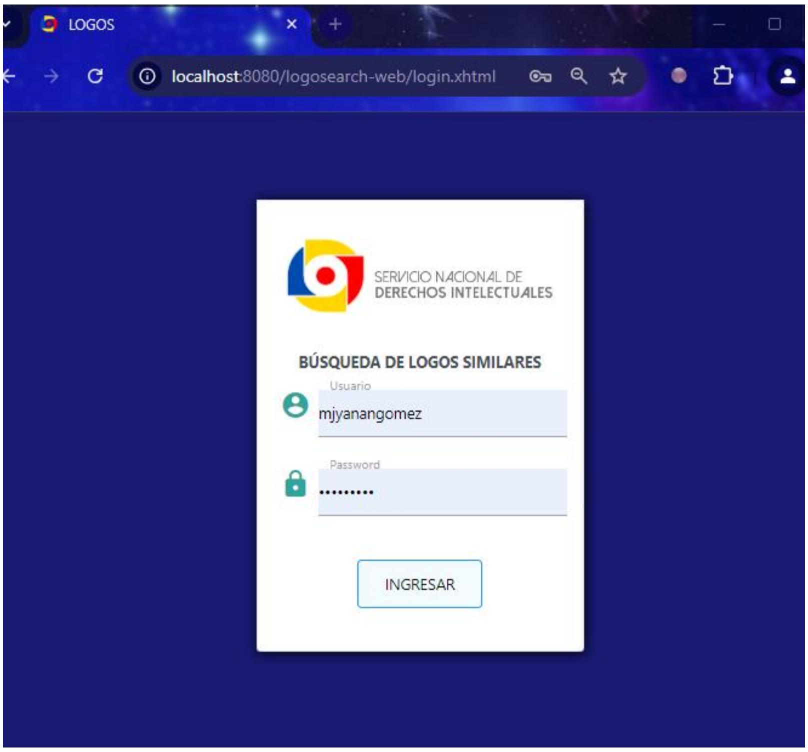
Uniteco. (6 de 4 de 2022). *EL MACHINE LEARNING EN MEDICINA, EL FUTURO DE LA PROFESIÓN*. Obtenido de https://acortar.link/8vdXYz

Weiss, D. (1993). *Logo Recognition Using Geometric Invariants . Proceedings of the Second International Conference on Document Analysis and Recognition, 1993. (págs. 894-897). Tsukuba: IEEE.*

