Pontificia Universidad JAVERIANA Cali

Facultad de Ingeniería y Ciencias

Acta de Correcciones al Proyecto de Grado
Ingeniería de sistemas

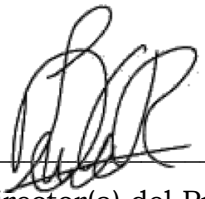
Fecha: 15/02/2024

Autores: Nicolle Naranjo Astaiza, María José Suarez

Nombre del Proyecto de Grado: Modelo de red neuronal convolucional para la traducción de gestos en lenguaje de señas colombiano a texto enfocado en el sector hotelero colombiano

Director: Diego Linares, Gloria Inés Álvarez

Como indica el artículo 2.27 de las Directrices de Trabajo de Grado, he verificado que los estudiantes indicados arriba han implementado todas las correcciones que los Jurados del Proyecto de Grado definieron que se efectuaran, como consta en el Acta de Calificación correspondiente.



Firma de Director(a) del Proyecto de Grado

Nota de Aceptación

Aprobado por el Comité de Trabajo de Grado en cumplimiento de los requisitos exigidos por la Pontificia Universidad Javeriana para optar el título de Ingeniero de Sistemas y Computación.

Dr. CAMILO ROCHA
Decano de la Facultad de Ingeniería

Dr. GERARDO MAURICIO SARRIA
Director Carrera Ingeniería Sistemas y Computación.

Dr. DIEGO LUIS LINARES
Dr. GLORIA INES ALVAREZ
Director(a) Trabajo

Dr. MARIA CONSTANZA PABON
Jurado 1

Mag. MARIO JULIAN MORA
Jurado 2

Modelo de red neuronal convolucional para la traducción de gestos en lenguaje de señas colombiano a texto enfocado en el sector hotelero colombiano

Nicolle Naranjo Astaiza
María José Suárez

Pontificia Universidad Javeriana Cali
Facultad de Ingeniería y Ciencias
Ingeniería de Sistemas y Computación
Cali
2023

Modelo de red neuronal convolucional para la traducción de gestos en lenguaje de señas colombiano a texto enfocado en el sector hotelero colombiano

Nicolle Naranjo Astaiza
María José Suárez

Proyecto de Grado

*D*irectores: Dr. Diego Linares
Dra. Gloria Inés Álvarez

Pontificia Universidad Javeriana Cali
Facultad de Ingeniería y Ciencias
Ingeniería de Sistemas y Computación
Cali
2023

Resumen

El objetivo principal de este proyecto es apoyar a las personas con discapacidad sensorial auditiva, permitiéndoles llevar a cabo acciones básicas en entornos donde el personal no está debidamente capacitado para ofrecer una atención totalmente adecuada, como suele ser en el sector hotelero colombiano. Por lo tanto, es cada vez más necesario el desarrollo de una herramienta computacional capaz de reconocer y traducir gestos manuales del lenguaje de señas colombiano (LSC) a texto. Esta herramienta no solo busca cubrir una necesidad real en la sociedad, sino que también permitirá una atención inclusiva y una experiencia positiva para todos los involucrados.

Los resultados obtenidos de este proyecto demuestran de manera concluyente la funcionalidad de la herramienta desarrollada. La identificación de los datos es efectiva, y la traducción a texto es precisa. La herramienta es capaz de reconocer los gestos manuales del LSC de acuerdo con el corpus utilizado.

Keywords: Lenguaje de señas colombiano, sector hotelero, Colombia, accesibilidad, Discapacidad, Necesidad

