



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ciencias Naturales y Matemáticas

“Modelo de reconocimiento de dígitos manuscritos mediante
redes neuronales para formularios físicos”

PROYECTO INTEGRADOR

Previo la obtención del Título de:

INGENIERO EN ESTADÍSTICA INFORMÁTICA

Presentado por:

Emily Adriana Alcívar Tóala

Armando Andrés Avilés Candelario

GUAYAQUIL - ECUADOR

Año: 2022

DEDICATORIA

A Dios por darme la oportunidad y no dejarme sola en el camino.

A mis padres Juan Carlos y Virginia por apoyarme y creer en mí.

Emily Adriana Alcívar Tóala

A Dios por tenerme con salud y gozo para cumplir una meta más.

A mis profesores y tutores, porque sin su formación y guía este trabajo no tendría la calidad esperada.

Armando Andrés Avilés Candelario

AGRADECIMIENTOS

A Dios le agradezco por darme sabiduría, fortaleza y poner a las personas correctas en mi camino.

A mis padres por su guía constante, su apoyo, comprensión y motivación.

A mi tío José que siempre me ayudó con consejos, motivación y siguiendo cada uno de mis pasos.

A toda mi familia que confió en mí.

Emily Adriana Alcívar Tóala

A Dios por haber puesto en mi camino las herramientas necesarias para alcanzar esta meta.

A mi mamá por haberme formado con carácter para no dejarme vencer ante las adversidades durante la carrera.

Finalmente, a mis compañeros de clase por siempre haber tenido las más altas expectativas de este logro.

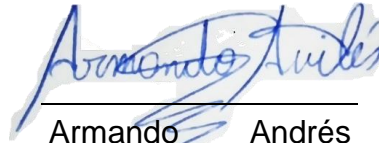
Armando Andrés Avilés Candelario

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, me corresponde conforme al reglamento de propiedad intelectual de la institución; Yo, Emily Adriana Alcívar Tóala y yo Armando Andrés Avilés Candelario, damos nuestro consentimiento para que la ESPOC realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”



Emily Adriana
Alcívar Tóala



Armando Andrés
Avilés Candelario

EVALUADORES

Ph.D. Sandra García Bustos

PROFESOR DE LA MATERIA

M.Sc. Jéssica Menéndez Campos

PROFESOR TUTOR

ÍNDICE GENERAL

Contenido

CAPÍTULO 1	10
1. Introducción	10
1.1 Justificación del problema	11
1.2 Objetivos	12
1.2.1 Objetivo General	12
1.2.2 Objetivos Específicos	13
1.3 Marco Teórico	13
1.3.1 Inteligencia Artificial	13
1.3.2 Aprendizaje automático (<i>Machine learning</i>)	14
1.3.3 Métodos de clasificación	16
1.3.4 Reconocimiento óptico de caracteres	17
1.3.5 Redimensionamiento de una imagen	17
1.3.6 Arreglo bidimensional de píxeles	18
1.3.7 Binarización	19
1.3.7.1 Escala de grises de una imagen	19
1.3.7.2 Binario invertido	20
1.3.7.3 Fragmentación o segmentación de píxeles	20
1.3.8 Adelgazamiento de componentes	21
1.3.9 Promedio de píxeles	21
1.3.10 Aprendizaje profundo (<i>Deep learning</i>)	21
CAPÍTULO 2	27
2. METODOLOGÍA	27
2.1 Creación de la data de entrenamiento, validación y testeo	27
2.1.1 Tratamiento de la imagen	28
2.1.2 Categorización manual	29
2.2 Creación del modelo Redes Neuronales	29

CAPÍTULO 3	31
3. Resultados Y ANÁLISIS.....	31
3.1 Acurracy y Loss	31
3.1.1 Modelo 1 con filtros 64-64-128-256 plus.....	31
3.1.2 Modelo 2 con filtros 50-100-500-1000-10 plus.....	32
3.1.3 Modelo 3 con filtros 64-64-128-256	33
3.1.4 Modelo 4 con filtros 50-100-500-1000-10	35
3.2 Matriz de confusión de las predicciones	36
3.3 Elección del mejor modelo	37
CAPÍTULO 4	39
4. CONCLUSIONES Y RECOMENDACIONES.....	39
4.1 Conclusiones	39
4.2 Recomendaciones.....	39

Lista de Tablas y Figuras

Figura 1: Composición de la Inteligencia Artificial	14
Figura 2: Adelgazamiento de Componentes	21
Figura 3: Operación con Filtro Convolutivo	25
Figura 4: Segmentación de la data de entrenamiento	28
Figura 5: Categorización manual	29
Figura 6: Accuracy del modelo 1 64-64-128-256 plus	32
Figura 7: Loss del modelo 1 64-64-128-256 plus	32
Figura 8: Accuracy del modelo 2 50-100-500-1000-10 plus	33
Figura 9: Loss del modelo 2 50-100-500-1000-10 plus	33
Figura 10: Accuracy del modelo 3 64-64-128-256	34
Figura 11: Loss del modelo 3 64-64-128-256	34
Figura 12: Accuracy del modelo 4 50-100-500-1000-10	35
Figura 13: Loss del modelo 4 50-100-500-1000-10	35
Figura 14: Matriz de confusión modelo 1	36
Figura 15: Matriz de confusión modelo 2	36
Figura 16: Matriz de confusión modelo 3	37
Figura 17: Matriz de confusión modelo 4	37
Figura 18: Predicciones del modelo	38
Tabla 1: Exactitud y precisión de los 4 modelos	38

Resumen

Los sistemas de Reconocimiento Óptico de Caracteres (ROC) muestran resultados prometedores para convertir un texto impreso o escrito a mano en uno con formato ASCII o UNICODE, con el fin de que la digitación de documentos resulte lo menos operativa posible. En esta tesis proponemos un modelo de reconocimiento de dígitos manuscritos mediante redes neuronales para formularios físicos, que sea capaz de almacenar, en formato digital, los resultados que se llenan a mano. Primero, damos una revisión de la situación que aborda esta problemática, así como los conceptos claves que se abordarán. Segundo, tratamos a detalle los procedimientos y la implementación de las redes neuronales, desde que se extrae la información de las imágenes, hasta que los modelos son capaces de dar su predicción. Tercero, damos un análisis de los modelos obtenidos, determinando que aquel con la configuración 50-100-500-1000-10 cumple mejor con los objetivos del proyecto. Finalmente, con base en los resultados, sacamos algunas conclusiones y destacamos los más relevantes beneficios que se reflejan en la práctica.

ABSTRACT

Optical Character Recognition (OCR) systems show promising results for converting printed or handwritten text into ASCII or UNICODE format, in order to make document typing as painless as possible. In this thesis we propose a handwritten digit recognition model using neural networks for physical forms, which is able to store, in digital format, the results that are filled in by hand. First, we give a review of the situation that addresses this problem, as well as the key concepts that will be addressed. Second, we discuss in detail the procedures and implementation of neural networks, from the time the information is extracted from the images, until the models are able to give their prediction. Third, we give an analysis of the models obtained, determining that the one with the 50-100-500-1000-10 configuration best meets the objectives of the project. Finally, based on the results, we draw some conclusions and highlight the most relevant benefits that are reflected in practice.

CAPÍTULO 1

1. INTRODUCCIÓN

La inteligencia artificial (IA) en la actualidad es tendencia mundial y no porque sea algo revelado recientemente. El nacimiento de este término data del año 1956 por John McCarthy. Pero hasta ahora es en donde se ha descubierto su gran utilidad, por medio del aprendizaje profundo (*Deep learning*) y los avances que ha presentado, ha solucionado problemáticas que resultaban casi imposible con los métodos tradicionales en sistemas de computación. Gracias a esto se han desarrollado aplicaciones a través del entrenamiento de las redes neuronales para automatizar tareas complejas. (Abeliuk & Gutiérrez, 2021).

Todos estos resultados los podemos ver reflejados en varios aplicativos del mundo actual que usan el reconocimiento de rostros a través de imágenes, incluso el reconocimiento de imágenes que usa Google para su buscador, son ejemplos de lo que nos permite realizar los sistemas de IA y el gran número de aplicativos que se pueden hacer a través del entrenamiento de redes neuronales artificiales (Durán, 2017).