ANEXOS

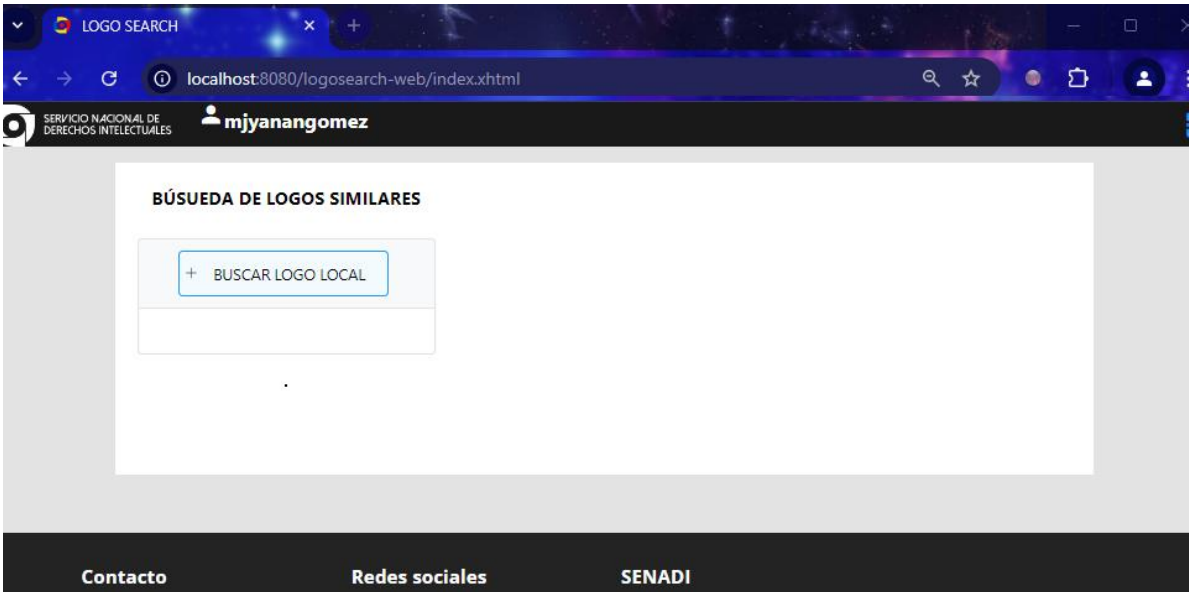
Anexo 1. *Aplicación Web desarrollada*

Figura 30. *Login de Usuario*



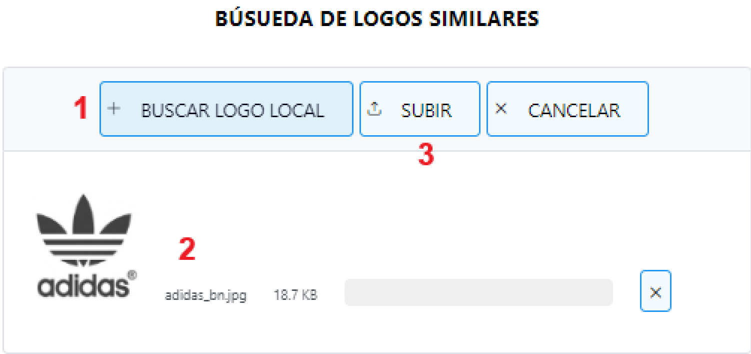
Fuente: *Yanangómez (2024)*

Figura 31. *Pantalla Principal*



Fuente: *Yanangómez (2024)*

Figura 32. Carga de imagen local

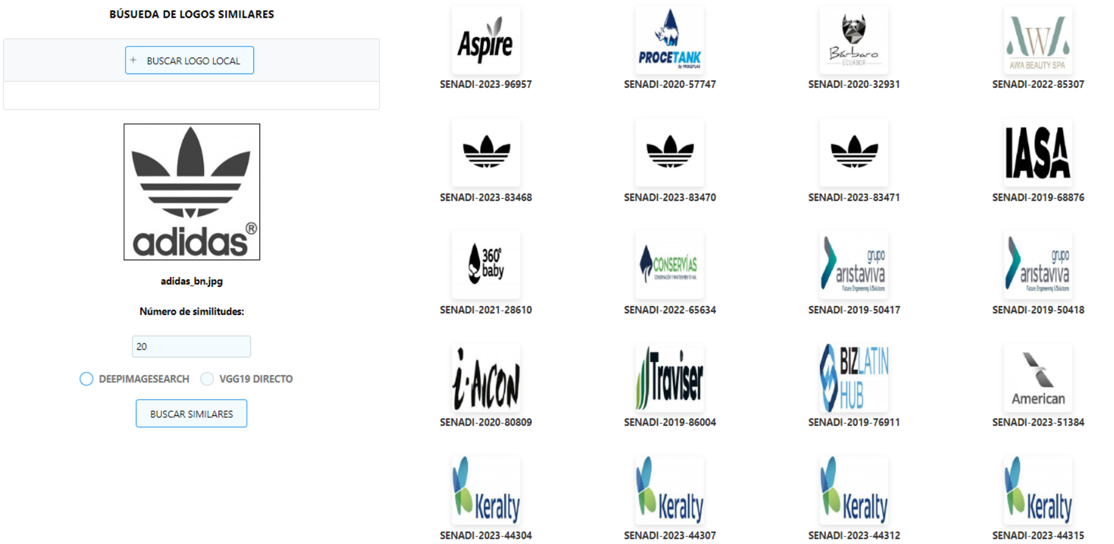


Fuente: Yanangómez (2024)