Índice

1. Descripción del Problema **8**

1.1. Planteamiento del Problema 8

1.1.1. Formulación 9

1.1.2. Sistematización 9

1.2. Objetivos 9

1.2.1. Objetivo General 9

1.2.2. Objetivos Específicos 9

1.3. Justificación 10

2. Marco de Referencia **10**

2.1. Marco Teórico 10

2.1.1. Lenguaje de señas 10

2.1.2. Aprendizaje automático 13

2.1.3. Visión artificial 13

2.1.4. Redes neuronales 13

2.1.5. Redes neuronales convolucionales 14

2.1.6. Transferencia de aprendizaje 14

2.1.7. Modelo VGG-16 15

2.2. Antecedentes 17

3. Desarrollo de la solución **18**

4. Preparación de datos **18**

5. Construcción de modelos **19**

5.1. Redes neuronales convolucionales (CNN) 19

5.1.1. Modelo Inicial 19

5.1.2. Modelo Simplificado 20

5.1.3. Modelo Preentrenado 22

6. Evaluación de resultados **23**

6.1. Análisis Modelo Inicial 23

6.2. Análisis Modelo Inicial con búsqueda de hiperparámetros 25

6.3. Análisis Modelo Simplificado 28

6.4. Análisis Modelo Simplificado con búsqueda de hiperparámetros 30

6.5. Análisis Modelo VGG-16 33

6.6. Análisis Modelo VGG-16 con búsqueda de hiperparámetros 36

7. Evaluación final **38**

8. Despliegue del modelo 39

9. Conclusiones 40

10.Trabajos futuros 41

1. Descripción del Problema

1.1. Planteamiento del Problema

Para el año 2021, según una proyección del Departamento Administrativo Nacional de Estadística (DANE), se estimó que aproximadamente 459.784 personas en Colombia presentaban discapacidad auditiva [1]. De ese total, el 60 % eran adultos mayores que habían perdido la audición, mientras que el otro 40 % eran adultos y menores que habían nacido con discapacidad auditiva o la habían adquirido en algún momento de su vida [2]. Esta condición puede afectar el desarrollo del lenguaje y ocasionar dificultades en la competencia lectora funcional, así como en la habilidad de expresarse y entender el español escrito.

Como resultado del estudio previo, un porcentaje significativo de las personas con discapacidad auditiva en Colombia se encuentran excluidas de situaciones cotidianas, como comprender folletos informativos, aplicaciones web o solicitar instrucciones, entre otras. Esto se debe a la dificultad que tienen para leer y escribir en español. Según un estudio realizado por La República en 2022 [3], la dificultad de lectura y escritura en personas que nacen con discapacidad auditiva se debe a que no aprendieron su primer idioma en su totalidad desde temprana edad. Es necesario que se apropien de su primer lenguaje para poder expresarse y comprender, y así eventualmente puedan aprender español escrito. De hecho, un artículo publicado en The Conversation en 2020 [4] indica que las personas con discapacidad sensorial auditiva pueden ser mucho más eficaces a la hora de leer texto, pero no todos pueden llegar a aprender esta habilidad. Por lo tanto, es importante que, mientras se incluye a esta población en el sistema educativo, se proporcionen herramientas que les permitan expresarse sin inconvenientes en el lenguaje que conocen.

En una investigación llevada a cabo por Carmela Isabel De la Hoz, Administradora Hotelera y de Turismo, que analizó la accesibilidad en sector hotelero en Barranquilla [5], se evidenció que solo el 12 % de los hoteles encuestados disponían de personal debidamente capacitado en el dominio del Lenguaje de Señas. Esto confirma las limitaciones existentes en la formación del personal para poder comunicarse adecuadamente con estas personas, lo que dificulta brindar una atención satisfactoria. Por ello, con las dificultades encontradas, es importante buscar medidas que apoyen a las personas con discapacidad sensorial auditiva que solo saben comunicarse con lenguaje de señas, para permitirles accesibilidad a la hora de entablar una conversación con el personal del hotel. La implementación de accesibilidad y, a su vez, la capacitación que se debe dar al personal de los hoteles para atender a las personas con discapacidad sensorial auditiva es un punto muy importante a resaltar, es necesario que los hoteles y establecimientos turísticos en Colombia puedan desarrollar medidas no solo por acciones legales en cuanto a ofrecer un servicio digno y accesible, sino también para que estas situaciones apoyen a la inclusión de estas personas a la sociedad, y así, poder realizar acciones básicas por sí solos.

En conclusión, otro enfoque para afrontar la situación mencionada anteriormente

que afecta la población de personas con discapacidad sensorial auditiva de Colombia y que saben lenguaje de señas es el desarrollo de una herramienta computacional que detecte e identifique imágenes de los gestos manuales de LSC y traduzca su significado a texto. Es decir, esta herramienta ayudará a aquellos que solo pueden expresarse en LSC para que logren comunicarse con el personal de los hoteles (que no estén capacitados para comunicarse en LSC) y se hagan entender. Adicionalmente, ayudaría a brindar un servicio accesible e inclusivo, mejorando la experiencia del usuario y la imagen del hotel.

1.1.1. Formulación

¿Cómo desarrollar una herramienta computacional capaz de detectar gestos manuales en un ambiente hotelero para apoyar a las personas con diversidad sensorial auditiva?

1.1.2. Sistematización

- ¿Cómo preparar un dataset con imágenes de gestos enfocado a un ambiente hotelero?
- ¿Cómo entrenar un modelo de reconocimiento de imágenes?
- ¿Cómo evaluar el modelo entrenado?
- ¿Cómo desarrollar un prototipo funcional que implemente el modelo de aprendizaje?