Uno de los tipos de redes neuronales más utilizados son las convolucionales que son usadas con mayor frecuencia para resolver problemas de clasificación, localización e identificación de objetos dentro de imágenes, por su efectividad en el campo de la visión artificial. Es decir que de entrada recibe una imagen y como salida me proporciona una clasificación (Bonilla, 2020).

El reconocimiento de caracteres manuscritos para los seres humanos es una tarea que se ha desarrollado desde niños cuando les enseñan a leer y escribir (Flores, 2006). Resulta algo sencillo hasta que su finalidad es la digitalización masiva de información, la cual podemos optimizar al

automatizar esta operación, a través de reconocimiento de dígitos con imágenes de los documentos físicos. En otras palabras, mediante un modelo de redes neuronales que permita ingresar una imagen y me devuelva un archivo de texto con la información que el documento posee.

Los formularios son un instrumento para recolectar información, que en ocasiones forman parte de procesos comerciales que demandan mucho tiempo y administración, ya que se presentan grandes volúmenes de datos. El procesamiento de los formularios estadísticos provenientes de los registros administrativos de Instituciones Públicas, como aquellos que el INEC se encarga de realizar para recolectar y producir la información estadística del país, requiere de 2 pasos previos a su ubicación en una base de datos: crítica y codificación, y digitación.

El Instituto Nacional de Estadística y Censo (INEC) que se encarga de recolectar y producir toda la información estadística de un país, presenta una necesidad tan grande en el procesamiento de los datos. Usualmente para recolectar datos se hace el uso de formularios físicos, que resultan muy útiles, pero algo complejos al momento de digitalizar los datos. Es un desafío muy grande ya que requiere intervención manual, por ende, resulta un proceso lento y vulnerable a errores.

En el presente estudio se construye un conjunto de datos de entrada para el entrenamiento de un modelo de aprendizaje automático que permite el reconocimiento de dígitos a partir de imágenes. El conjunto de datos se creará a partir de caracteres manuscritos registrados en formularios físicos de los registros administrativos. Y tiene como finalidad optimizar el procesamiento de información de los formularios físicos.

1.1 Justificación del problema

En una época globalizada y de producción digital de escala masiva, hay procesos que se realizan de forma manual como llenar un formulario de datos, documentos comerciales y financieros (cheques, pagarés), o

completar una ficha técnica con cierta periodicidad. Mohammad, Musa y Ali (2022) comparten la idea de que, en ocasiones, se requiere extraer los datos de estos documentos con el fin de almacenarlos en bases de datos corporativas o de carácter oficial, para que otros usuarios le den el mejor de los usos.

La tarea de la digitación resulta cada vez más operativa, con el crecimiento del volumen de los datos que nuestro medio lo exige. Por tal motivo, el trabajo de reconocer los caracteres, tanto alfabéticos (nombre, dirección, tipología) como numéricos (teléfono, fechas, opciones) puede ser reemplazado por un programa entrenado que lo realice de manera automática sin el esfuerzo que le toma a una persona experta en el tema. De esta manera, Gupta y Jalal (2022) indican que las metodologías de Aprendizaje Profundo (*Deep Learning*) usa técnicas de Redes Neuronales Artificiales para extraer las características del texto de forma automática.

Un mismo formulario puede estar compuesto por varias preguntas y dividido en más de una sección que, con el tiempo, llevan a reducir la efectividad de la digitación por parte del encargado. Caso contrario, el algoritmo de reconocimiento aprende y mejora su función con cada documento que revisa, por lo que su eficiencia mejora. Por otra parte, Singh, Garg y Kumar (2022) respaldan la idea de un sistema de reconocimiento de caracteres escritos a mano, aunque argumentan que es una tarea retadora, debido a los variados estilos de escritura de los individuos.

1.2 Objetivos

1.2.1 Objetivo General

Construir y entrenar un modelo de reconocimiento de dígitos manuscritos, por medio de redes neuronales convolucionales, para ser aplicado en formularios de llenado físicos.

1.2.2 Objetivos Específicos

- Clasificar las imágenes de los registros administrativos para estandarizar adecuadamente los datos de entrenamiento y testeo según el método a utilizar.
- Probar el modelo con el conjunto de datos de prueba para poder realizar una mejor clasificación.
- Comparar distintos modelos de clasificación para determinar el que posee mejores resultados.

1.3 Marco Teórico

1.3.1 Inteligencia Artificial

La inteligencia artificial (IA) es un término que nace en el año 1956 por John McCarthy, junto con el primer programa de IA por Allen Newell, Herbert Simon y Cliff Shaw denominado *Logic Theorist*. Desde entonces se han dado múltiples aportes. En 1964 se presentó el primer *chatbot* llamado Eliza y su principal función era conversar con una persona en inglés, tiempo después en 1997 se dio a conocer *Deep blue*, un programa que logró derrotar al campeón mundial de ajedrez. Asimismo, se han desarrollado asistentes virtuales reconocidos como Alexa, lanzada por Amazon en 2014. (Abeliuk & Gutiérrez, 2021).

Lasse Rouhiainen (2018) plantea que la inteligencia artificial es un tema complejo y difícil de definir, ya que se encuentran diferentes definiciones sobre el tema. Sin embargo, él la conceptualiza como “la capacidad de las máquinas para usar algoritmos, aprender de los datos y utilizar lo aprendido en la toma de decisiones tal y como lo haría un ser humano” (p.17).

De manera similar lo define Carmelo Bonilla (2020), como un proceso en donde se automatizan tareas que usualmente son realizadas por los seres humanos.

Actualmente se usan estas tecnologías en beneficio de los humanos para lograr mayor eficiencia en casi todos los ámbitos o situaciones en donde se pueda aplicar la IA (Lasse Rouhiainen, 2018). Entre ellas tenemos las más usadas en la actualidad:

- Reconocimiento de imágenes
- Detectar y clasificar objetos
- Metodologías en prevención de riesgos
- Algoritmos de recomendación

Además de ser un área que tiene adherido el aprendizaje automático (*Machine Learning*) y aprendizaje profundo (*Deep Learning*) (Paula Uribe, 2020).

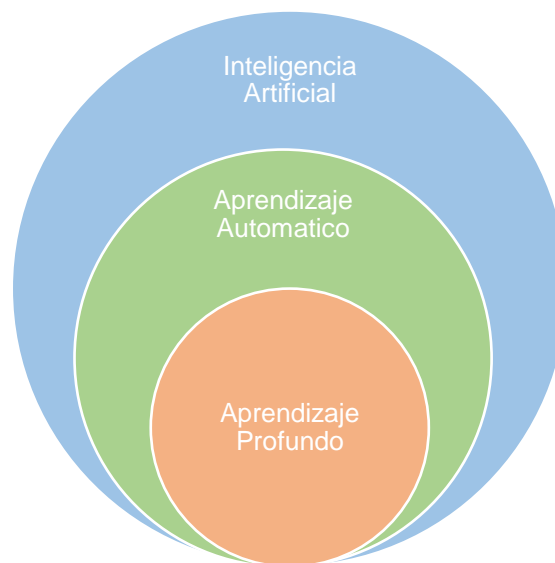


Figura 1: Composición de la Inteligencia Artificial

1.3.2 Aprendizaje automático (*Machine learning*)

Considerado como componente principal de la IA, que mediante algoritmos y el procesamiento de los datos examina posibles patrones. *Machine Learning* (ML) utiliza algoritmos matemáticos en donde el ordenador imita la forma en que los humanos aprendemos (Martínez Llamas, J. 2018). En la Figura 1; **Error! No se encuentra el origen de l**

a referencia. se ilustra la composición de la Inteligencia Artificial, así como el nivel de profundización de cada componente en algoritmos de aprendizaje.

Lasse Rouhiainen (2018) lo define como una rama de la informática donde los ordenadores tienen la capacidad de aprender, sin antes ser programados. Dando como resultado predicciones o sugerencias según sea el caso, ya que logra aprender patrones mediante los datos. Son usados para detectar spam de correos electrónicos, este algoritmo una vez que logra identificar correos basura los separa de los que contienen información importante.

Tal como sostiene Bonilla (2020) el aprendizaje automático es una ciencia que posibilita acciones al ordenador sin previa programación, además asegura que para perfeccionar los algoritmos es importante que sean entrenados con una gran cantidad de datos y logren mayor precisión. Ya que el algoritmo aprende con los ejemplos dados, de esta manera podemos notar si está acertando o cometiendo errores.