Figura 33. Imagen local cargada

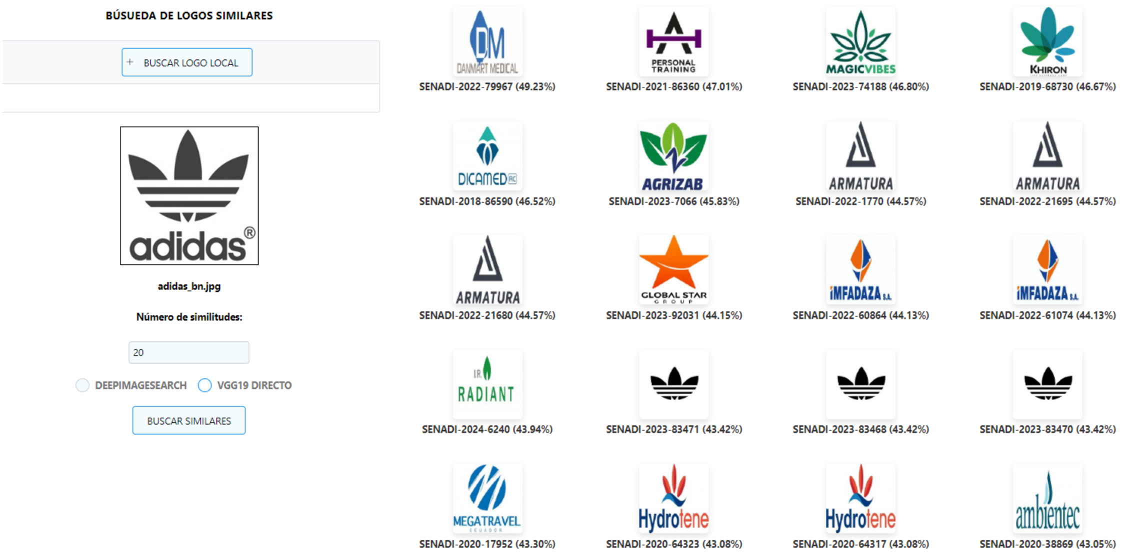
Fuente: Yanangómez (2024)

__Figura 34.__ Resultados con DeepImageSearch



__Fuente:__ Yanangómez (2024)

__Figura 35.__ Resultados con VGG19



__Fuente:__ Yanangómez (2024)

__Figura 36.__ Información del Signo Distintivo

INFORMACIÓN DEL TRÁMITE SELECCIONADO ×

Trámite:	**SENADI-2023-83468**
Denominación:	**Logo (Trefoil version 5)**
Fecha de Presentación:	**2023-11-01**
Tipo:	**MARCA DE PRODUCTO**
Número Gaceta:	**732**
Estado:	**TITULO VERIFICAR RECURSOS**
Expediente:	**IEPI-01-01-01-2023-01-010114**
Clasificación Internacional:	**Productos para limpieza del calzado.**
Descripción:	**La marca figurativa de productos solicitada consiste en un diseño característico conformado por tres hojas de color negro, ubicadas una al lado de la otra y unidas en su parte inferior, las cuales en su parte inferior son cruzadas por tres barras horizontales en color blanco, las cuales entrecortan las hojas, todo esto se encuentra sobre una base de color blanco, tal como se desprende del arte que se adjunta. El solicitante se reserva el derecho de exclusividad sobre el diseño característico como marca figurativa de productos, pudiendo usarla en distinto color y tamaño, sola o acompañada de cualquier diseño, frase, signo o leyenda.**

CERRAR

__Fuente:__ Yanangómez (2024)

__Anexo 2.__ Evaluaciones de los modelos

Figura 37. *Prueba 5*

Fuente: *Yanangómez (2024)*

Figura 38. *Prueba 10*

Fuente: *Yanangómez (2024)*