1.2. Objetivos

1.2.1. Objetivo General

Desarrollo de una herramienta computacional capaz de detectar gestos manuales en un ambiente hotelero para apoyar a las personas con diversidad sensorial auditiva.

1.2.2. Objetivos Específicos

- Preparar un dataset en un ambiente hotelero para apoyar a las personas con diversidad sensorial auditiva
- Entrenar modelo de aprendizaje para reconocimiento y clasificación de imágenes.
- Evaluar modelo entrenado para que sea capaz de clasificar nuevas imagenes de manera exitosa.

- Desarrollar un prototipo funcional que implemente el modelo de aprendizaje que permita probar y demostrar su eficacia en la clasificación de imágenes, a su vez, evaluando la viabilidad.

1.3. Justificación

A pesar de que una gran parte de la comunidad con discapacidad auditiva puede aprender a leer, escribir y comunicarse adecuadamente con el mundo, no todos estos tienen las mismas capacidades, como tampoco los mismos recursos para acceder a esa educación y terapia que les ayuda a llegar a ese punto. Estos los excluye de muchas situaciones cotidianas que afectan su desarrollo personal y su socialización con el mundo (para este contexto en el sector hotelero). Con este proyecto se quiere crear un prototipo que aporte a la investigación de herramientas que ayuden a aquellas personas con diversidad auditiva que saben lenguaje de señas. Esto permitirá apoyar la comunicación entre aquellos que saben lenguaje de señas y los que no, de una manera viable. Por lo tanto, aporta a la comunidad con diversidad sensorial auditiva con problemas de entendimiento del español escrito al implementar un recurso que permite la comunicación entre ellos y la sociedad usando el lenguaje de señas que ya conocen y donde se expresan mejor.

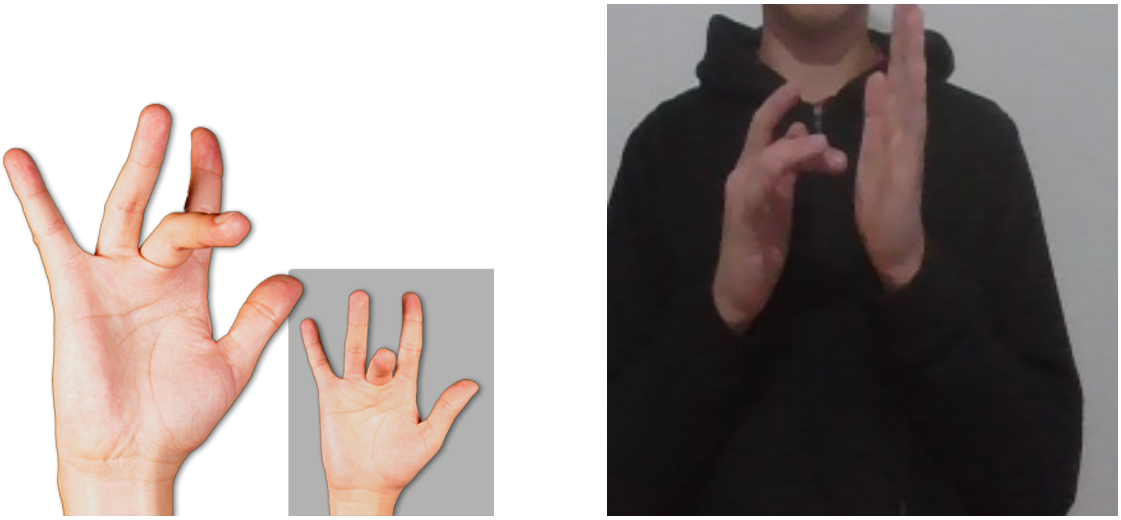
2. Marco de Referencia

2.1. Marco Teórico

El reconocimiento de imágenes ha avanzado bastante desde los años 60s, época en donde la inteligencia artificial comenzaba a tomar más peso y se convertía en una disciplina. En la actualidad se aplican algoritmos complejos y aprendizaje profundo para la extracción y clasificación de datos en imágenes digitales. Una implementación de este es el reconocimiento de lenguaje de señas a partir de diferentes métodos y técnicas [6], antes de aplicarlos se necesitan definir algunos conceptos importantes como: lenguaje de señas, aprendizaje automático, visión artificial, redes neuronales, redes neuronales convolucionales, transferencia de aprendizaje y VGG-16.

2.1.1. Lenguaje de señas

El lenguaje de señas es como otros lenguajes, pero sin la necesidad de la voz y la escucha, lo que la hace una herramienta útil para aquellos con diversidad sensorial auditiva. Esta se basa en el movimiento y los gestos manuales apoyados de expresiones faciales para comunicarse. Debido a que es utilizada principalmente por la comunidad con problemas de audición, no es común que las personas externas estén familiarizadas o entienden este lenguaje; lo cual crea una brecha social entre



(a) Imagen animada.