Existen varios tipos de aprendizaje automático entre ellos: Aprendizaje Supervisado, Aprendizaje No Supervisado y Aprendizaje de Refuerzo.

1.3.2.1 Aprendizaje Supervisado

Rodríguez (2018) afirma que un aprendizaje es supervisado cuando los datos que usamos para entrenar el algoritmo han sido etiquetados, es decir han sido organizados previamente para enseñarle cómo categorizar la información nueva.

Lasse Rouhiainen (2018) lo establece como un método que requiere la intervención humana para dar retroalimentación. Por ejemplo, ingresar al algoritmo imágenes de gatos para que posteriormente identifique imágenes similares.

Para el aprendizaje requerimos de un conjunto de datos numerosos, para así poder destinar una parte de ellos al entrenamiento del ordenador, otro a la prueba y de ser el caso a la validación del

modelo. De esta manera podemos medir su precisión, es decir qué tanto está acertando en base a lo aprendido. Existen 2 tipos de problemas en el aprendizaje automático: Clasificación y Regresión según Rodríguez (2018).

1.3.3 Métodos de clasificación

Se caracterizan porque la variable a predecir es de tipo discreta o contiene categorías. Pueden ser de tipo binaria, múltiple u ordenada. La finalidad de este algoritmo es indicarnos a qué grupo o categoría pertenece nuestro objeto de estudio. Es decir, mediante los datos, busca patrones y los agrupa, posteriormente los compara con nuevos ingresos asignándole uno de los grupos (Sandoval, 2018).

Existen múltiples algoritmos o métodos de clasificación, en este caso veremos *Random forest*, KNN (*K- Nearest neighbors*)

1.3.3.1 Random Forest

Pechuán et. al. (2015) lo describe como un algoritmo que se crea por conjuntos de varios modelos poco eficientes como los árboles de decisión, es decir se entrena cada uno de los modelos usando diferentes conjuntos de entrenamiento. Se extrae un número fijo de muestras de los datos de entrenamiento, y con cada una de estas se construye un modelo. De la misma manera se selecciona un conjunto de variables de interés para cada muestra, se hacen predicciones de cada modelo y se elige la predicción final por votos de la mayoría (la predicción que más se repita).

Sandoval (2018) enfatiza que los árboles de decisión y *Random Forest* son modelos con mayor precisión, estabilidad y facilidad en la interpretación. Utilizan reglas de decisión figuradas como un árbol,

lo que es una ventaja porque los convierte en modelos más elaborados, sin embargo, pueden llegar a perder rendimiento.

1.3.3.2 KNN

Este algoritmo de clasificación como su nombre lo indica (K vecinos cercanos), se fundamenta en el cálculo de las distancias más cercanas, entre el conjunto de entrenamiento y un nuevo ingreso. K es el número de vecinos (clases, categorías, conjunto) que tienen mayor similitud al nuevo ingreso, al cual se le asigna la que tenga mayor aparición entre los diferentes K (Pechuán et al., 2015).

1.3.4 Reconocimiento óptico de caracteres

Raza, et. al (2022) explica que el sistema de reconocimiento óptico de caracteres convierte un texto impreso o escrito a mano en un texto con formato digital como ASCII o UNICODE. Dicho sistema existe debido a la necesidad de un formato computarizado y entendible con el fin de la extracción del conocimiento.

Generalmente, estas técnicas tienen una menor efectividad para el reconocimiento en texto escrito a mano en comparación con el impreso. Por tal motivo, un mayor número de autores en los últimos años se han enfocado en los diversos componentes del reconocimiento de caracteres, tales como redimensionar una imagen, el arreglo bidimensional de píxeles, binarización, la segmentación de la región y el adelgazamiento de componentes.

1.3.5 Redimensionamiento de una imagen

Para Talebi y Milanfar (2021), el reconocimiento de un objeto resultará en un proceso más eficiente si la imagen se redimensiona a una resolución espacial considerablemente menor. Además, modificar la escala de un objeto permite manipular y adecuar sus características

físicas a un determinado modelo o hacer correcciones necesarias de escala, de ser necesario. Por otro lado, es importante tener en cuenta que el redimensionamiento no mejora las cualidades visuales de la imagen, sino que se usa para conseguir datos de entrada más eficientes, tanto para entrenar como para evaluar un modelo que produzcan resultados más precisos.

De igual manera, las limitaciones de memoria exigen el redimensionamiento de las imágenes, pues modelos complejos como las Redes Neuronales Convolucionales no pueden ser entrenados con objetos de alta resolución en la mayoría de los ordenadores. Asimismo, cambiar las dimensiones resulta en un método de estandarización para un conjunto de imágenes no homogéneas en tamaño.

1.3.6 Arreglo bidimensional de píxeles

Koloda et. al. (2022) nos explica que las imágenes digitales son representadas en arreglos bidimensionales, donde cada píxel está ubicado en una posición entera. En ocasiones, la recolección de la muestra resulta en posiciones no enteras de los píxeles, lo cual no permite obtener una buena visualización del objeto, pero tampoco tener unos adecuados datos de entrada para el procesamiento de la imagen.

Además de estandarizar la escala de los objetos, en ocasiones el conjunto de imágenes también es heterogéneo en forma y se requiere de una manipulación en el rango sobre el que se extienden los píxeles de esta. En otras palabras, el dominio de la imagen no es fijo. Por ejemplo, si se obtiene un documento escaneado o fotografiado con una ligera distorsión (en forma de trapecio), lo más probable es que el modelo arroje un error por la dimensionalidad de la imagen. Para esto, Bogaerts, et. al (2020), explica que un arreglo bidimensional relaciona la necesidad de baja resolución y la de un intervalo adecuado de los píxeles.

1.3.7 Binarización

La binarización es un proceso de transformación de datos en un vector binario (ceros y unos) que se usa con el fin de mejorar la eficiencia de algoritmos de clasificación. Las imágenes, los sonidos o textos son conceptos complejos de procesar, por lo que su transformación en forma binaria resulta de gran ayuda para la lectura de data y optimización de recursos computacionales. Esta transformación se realiza de manera que mide la presencia o ausencia de color (0 para negro, 1 o 255 para blanco).

Shi, et. al. (2022) nos explican que la binarización contribuye a una versión de una Red Neuronal Convolutiva, en la que las características y ponderaciones están representadas por tan solo 1 bit. Por tal motivo, este proceso mejora la eficiencia de estos modelos, pues se reduce el número de parámetros, el uso de memoria y la cantidad de operaciones realizadas. En aplicativos para dispositivos móviles, por ejemplo, los modelos que reducen el tiempo de ejecución son altamente utilizados debido a la experiencia de usuario que ofrecen, lo cual resulta en una ventaja y una opción para considerar.

Por otra parte, Shang, et. al. (2022) también nos comenta que a costas de la eficiencia que ofrece la binarización, hay una reducción en la precisión del modelo, pues se puede ver como una pérdida de información al condensar un conjunto de características y ponderaciones en un solo bit de memoria. Existen métodos para reducir el error que no se mitiga por la binarización de las ponderaciones, pero resulta difícil mitigar el error por la binarización de las características de los píxeles.

1.3.7.1 Escala de grises de una imagen

Respecto a la optimización de recursos, Magro (2013) nos comenta que la gama de colores es otro factor que se considera para procesar una imagen, sin importar el fin que esta tenga. Tanto para transmisión, descarga, almacenamiento y visualización, como para el procesamiento de sus características propias. La visión artificial, la segmentación y el reconocimiento óptico de caracteres son sistemas particulares en los

que se acorta el tiempo de procesamiento y se simplifican las operaciones realizadas internamente.

Por tal motivo, se prefiere trabajar con un filtro gráfico sobre imágenes en escala de grises, donde se defina un método que encuentre un umbral (T) óptimo a partir del cual se determine si un píxel toma el valor de 255 o de 0.

1.3.7.2 Binario invertido

Por otra parte, Molina (2018) nos indica que en ocasiones se presentan limitantes para una correcta binarización, como la poca o no uniforme iluminación de la imagen. En este tipo de casos, es preferible trabajar con una inversión y sustitución de píxeles, que dan como resultado el negativo de la imagen original. En píxeles donde la binarización arrojaría un valor de 0, le asigna 255 y viceversa.

De manera similar, se requiere determinar un umbral para asignar valores a los píxeles, ya sea por similitud, agrupación, distribución valores mínimos y máximos o mera manualidad.

1.3.7.3 Fragmentación o segmentación de píxeles

Esta es una técnica que consiste en dividir la matriz de píxeles en segmentos según las características comunes de los píxeles dentro de una misma región que no se superponga a las demás. De esta manera, se encuentran bordes o trazados de una figura, permitiendo identificar un objeto en una determinada sección de la imagen.

Sungho (2022) nos explica que el número de regiones es influyente tanto en el tiempo de procesamiento, como en la visibilidad de las fronteras del trazado. Si se divide la imagen en más regiones, se tendrá una mejor visibilidad y capturará más información de la imagen, pero el costo computacional será mayor.

1.3.8 Adelgazamiento de componentes

El adelgazamiento de componentes es un algoritmo que aplica un conjunto de condiciones a los píxeles de una imagen desde los extremos hacia el centro, para eliminar a aquellos que las cumplen, con el fin de obtener la silueta del trazo que se pretende reconocer. Shyla (2022) explica que este procedimiento identifica los bordes del trazado al encontrar píxeles adyacentes, es decir aquellos pares de píxeles vecinos tanto horizontal, vertical y diagonalmente que tienen el mismo nivel de gris (misma luminosidad) y que en su conjunto forman una figura o región de la imagen. En la Figura 2 se ilustra el resultado del adelgazamiento de componentes para definir el trazo que sigue la letra “a” en la imagen.



Figura 2: Adelgazamiento de Componentes

1.3.9 Promedio de píxeles

El promedio de píxeles es un proceso que consiste en promediar los $n \times n$ píxeles vecinos no superpuestos y combinarlos en igual orden para formar una nueva imagen de menor tamaño que la original. Debarnot (2022) indica que al reducir la resolución de la imagen en una proporción igual al cuadrado del número de píxeles vecinos que se tomen, la precisión que se deja de ganar es menor. En otras palabras, el aumento en la eficiencia en el procesamiento de la imagen y la modelización es mayor que la pérdida de precisión como producto de una menor resolución.

1.3.10 Aprendizaje profundo (*Deep learning*)

Chollet (2021) nos enseña que el campo del aprendizaje profundo se enfoca en la representación de la data de manera que esta pueda brindar

conclusiones, mediante el entendimiento de la jerarquía de conceptos que se puede extraer del pozo de datos. De esta manera, muchos campos con un rol clave en el desarrollo de esta tecnología tienen avances importantes tales como la infraestructura de la data con el fin de coleccionar, almacenar y manipular grandes cantidades de datos. Así mismo, los procesos de optimización logran progresos tanto en tiempo de ejecución como en precisión, particularmente aquellos en los que interactúan *software* con *hardware*.

En parte, el éxito del conjunto de algoritmos de aprendizaje profundo radica en que las estructuras computacionales trabajan con múltiples e iterativas transformaciones no lineales de datos, no solo en forma matricial, sino también tensorial, lo que permite capturar la mayor cantidad de características de los registros y ofrecer interpretaciones más precisas. Es así como el complejo proceso de extracción de datos resulta considerablemente más sencillo de realizar según la estructura que tengan o que se les den a los mismos.

1.3.10.1 Redes Neuronales

Las distintas aplicaciones del aprendizaje estadístico concentran sus esfuerzos en automatizar procesos de clasificación y predicción en distintos campos y diversos formatos de datos de entrada (imagen, sonido, texto, etc.). Entre las técnicas más conocidas e investigadas están las Redes Neuronales que, como nos explica Künkel (2021), son una técnica computacional conformada por un conjunto (o conjuntos según su complejidad) de neuronas artificiales (unidades) conectadas entre capas que realizan un determinado proceso con datos de entrada de una unidad anterior para entregar datos de salida a unidades siguientes. Cada neurona lleva un proceso de aprendizaje continuo que permite mejorar la precisión de todo el modelo como un solo conjunto.

IBM Corporation (2021) comenta que, en un inicio, las unidades entre capas se conectan por medio de ponderaciones de manera aleatoria, pero al comparar los resultados obtenidos con los conocidos, las

ponderaciones cambian gradualmente hasta cumplir el/los criterios de parada y aumentar la precisión del modelo.

A pesar de su gran aplicabilidad para la clasificación y predicción, esta técnica carece de interpretabilidad, pues la forma de las relaciones entre las variables de explicación y a ser explicadas se determinan durante el proceso de aprendizaje de manera que los datos se ajusten lo más posible sin la necesidad de supuestos sobre parámetros.

1.3.10.1.1 Convolución

Un concepto fundamental para la detección y reconocimiento de objetos es la convolución. Gatica (2021) detalla que es un operador matemático que resulta en una función simétrica a dos funciones de entrada, de manera que representa la medida en la que se superponen la primera función con una forma invertida y trasladada de la segunda. Matemáticamente, la convolución de dos funciones continuas f y g , se representa de la siguiente manera:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(T)g(t - T)dT$$

Análogamente, se emplea la siguiente forma convolucional para funciones discretas:

$$f[p] * g[p] = \sum_q f[q]g[p - q]$$

Dentro del contexto del reconocimiento de objetos, los distintos problemas de enfoque encuentran la solución en una función de convolución de la imagen original y el efecto de la distorsión generada.

1.3.10.1.2 Capa de entrada

Wu (2017) nos explica la arquitectura de una red neuronal compuesta por una capa de entrada, una capa de salida y, de existir, una o varias capas ocultas. La capa de entrada está conformada por nodos que

reciben los datos de las variables de explicación y los transmite a los nodos de la siguiente capa.

1.3.10.1.3 Capa de salida

La capa de salida es el último conjunto de nodos de la red neuronal y contiene los resultados finales de todos los procesos intermedios. Estos resultados se usarán para medir la efectividad del modelo por medio de las métricas de error.

1.3.10.1.4 Capas ocultas

Por otro lado, las capas ocultas contienen los nodos con los resultados de los diversos subprocesos entre la capa de entrada y la de salida.

Las redes neuronales de una sola capa, es decir, conformadas solo por las capas de entrada y salida, dieron paso al desarrollo de redes de múltiples capas. Aquellas estructuras con una capa oculta llevan el nombre de redes poco profundas, mientras que las compuestas por dos o más capas intermedias son conocidas como redes neuronales profundas.

Dicha clasificación se realiza debido a que el desarrollo de cada tipo de red neuronal marca un considerable avance en sus aplicaciones.

1.3.10.1.5 Filtro o Kernel

Como se ha comentado en diversas ocasiones, en el proceso de reconocer un objeto se presentan situaciones que requieren difuminar o dar nitidez a una imagen, detectar los bordes del texto, deformar el objeto, o alguna otra transformación para ajustar la imagen a los requerimientos del modelo. Kim (2017) nos explica que estos filtros que se realizan a nivel matricial se denominan Kernel, matriz de convolución o máscara. Consiste en una matriz bidimensional pequeña (hasta 5x5) que transforma a la matriz de píxeles de la imagen original al filtro requerido.

En la Figura 3 el autor nos muestra un ejemplo para dos filtros de convolución con una matriz de píxeles de 4x4.

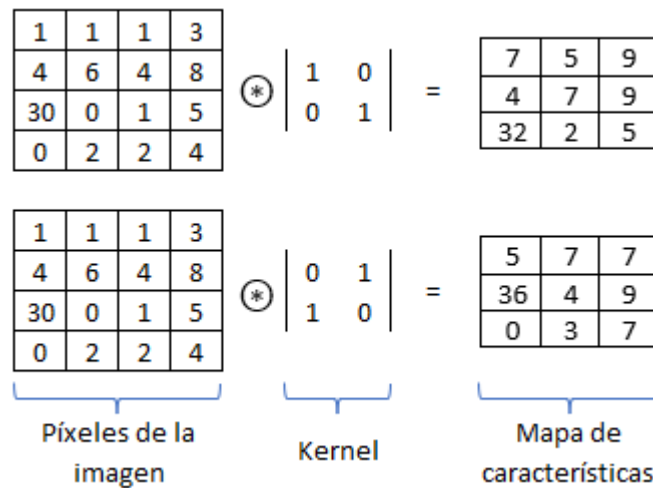


Figura 3: Operación con Filtro Convolutivo

1.3.10.1.6 Redes Neuronales Convolucionales

Li (2021) explica que, a diferencia de los métodos tradicionales, una red neuronal convolutiva extrae las características de los datos con estructuras convolucionales, con base en la percepción visual. Entre las diversas capas de la red se emplean filtros o Kernels que captan las características de la imagen y la pasan al siguiente conjunto de neuronas según una función de activación y un umbral, tal como las señales eléctricas neuronales

1.3.10.2 Data MNIST

Baldomins, et. al (2019) nos explica que MNIST (*Modified National Institute of Standards and Technology*) es un conjunto de datos de dígitos escritos a mano con sus respectivas etiquetas, ampliamente utilizado para facilitar el modelamiento de nuevos conjuntos de datos de similar procedencia, así como con fines didácticos sobre modelos ya entrenados. Dicha base contiene un total de 60000 imágenes de entrenamiento y 10000 de testeo en un formato fácilmente interpretable, con etiquetas del 0 al 9 y una cuadrilla de 28x28 píxeles con valores entre 0 y 255 que corresponden a su luminosidad.