(b) Imagen extraída del dataset.

Figura 1: Gesto de “Abril” en LSC.

ambas comunidades. Por lo que para construir un puente que facilite la conversación con aquellos con problemas de audición se han hecho múltiples investigaciones y experimentos en el área de las ciencias de la computación para encontrar distintas soluciones [7].

A continuación se muestran algunas imágenes con ejemplos de gestos y sus respectivos significados:

Figura 2: Gesto de “Con mucho gusto” en LSC

Figura 3: Gesto de “Habitación” en LSC.

2.1.2. Aprendizaje automático

Es una rama de la inteligencia artificial que se enfoca en construir sistemas que aprenden en función de los datos que consumen. Esta área busca estudiar y desarrollar métodos que puedan "aprender" y usar datos como entrenamiento para optimizar su rendimiento. Esto le permite a los modelos hacer predicciones y tomar decisiones más precisas de manera autónoma [8]. Existen tres tipos principales de aprendizaje automático:

- **Aprendizaje supervisado:** los datos disponibles ya se encuentran procesados y etiquetados, se usa para clasificar datos.
- **Aprendizaje no supervisado:** los datos disponibles no están etiquetados y se usan para explorar datos desconocidos y encontrar patrones.
- **Aprendizaje por refuerzo:** El modelo aprende a través de la interacción con un entorno, tomando decisiones para maximizar una recompensa a lo largo del tiempo.

2.1.3. Visión artificial

Es un campo de la inteligencia artificial dedicado al estudio y desarrollo de métodos que permiten adquirir, analizar y procesar imágenes extraídas del mundo real para obtener información y tomar decisiones a partir de estas. Esta busca trabajar de manera similar a la visión de los humanos para la distinción y extracción de características. [9].

2.1.4. Redes neuronales

Son sistemas computacionales basados en el funcionamiento del cerebro animal, pues consiste en un conjunto de nodos interconectados entre sí (neuronas artificiales) en una estructura de capas. Estos sistemas son un tipo de aprendizaje automático conocido como aprendizaje profundo [10]. Existen varios tipos de redes neuronales:

- **Monocapa (perceptrón simple):** está compuesta por una sola capa de entrada y una sola capa de salida donde luego se realizan los cálculos.
- **Multicapa (perceptrón múltiple):** compuesta por una capa de entrada y una de salida, pero con varias capas intermedias entre ambas para procesar y hacer cálculos.
- **Convolucional:** es una variación de la red neuronal multicapa, donde no todas las neuronas están interconectadas con la siguiente capa, sino que especializa algunas para reducir la complejidad y los recursos.

- **Redes Preentrenadas:** estas redes se han entrenado con más de un millón de imágenes, y pueden clasificar hasta 1000 categorías diferentes [11]. Pueden ser utilizadas como punto de partida para tareas específicas, por ello, su aplicación en la transferencia de aprendizaje ofrece eficiencia y simplicidad en comparación con la creación de una red neuronal desde cero.

2.1.5. Redes neuronales convolucionales

Las redes neuronales convolucionales (CNN) destacan por su eficacia en el procesamiento de imágenes, voz y señales de audio. Están compuestas por tres tipos principales de capas: la capa convolucional, la capa de agrupación y la capa totalmente conectada [12].

La capa convolucional, fundamental en las CNN, realiza la mayoría de los cálculos. Utiliza un detector de características, conocido como kernel o filtro, para explorar la imagen y realizar operaciones de convolución. Esta capa identifica patrones y características cada vez más grandes a medida que procesa la imagen, contribuyendo a la identificación final del objeto.

La capa de agrupación, también llamada submuestreo, reduce la dimensión de la entrada mediante la aplicación de una función de agregación, como la agrupación máxima o media.

La capa totalmente conectada en una red neuronal implica que cada nodo de salida está directamente vinculado a cada nodo de la capa anterior. Su función principal es clasificar utilizando las características extraídas por capas anteriores y filtros. A menudo, utiliza la función de activación softmax para asignar probabilidades a las distintas clases.

2.1.6. Transferencia de aprendizaje

En la transferencia de aprendizaje, se aprovecha la idea de reutilizar elementos de modelos de aprendizaje automático preentrenados en nuevos modelos destinados a tareas similares. Al compartir conocimientos entre modelos previamente desarrollados para funciones afines, se logra optimizar recursos y reducir la necesidad de datos etiquetados durante el entrenamiento. Este enfoque, cada vez más esencial en la evolución del aprendizaje automático, ha ganado prominencia en el desarrollo de nuevos modelos al disminuir significativamente la carga de recursos y tiempo requeridos en comparación con el enfoque tradicional, especialmente en el aprendizaje supervisado, donde la etiqueta de grandes conjuntos de datos es esencial [13].