Rosyda & Purboyo (2018) presentan en su trabajo los obstáculos más comunes en el reconocimiento de imágenes y, en particular, en el reconocimiento de caracteres manuscritos. Por un lado, menciona el ángulo de visión, las condiciones de luminosidad, la claridad de la imagen, así como los diversos estilos de escritura, como algunos de los factores que afectan el proceso de reconocer un texto en comparación con uno escrito a computadora. De igual manera, muestra una serie de métodos que, de manera secuencial, resuelven cada uno de estos problemas.

CAPÍTULO 2

2. METODOLOGÍA

En este capítulo se detalla el procedimiento para generar los datos de entrenamiento, testeo y validación complementarios. Se obtuvieron de la extracción de imágenes donde se encontraban manuscritos de dígitos en cada formulario, para luego ser llevados a arreglos bidimensionales de píxeles y almacenados en archivos de texto.

Posteriormente, estos archivos se utilizaron en la construcción de los modelos de clasificación, y su finalidad es predecir qué dígito es el que se encuentra en una sección específica del formulario.

2.1 Creación de la data de entrenamiento, validación y testeo.

Para la elaboración de la base de datos entrenamiento y validación se tomaron formularios llenos y escaneados, de los meses de enero y febrero del 2022 provenientes del Registro Civil, equivalentes a 19.078 registros, pero para el testeo se tomó los 1.254 registros del mes de marzo del 2022. De dichos formularios se consideraron las 3 secciones donde aparecían los dígitos manuscritos, que a su vez se dividían según la cantidad de dígitos presentes en cada sección. Este proceso se lo conoce como segmentación o fragmentación y es vital realizarlo para reconocer cada uno de los dígitos presentes, las zonas en blanco, separación entre dígitos, su límite superior e inferior, etc. Todo esto es posible si se los tiene localizados dentro del documento (coordenadas).



Figura 4:Segmentación de la data de entrenamiento

2.1.1 Tratamiento de la imagen

Una vez extraídos los objetos (imágenes) se le realiza un tratamiento a la imagen con la finalidad de mejorar su calidad y descartar aquellos poco útiles.

Primero le realizamos una binarización que, como se indicó en el marco teórico, es un proceso de transformación en donde una imagen que inicialmente tiene varios colores se convierte a blanco y negro. Es decir, en escala de grises para facilitar el procesamiento y reconocimiento: asigna blanco donde exista ausencia de color y negro donde se encuentre color, esto se representa en una escala de 0 a 255 (0 negro y 1-255 blanco). Seguido de esto se trata de homogenizar el tamaño de las imágenes, ya que en su mayoría tenían tamaños variados. Por ese motivo se redimensionó cada imagen a 28x28 píxeles, considerando una resolución similar para todos, lo que nos indica que cada imagen será un arreglo bidimensional con valores de 0 a 255.

Además, se efectúa un adelgazamiento de componentes y suavizado de la imagen con la finalidad de únicamente obtener los contornos de interés sin perder su forma original. Por último, aplicamos el binario invertido para eliminar el ruido, mejorar el procesamiento de las imágenes y enseguida almacenamos.

2.1.2 Categorización manual

Una vez almacenadas todas estas imágenes tratadas, definimos carpetas para cada categoría, es decir del 0 al 9 y procedemos a ubicarlas según corresponda. Después leemos cada imagen contenida en las categorías como un arreglo bidimensional y lo almacenamos como una matriz de datos, la cual tendrá 784 características o atributos y su etiqueta que indica a qué categoría pertenece.

label	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	4
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	1	9	15	5	0
2	0	0	0	0	0	2	28	67	210	255	218	107	20	1
2	0	0	0	0	0	1	1	6	21	25	25	25	25	25
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	1	2
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	4

Figura 5: Categorización manual

2.2 Creación del modelo Redes Neuronales

La red neuronal convolucional toma de entrada los píxeles de las imágenes que almacenamos, en este caso se complementó la data ya existente conocida como MNIST que posee una matriz de píxeles con 60000 filas de dígitos manuscritos de tamaño 28x28. Sin embargo, como parámetro inicial del modelo se cambian todos los datos a un arreglo de 32x32. Esto es un equivalente a utilizar 1024 neuronas que forman la capa de entrada del algoritmo.

Con la entrada de datos lo que hace la red neuronal es asignarles pesos a los elementos de la imagen para distinguirlas unas de otras. Se define el Kernel el cual se considera como un filtro, para extraer las características importantes (bordes, enfoques) o patrones de las imágenes. Elegimos

varias configuraciones para la capa oculta, en otras palabras, un tamaño de kernel y cantidad de filtros diferentes. Una vez realizado el algoritmo se procede a entrenar el modelo con la data MNIST unida con la creada para entrenamiento y validación. Además, se define que el 80% de los 5.016 registros, junto con 14.062 datos de la data MNIST, sean para el entrenamiento y el 20% restante para la validación. Al final nos indica el *accuracy* de la validación del modelo y la precisión de cada una de las categorías, mientras más cerca de 1 sean estos valores nos indica que mejor es el modelo.

Por último, evaluamos el modelo con la data de test que realizamos, esto nos ayuda a identificar cómo funciona el algoritmo con datos que no ha visto antes y ver su rendimiento. Lo visualizamos mediante la matriz de confusión.

CAPÍTULO 3

3. RESULTADOS Y ANÁLISIS

En este capítulo se muestran los resultados de los modelos entrenados con las diferentes configuraciones en la capa oculta y data de entrenamiento. Primero, se muestran las gráficas del accuracy durante el entrenamiento y validación para evaluar el rendimiento del modelo al clasificar, seguido del Loss (perdida) para medir la desviación entre las predicciones y los valores de verdad. Además, se obtiene la matriz de confusión con los datos de testeo y determinamos el mejor modelo. Finalmente se evalúan cada una de las clases del modelo seleccionado.

3.1 Accuracy y Loss

Se graficaron cada uno de los resultados en el entrenamiento de los modelos. En total 4 modelos con configuraciones diferentes en la capa oculta. Inicialmente se probaron dos de los cuatro modelos con la data MNIST unida a la data de entrenamiento creada con los formularios físicos (se los denominó modelos plus). Posteriormente con las mismas configuraciones, pero únicamente con la data de entrenamiento creada se entrenaron los dos siguientes modelos.

3.1.1 Modelo 1 con filtros 64-64-128-256 plus

Este modelo se construyó y entreno con la data MNIST unida a la data creada, es decir con 79083 registros con 1024 características.

Se planteó la configuración para la capa oculta de convoluciones 64-64-128-256 y 18 épocas, que como vemos en la Figura 6 el *accuracy* que toma valores de 0 a 1 aumenta tanto para los datos de entrenamiento como validación lo que nos indica que el modelo no está sobreentrenado. En la Figura 7 vemos que el Loss disminuye para ambos casos indicando que los valores predichos y reales no están muy distantes.

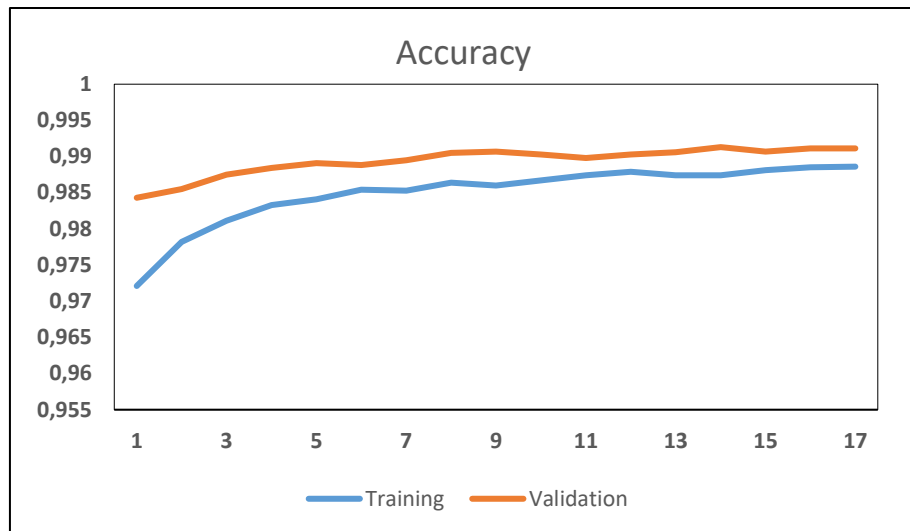


Figura 6: Accuracy del modelo 1 64-64-128-256 plus

Fuente: Elaboración propia



Figura 7: Loss del modelo 1 64-64-128-256 plus

Fuente: Elaboración propia

3.1.2 Modelo 2 con filtros 50-100-500-1000-10 plus

Al igual que el modelo 1 este se construyó con ambas datas, pero con diferentes filtros. Evidentemente se nota un cambio en la gráfica 3.3 donde el *accuracy* entre el entrenamiento y validación tienen un valor similar además mayor al de la Figura 8 para el Modelo 1. Sin embargo, en la Figura 9 vemos que el valor del Loss a pesar de que disminuye, es mayor que el modelo 1 y se encuentra más distante entre el entrenamiento y validación.

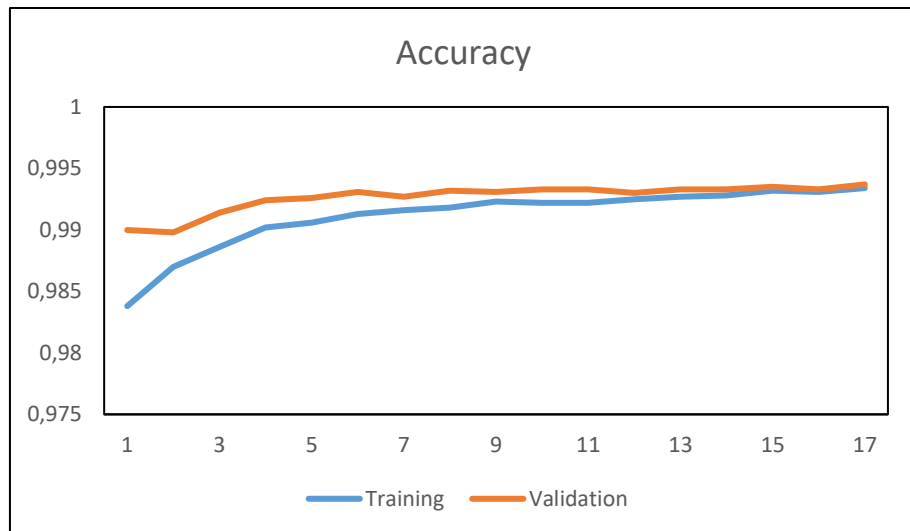


Figura 8: Accuracy del modelo 2 50-100-500-1000-10 plus

Fuente: Elaboración propia



Figura 9: Loss del modelo 2 50-100-500-1000-10 plus

Fuente: Elaboración propia

3.1.3 Modelo 3 con filtros 64-64-128-256

Con la misma configuración en la capa oculta del modelo 1 como referencia, se realizó el entrenamiento del modelo 3 pero únicamente con la data de entrenamiento creada que cuenta con 19083 registros. Además, se incrementó el tamaño de épocas a 35 con la finalidad de mejorar la exactitud del modelo.

Para este modelo vemos en la Figura 10 que posee mejores resultados muy cercanos entre la validación y el entrenamiento, alcanzando casi el valor máximo del accuracy. Pero analizando la Figura 11 notamos que el valor de loss es mínimo y se mantiene a la par en el entrenamiento y validación a lo largo de las épocas definidas.

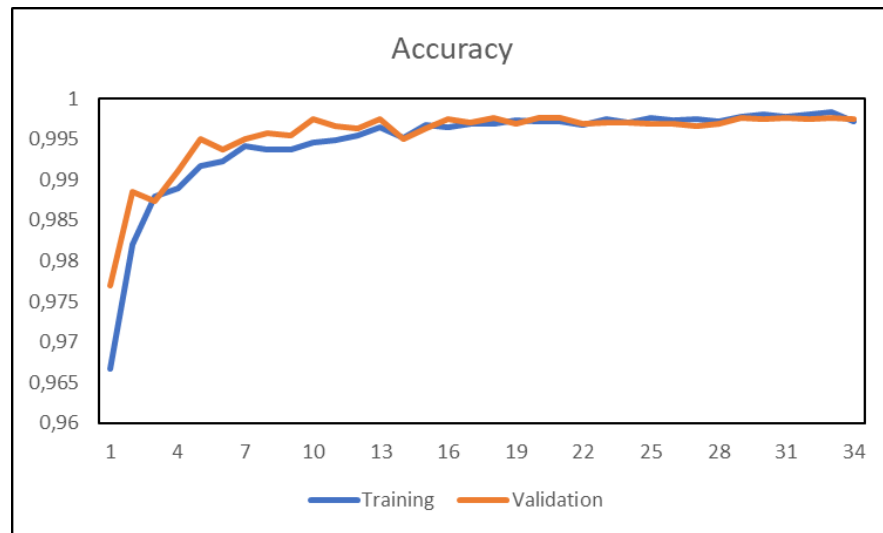


Figura 10: Accuracy del modelo 3 64-64-128-256

Fuente: Elaboración propia

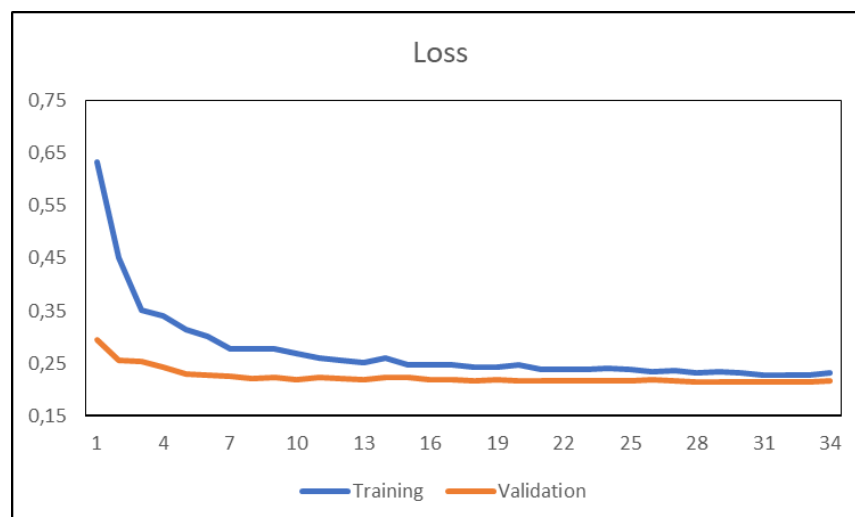


Figura 11: Loss del modelo 3 64-64-128-256

Fuente: Elaboración propia

3.1.4 Modelo 4 con filtros 50-100-500-1000-10

Finalmente entrenamos el modelo 4 con la configuración 50-100-500-1000-10, los 19083 registros y 35 épocas. Dando un accuracy muy bueno en relación con los otros modelos como se muestra en la Figura 12 donde alcanza un valor muy cercano a 1, es decir el modelo ha logrado aprender con mucha exactitud.