Esta técnica se está siendo utilizada principalmente en:

- Procesamiento de lenguaje natural
- Visión artificial

- **Redes neuronales:** el entrenamiento de estas redes es intensivo en recursos debido a la complejidad de los modelos. La transferencia de aprendizaje se emplea para optimizar este proceso y reducir la carga de recursos. La transferencia de conocimiento o características entre redes facilita el desarrollo eficiente de nuevos modelos, aplicando este conocimiento a través de diversas tareas o entornos.

2.1.7. Modelo VGG-16

Visual Geometry Group (VGG) es una arquitectura estándar de CNN profunda creada por Karen Simonyan y Andrew Zisserman de la Universidad de Oxford en 2014 [14]. Este modelo es muy popular para la detección y clasificación de imágenes. En general, su estructura utiliza 16 capas, 138 millones de parámetros e implementa capas convolucionales, pooling y una capa completamente conectada de salida.

El VGG16 utiliza 16 capas convolucionales con activación de unidad lineal rectificada (ReLu), una de las funciones más populares que permite reducir el tiempo de entrenamiento. Las 13 primeras capas implementan un campo receptivo de 3x3 y van duplicando la cantidad de filtros kernel a medida que se van agregando capas: empezando por 64, luego 128, 256 y 512 filtros respectivamente. Cada capa de convolución va acompañada de capas Max-Pooling de 2x2. Las últimas 3 capas son conocidas como capas completamente conectadas, es decir que cada neurona de entrada se encuentra conectada a otra neurona de salida, las primeras dos cuentan con 4096 canales y la última 1000 canales [15].

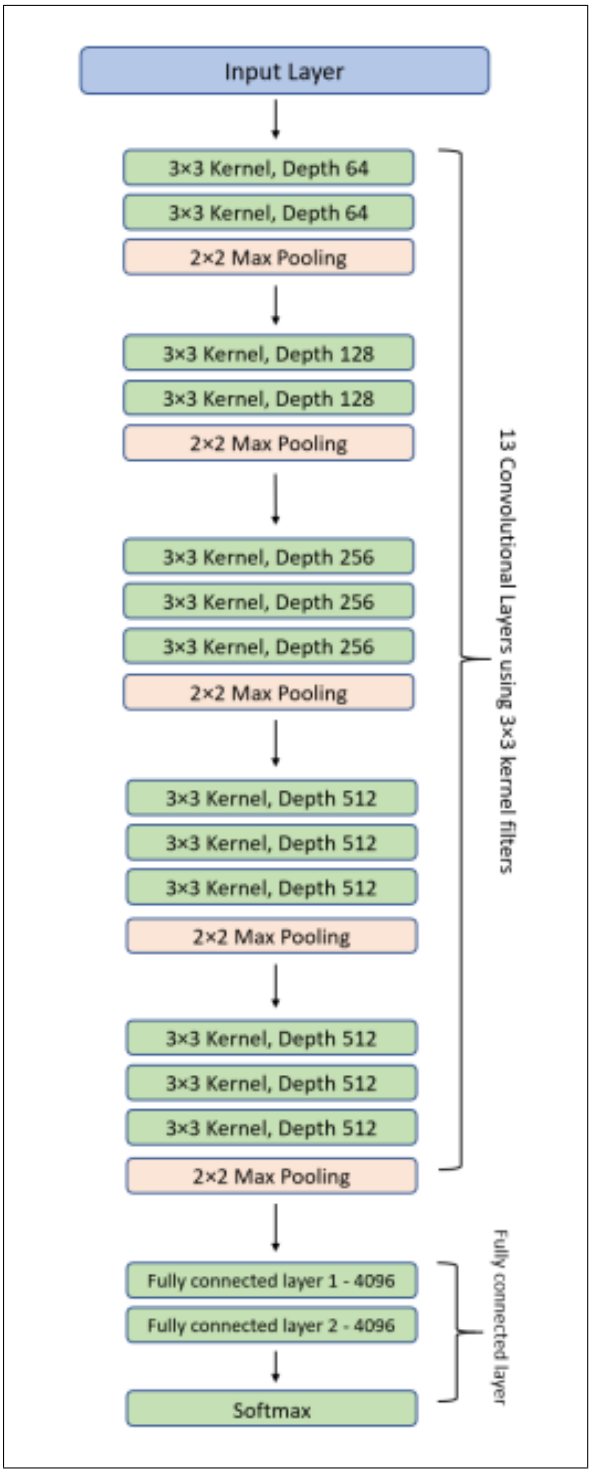


Figura 4: Arquitectura Modelo VGG 16 [16]