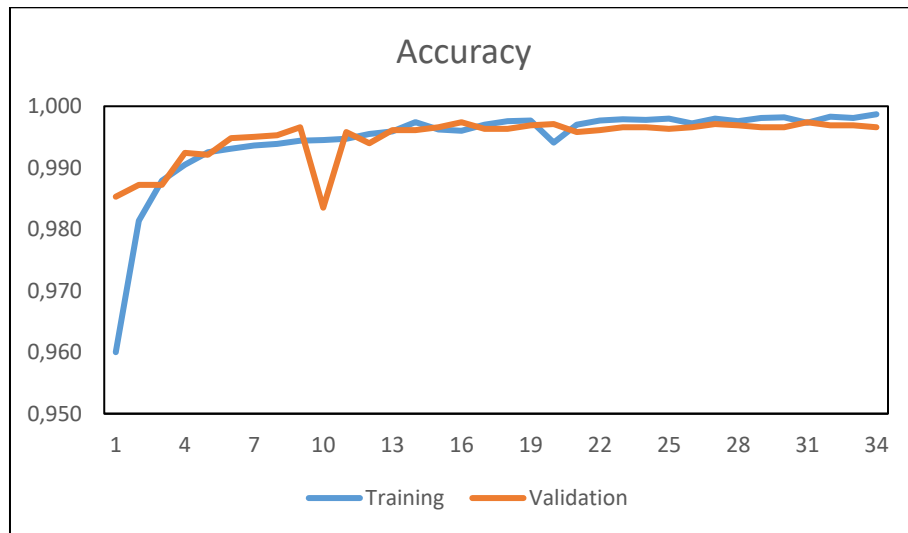


Figura 12: Acurracy del modelo 4 50-100-500-1000-10

Fuente: Elaboración propia

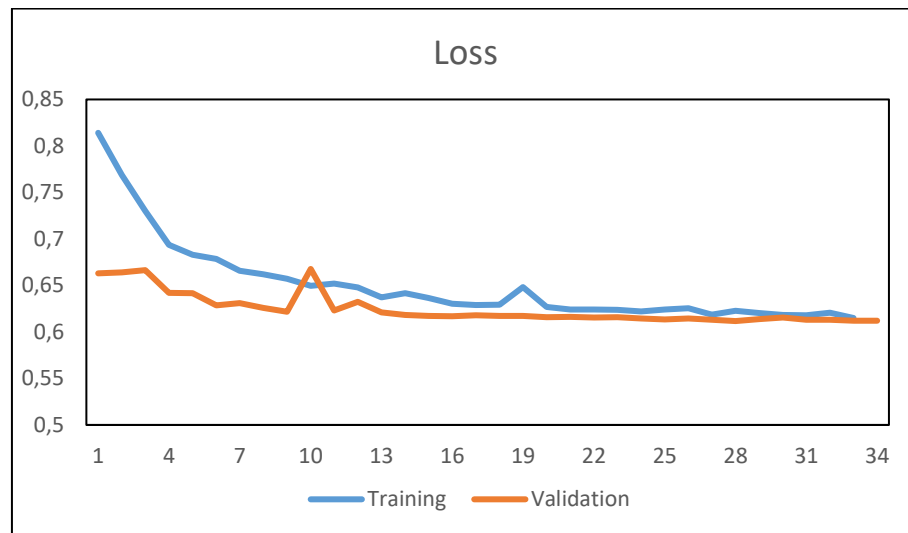


Figura 13: Loss del modelo 4 50-100-500-1000-10

Fuente: Elaboración propia

3.2 Matriz de confusión de las predicciones

Con la data de testeo tomada de los formularios de marzo del 2022 se generaron 1254 registros, que el modelo nunca antes había visto con el fin de identificar como clasificaba estos nuevos datos. Y determinar cuál de los modelos es el más exacto y preciso para predecir cada una de las clases.

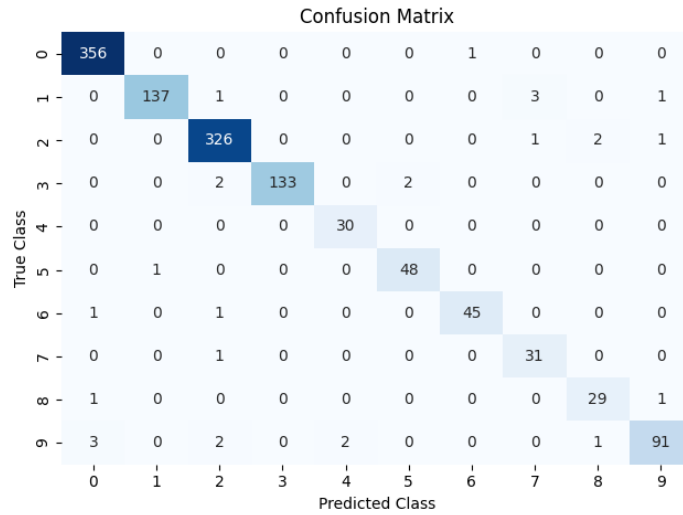


Figura 14: Matriz de confusión modelo 1

Fuente: Elaboración propia

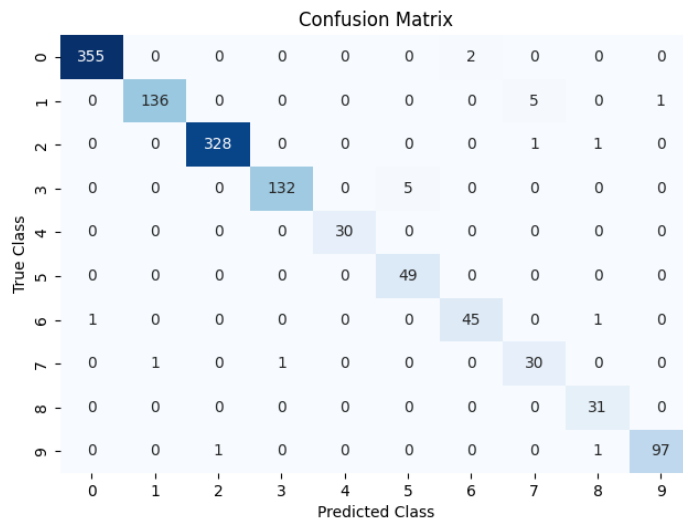


Figura 15: Matriz de confusión modelo 2

Fuente: Elaboración propia

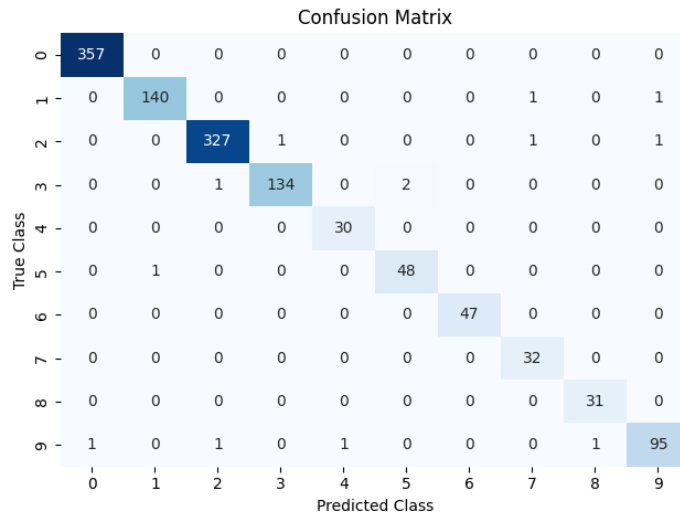


Figura 16: Matriz de confusión modelo 3

Fuente: Elaboración propia

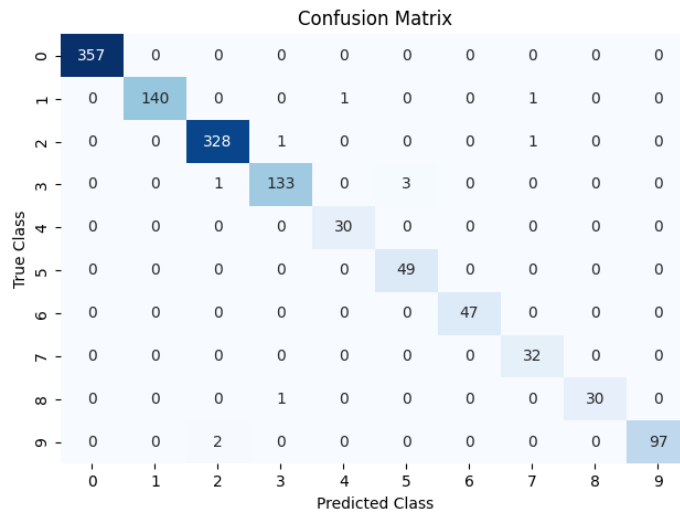


Figura 17: Matriz de confusión modelo 4

Fuente: Elaboración propia

3.3 Elección del mejor modelo

Una vez que se tiene el resultado de cada modelo entrenado y testeado, se procede a comparar para determinar el mejor modelo como se muestra en la Tabla 1. El resumen de cada modelo nos indica que en su mayoría poseen buenos resultados, son mayores al 90% pero de los 4 modelos el que destaca es el modelo 4 con una exactitud de 0.9987 y precisión de 0.9912.