2.2. Antecedentes

En los últimos años, el reconocimiento de lenguaje de señas se ha convertido en una de las áreas populares para la investigación en ciencias de la computación. Resaltando lo esencial que es la comunicación entre personas que saben lenguaje de señas y aquellas que no, se han propuesto múltiples enfoques que buscan construir traductores para diversos lenguajes de señas que faciliten la comunicación. Para la construcción de estas soluciones, se hace frente a desafíos como: la detección precisa de señas en entornos variables, la selección de herramientas y datos. Dentro de los numerosos métodos utilizados, se pueden destacar dos grupos: las soluciones basadas en el uso de sensores con mediciones a mano y el basado en visión, donde se utilizan cámaras [17].

Para ambas metodologías, se han abordado distintas implementaciones, desde enfoques clásicos que utilizan modelos de Markov para el reconocimiento de patrones y secuencias hasta el uso de modelos con aprendizaje profundo. Entre los diferentes planteamientos que aplican métodos clásicos continuos, se tienen ejemplos como el uso de arquitecturas de red de cápsula profunda unidimensional (CapsNet) para el reconocimiento continuo de lenguaje de señas indio (ISL) mediante señales obtenidas de un sistema de unidad de medición inercial (IMU), un dispositivo electrónico sensorial que mide la aceleración, orientación, velocidades y otras fuerzas [18]. Otro estudio orientado más a modelos y sistemas de reconocimiento, propone un sistema automático que descompone los signos en subunidades manejables, utilizando un marco para segmentar y rastrear objetos de la piel en videos continuos de gestos de lenguajes de señas americano, y emplea un algoritmo de refuerzo para combinar características extraídas en clasificadores sólidos para cada signo; implementado una estrategia de aprendizaje conjunto para mejorar la clasificación de gestos de señas al compartir subunidades entre clases [19]. Enfoques más modernos implementan aprendizaje profundo, como el estudio para reconocimiento de lenguaje de señas alemán (DGS) que utiliza una red 3D-ResNet y un codificador-decodificador con CTC, optimizados alternadamente, mejorando la precisión de reconocimiento en benchmarks relevantes[20].

Basándose en los diferentes métodos que se han implementado para el reconocimiento eficaz de los lenguajes de señas, estos han demostrado tener resultados considerablemente buenos que tienen sus ventajas y desafíos. Estos aún deben de enfrentarse a nuevos retos y limitaciones como: no depender de la interacción con dispositivos hardware, mejor precisión en los resultados, la falta de conjuntos de datos y la inclusión de una mayor variedad de lenguajes de señas, entre otras. Esta revisión detallada de los diversos estudios ofrece un referente para comprender cómo se desarrollan los sistemas de reconocimiento de lenguaje de señas y cómo han evolucionado las estrategias para manejar señales. Aunque se ha realizado una considerable investigación en esta área, el reconocimiento continuo del lenguaje de señas sigue siendo un desafío que requiere una mayor exploración y más soluciones innovadoras.

3. Desarrollo de la solución

Para identificar un modelo adecuado para el entrenamiento con este conjunto de datos, es esencial reducir el espacio de búsqueda que pueda ser adecuado para el corpus que se utilizó. Con esto en mente, se propuso la creación y manejo de tres modelos distintos: uno implementado desde cero, otro preentrenado y un tercero obtenido de un artículo. Cada uno de ellos fue entrenado.

Posteriormente, utilizando los mismos modelos, se realizó la búsqueda de hiperparámetros, lo que resultó en la obtención de otros 3 modelos adicionales, los cuales también fueron entrenados. De esta manera, se obtuvieron en total seis modelos, los cuales fueron analizados en función de sus métricas utilizando la carpeta "Test".

Con los resultados obtenidos, se estimó el modelo más adecuado y este fue implementado en el prototipo funcional.

Para llevar a cabo este proyecto, se emplearon las siguientes bibliotecas y herramientas en Google Colab: TensorFlow, Keras, Keras Tuner, OpenCV, Pandas, Numpy, Matplotlib, Scikit-learn.