Modelo	Accuracy(exactitud)	Precisión
1	0.9886	0.9776
2	0.9934	0.9833
3	0.9972	0.9896
4	0.9987	0.9912

Tabla 1: Exactitud y precisión de los 4 modelos

Fuente: Elaboración propia

En la Figura 18 se ven los resultados sobre un nuevo formulario que el modelo arroja, siendo estos correctamente predichos y validados en cada segmentación.

Uso Establecimientos de Salud Médicos de Libre
Médicos Legales o Registro

090107217

9) FECHA DE NACIMIENTO 0 12

1938 10 12

Año Mes Día

10) FECHA DE FALLECIMIENTO 03 04

2022 03 04

Año Mes Día

Figura 18: Predicciones del modelo

Fuente: Elaboración propia

CAPÍTULO 4

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- La calidad de las imágenes es de gran importancia al momento de extraer los datos para ser procesados, mientras posean mayor claridad y luminosidad ayudará al modelo a entender o predecir con facilidad.
- Los modelos realizados predicen correctamente más del 90% de los datos procesados, sin embargo, vimos que el mejor modelo es aquel que posee únicamente la data creada para entrenamiento y testeo.
- Construir modelos con datos externos e internos nos ayuda a que pueda ser aplicado en diferentes tipos de documentos físicos que posean dígitos manuscritos, identificando previamente donde se encuentran las secciones de interés.
- El uso de redes neuronales convolucionales facilita el recibir una gran cantidad de información, retener aquellos parámetros más importantes y desechar en gran cantidad aquellos que no aportan significativamente a la clasificación.

4.2 Recomendaciones

- Para obtener buenos modelos es importante tener una base de datos de calidad, que sea capaz de capturar la mayor cantidad de información relevante para el aprendizaje óptimo del modelo. Lo ideal sería complementar con una data de registros pasados.
- Probar diferentes configuraciones, ajustando los parámetros del modelo o utilizar algunas de estudios anteriores realizados.
- Mantener una constante revisión en la construcción de la data propia para ayuda a obtener modelos más robustos que produzcan mejores resultados al predecir los diferentes estilos de escritura.

Referencias

- Asanza, W. R., & Olivo, B. M. (2018). *Redes neuronales artificiales aplicadas al reconocimiento de patrones*. Editorial UTMACH.
- Baldominos, A., Saez, Y., & Isasi, P. (2019). A Survey of Handwritten Character Recognition with MNIST and EMNIST. *Applied Sciences*, 9(15), 3169. MDPI AG
- Bogaerts, W., Dahlem, M., Dwivedi, S., Jansen, R., & Rottenberg, X. (2020, February). Dispersive optical phased array circuit for high-resolution pixelated 2D far-field scanning controlled by a single wavelength variable. In *Smart Photonic and Optoelectronic Integrated Circuits XXII* (Vol. 11284, pp. 119-133). SPIE.
- Bonilla Carrión, C. (2020). *Redes Convolucionales*.
- Chilán Carrasco, M. F. (2020). *Construcción y entrenamiento de Redes Neuronales Artificiales en la Biblioteca de Tensorflow para el reconocimiento de caracteres numéricos* (Doctoral dissertation, Universidad de Guayaquil. Facultad de Ingeniería Industrial. Carrera de Licenciatura en Sistemas de Información.).
- Chollet, F. (2021). *Deep learning with Python*. Simon and Schuster.
- Gatica, G. N. (2021). *Introducción al análisis funcional. Teoría y aplicaciones*. Reverté.
- Gupta, N., Jalal, A.S. Traditional to transfer learning progression on scene text detection and recognition: a survey. *Artif Intell Rev* 55, 3457–3502 (2022). <https://doi.org/10.1007/s10462-021-10091-3>
- IBM Corporation. (2021). *El modelo de redes neuronales - IBM*. Obtenido de <https://www.ibm.com/docs/es/spss-modeler/saas?topic=networks-neural-model>

- Juan, R. Q., & Mario, C. M. (2011). Redes neuronales artificiales para el procesamiento de imágenes, una revisión de la última década. RIEE&C, Revista de Ingeniería Eléctrica, Electrónica y Computación, 9(1), 7-16.
- Kim, P. (2017). MATLAB Deep Learning. doi:10.1007/978-1-4842-2845-6
- Koloda, J., Seiler, J., & Kaup, A. (2022, May). Denoising-based image reconstruction from pixels located at non-integer positions. arXiv e-prints.
- Lars Künkel, Rajat M Thomas, Joris P W Verbiest, Detecting pulsars with neural networks: a proof of concept, Monthly Notices of the Royal Astronomical Society, Volume 506, Issue 1, September 2021, Pages 1111–1127, <https://doi.org/10.1093/mnras/stab1628>
- Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. IEEE transactions on neural networks and learning systems.
- Magro, R. (2013). Binarización de imágenes digitales y su algoritmia como herramienta aplicada a la ilustración entomológica. Boletín de la sociedad entomológica Aragonesa, 53, 443-464.
- Martínez Llamas, J. (2018). Reconocimiento de imágenes mediante redes neuronales convolucionales.
- Molina, E., Diaz, J., Hidalgo-Silva, H., & Chávez, E. (2018). Algoritmos de binarización robusta de imágenes con iluminación no uniforme. Revista Iberoamericana de Automática e Informática industrial, 15(3), 252-261.
- N S Jeya Shyla, W R Sam Emmanuel, Glaucoma Detection Using Multiple Feature Set With Recurrent Neural Network, The Computer Journal, 2022;, bxac093, <https://doi.org/10.1093/comjnl/bxac093>
- Rosyda, S. S., & Purboyo, T. W. (2018). A review of various handwriting recognition methods. International Journal of Applied Engineering Research, 13(2), 1155-1164.

- Rouhiainen, L. (2018). *Inteligencia artificial*. Madrid: Alienta Editorial.
- Sandoval Serrano, L. J. (2018). Algoritmos de aprendizaje automático para análisis y predicción de datos. *Revista Tecnológica*; no. 11.
- Shang, Y., Xu, D., Zong, Z., & Yan, Y. (2022). Network Binarization via Contrastive Learning. arXiv preprint arXiv:2207.02970.
- Shi, X., Qi, Z., Cai, J., Fu, K., Zhao, Y., Li, Z., Liu, X., & Liu, H. (2022). RepBNN: towards a precise Binary Neural Network with Enhanced Feature Map via Repeating. ArXiv, abs/2207.09049.
- Singh, S., Garg, N.K. & Kumar, M. Feature extraction and classification techniques for handwritten Devanagari text recognition: a survey. *Multimed Tools Appl* (2022). <https://doi.org/10.1007/s11042-022-13318-9>
- Sungho Suh, Sojeong Cheon, Wonseo Choi, Yeon Woong Chung, Won-Kyung Cho, Ji-Sun Paik, Sung Eun Kim, Dong-Jin Chang, Yong Oh Lee, Supervised segmentation with domain adaptation for small sampled orbital CT images, *Journal of Computational Design and Engineering*, Volume 9, Issue 2, April 2022, Pages 783–792, <https://doi.org/10.1093/jcde/qwac029>
- Syed Saqib Raza Rizvi, Muhammad Adnan Khan, Sagheer Abbas, Muhammad Asadullah, Nida Anwer, Areej Fatima, Deep Extreme Learning Machine-Based Optical Character Recognition System for Nastalique Urdu-Like Script Languages, *The Computer Journal*, Volume 65, Issue 2, February 2022, Pages 331–344, <https://doi.org/10.1093/comjnl/bxaa042>
- Talebi, H., & Milanfar, P. (2021). Learning to resize images for computer vision tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 497-506).
- Valentin Debarnot, Pierre Weiss, Blind inverse problems with isolated spikes, *Information and Inference: A Journal of the IMA*, 2022;, iaac015, <https://doi.org/10.1093/imaiai/iaac015>

Wu, J. (2017). Introduction to convolutional neural networks. National Key Lab for Novel Software Technology. Nanjing University. China, 5(23), 495.