4. Preparación de datos

El corpus cuenta con dos carpetas: "Entrenamiento" y "Validación", cada una con 39 subcarpetas que representan las clases de los gestos [21]. Estos son estáticos y se compone de imágenes. Las palabras que representan estas imágenes son sencillas y fáciles de comprender. Los 39 gestos incluidos este corpus son:

1. Abril
2. Adulto
3. Agosto
4. Bien
5. Bienvenido
6. Cama
7. Cinco
8. Cómo está
9. Con mucho gusto
10. Cuatro
11. Diciembre
12. Domingo
13. Dos
14. Enero
15. Febrero
16. Gracias
17. Habitación
18. Hola
19. Hotel
20. Jueves
21. Julio
22. Junio
23. Lunes
24. Mal
25. Martes
26. Marzo
27. Mayo
28. Miércoles
29. Niño
30. No
31. Noviembre
32. Octubre
33. Por favor
34. Sábado

35. Septiembre
36. Sí
37. Tres
38. Uno
39. Viernes

Cada gesto de la carpeta “Entrenamiento” cuenta con 1000 imágenes, excepto “Cama” y “Hotel”, que contienen 1001 imágenes cada uno. Cada gesto de la carpeta “Validación” tiene 50 imágenes. Todas las imágenes tienen dimensiones de 200 x 200 píxeles.

Para probar los modelos, se creó una carpeta “Test” con 50 imágenes aleatorias extraídas de la carpeta “Entrenamiento” por cada subcarpeta. Es importante destacar que estas imágenes fueron eliminadas de la carpeta “Entrenamiento” después de su selección para garantizar que el conjunto de prueba sea independiente y no contenga duplicados de las imágenes de entrenamiento.

En este orden de ideas, la carpeta “Test” quedó con 1950 imágenes y la carpeta “Entrenamiento” quedó con 37052 imágenes.

5. Construcción de modelos

5.1. Redes neuronales convolucionales (CNN)

Para la identificación y clasificación de imágenes se optó por utilizar CNN, pues su arquitectura permite adaptar fácilmente las dimensiones y características de las imágenes. Su estructura está diseñada para capturar los aspectos visuales relevantes de estas y encontrar patrones, haciendo un uso eficiente de los recursos. Esta red neuronal implementa un procesamiento de datos que se asemeja a una cuadrícula o matriz, la cual facilita la detección de características locales. Son estas cualidades que han convertido las CNN en una de las arquitecturas más populares para resolver problemas de visión por computadora, especialmente la identificación de objetos.

Uno de los desafíos de la investigación es plantear y encontrar los modelos de redes convolucionales más competentes para obtener los resultados más óptimos. Por lo tanto, se decidió comparar entre tres tipos de arquitecturas de CNN: un modelo inicial con pocas capas, un modelo simplificado tomado de otra investigación con más capas y un modelo preentrenado conocido. Adicionalmente, en cada uno de los modelos se implementaron hiperparámetros y métricas como herramientas e indicadores, para realizar una evaluación precisa y obtener una mejor retroalimentación.

5.1.1. Modelo Inicial

Para este primer modelo, se optó por una arquitectura simple como punto de partida para medir los resultados desde una estructura sencilla a otras más complejas.

De acuerdo a esto, se hizo lo siguiente:

Se creó una función que construye un modelo secuencial utilizando la librería de keras, esta permite crear un modelo con una estructura de pila lineal de capas, las cuales se recorren secuencialmente. Ya que se están implementando redes convolucionales, se incluyen dos tipos de capas esenciales en esta arquitectura: la capa de convolución y la capa de pooling. La primera se encarga de extraer la información específica de una imagen utilizando filtros o kernels, gracias a esto se pueden identificar patrones; por otro lado, se aplica la capa pooling, la cual hace una reducción de dimensionalidad de las características que genera la capa de convolución, al siempre escoger un valor máximo del conjunto de valores proporcionado. Para este primer modelo se usan únicamente dos capas de convolución y sus respectivas capas de pooling.

Estructura del modelo inicial		
Capa	**Parámetros**	**Valores**
Conv2D I	Filters	32
	Kernel size	3x3
	Activation	ReLu
	Input Shape	(200, 200, 3)
	Padding	Valid
MaxPooling2D I	Pool size	2x2
	Stride	2
Conv2D II	Filters	64
	Kernel size	3x3
	Activation	ReLu
	Input Shape	(200, 200, 3)
	Padding	Valid
MaxPooling2D II	Pool size	2x2
	Stride	2
Flatten	NA	NA
Dropout	Rate	0.5
Dense	Unidad	39
	Dimensionalidad	
	Activation	Softmax

Cuadro 1: Arquitectura Modelo Inicial

5.1.2. Modelo Simplificado

Este modelo fue creado por Yassine Ghouzam para el reconocimiento de dígitos entrenado en un conjunto de datos MNIST (National Institute of Standards and Technology) y la publicó en un notebook para la plataforma de Kaggle [22]. La base de datos MNIST es una colección de datos muy popular para el entrenamiento y procesamiento de imágenes. Para su aplicación, solo utilizó dos épocas de entrena-

miento, consiguiendo un 97 % de exactitud. Sin embargo, para este corpus y gestión de datos se implementaron 20 épocas.

Este CNN es muy similar al primero, pues implementa la librería de Keras para la construcción del modelo y también utiliza las capas importantes de convolución y pooling; sin embargo, este agrega dos capas extra de convolución (cuatro capas en total), donde por cada par de capas de convolución se agrega una capa de pooling y una de dropout, que regulariza la conexión entre nodos para asegurar que no hayan codependencias y evitar el sobreajuste.

Teniendo en cuenta que también se trata de un modelo secuencial, el orden afecta también cómo se están procesando los datos. A pesar de tener una estructura un poco más profunda que la arquitectura propuesta, esta es simplificada y más compacta.

Estructura del modelo simplificado		
Capa	**Parámetros**	**Valores**
Conv2D I	Filters	32
	Kernel size	5x5
	Input shape	(200, 200, 3)
	Padding	Same
	Activation	ReLu
Conv2D II	Filters	32
	Kernel size	5x5
	Input shape	(200, 200, 3)
	Padding	Same
	Activation	ReLu
MaxPooling2D I	Pool size	2x2
Dropout I	Rate	0.25
Conv2D III	Filters	64
	Kernel size	3x3
	Input shape	(200, 200, 3)
	Padding	Same
	Activation	ReLu
Conv2D IV	Filters	64
	Kernel size	3x3
	Input shape	(200, 200, 3)
	Padding	Same
	Activation	ReLu
MaxPooling2D II	Pool size	2x2
	strides	2
Dropout II	Rate	0.25
Flatten	NA	NA
Dense I	Unidad	256
	Dimensionalidad	
	Activation	ReLu
Dropout III	Rate	0.5
Dense II	Unidad	39
	Dimensionalidad	
	Activation	Softmax

Cuadro 2: Arquitectura Modelo Simplificado

5.1.3. Modelo Preentrenado

Para construir este modelo, se empleó la biblioteca TensorFlow, la cual posibilita la implementación de la arquitectura VGG 16. En primer lugar, se establece la entrada según el tamaño de las imágenes del conjunto de datos (200, 200, 3) y se excluye la capa superior totalmente conectada, es debido a que el modelo VGG 16 fue entrenada en el conjunto de datos ImageNet, por lo que es común modificar la

capa superior para adaptarla a la nueva tarea.

Posteriormente, se congelan todas las capas del modelo VGG 16. Este paso es para aprovechar el entrenamiento previo y evitar que esos conocimientos se modifiquen en exceso durante el entrenamiento. Además, la congelación mitigará el riesgo de sobreajuste al tener un conjunto de datos relativamente pequeño.

Finalmente, se agrega una capa de aplanado (Flatten) a la salida del modelo VGG16, seguida por una capa totalmente conectada (Dense) con 39 unidades y una función de activación softmax.

6. Evaluación de resultados

Para evaluar apropiadamente los resultados es necesario también escoger las métricas y herramientas adecuadas para medir su rendimiento. Para los tres modelos implementados se hizo una búsqueda con hiperparámetros para encontrar la configuración más óptima del modelo y evitar situaciones como el sobreajuste y adaptar correctamente antes de evaluar. Después de hacer el entrenamiento y obtener los resultados, se definieron métricas de evaluación para la clasificación de imágenes como: la precisión (precision), sensibilidad (recall) y la media entre ambas (F1-Score). Por último, para visualizar resultados y las métricas establecidas, la matriz de confusión es la opción más conveniente para representar los datos y analizarlos.

Para asegurar mejores resultados y ampliar el material a evaluar como se mencionó anteriormente, se utilizó Hyperparameter Tuning (Optimización de hiperparámetros) de Keras. Estos permiten modificar y ajustar tanto la arquitectura como el comportamiento de la CNN. Es una forma preestablecida de proporcionar valores e instrucciones alternas al modelo para experimentar de manera eficaz entre varias "rutas" de entrenamiento. En otras palabras, se trata de insertar espacios de búsqueda para definir unos rangos de valores que se irán probando en distintas combinaciones hasta finalmente encontrar los parámetros más adecuados.

6.1. Análisis Modelo Inicial

Para cada análisis se usan matrices de confusión y la tabla de métricas con los ponderados. La matriz de confusión muestra la cantidad de predicciones correctas e incorrectas que realiza el modelo en cada clase, como se muestra a continuación:

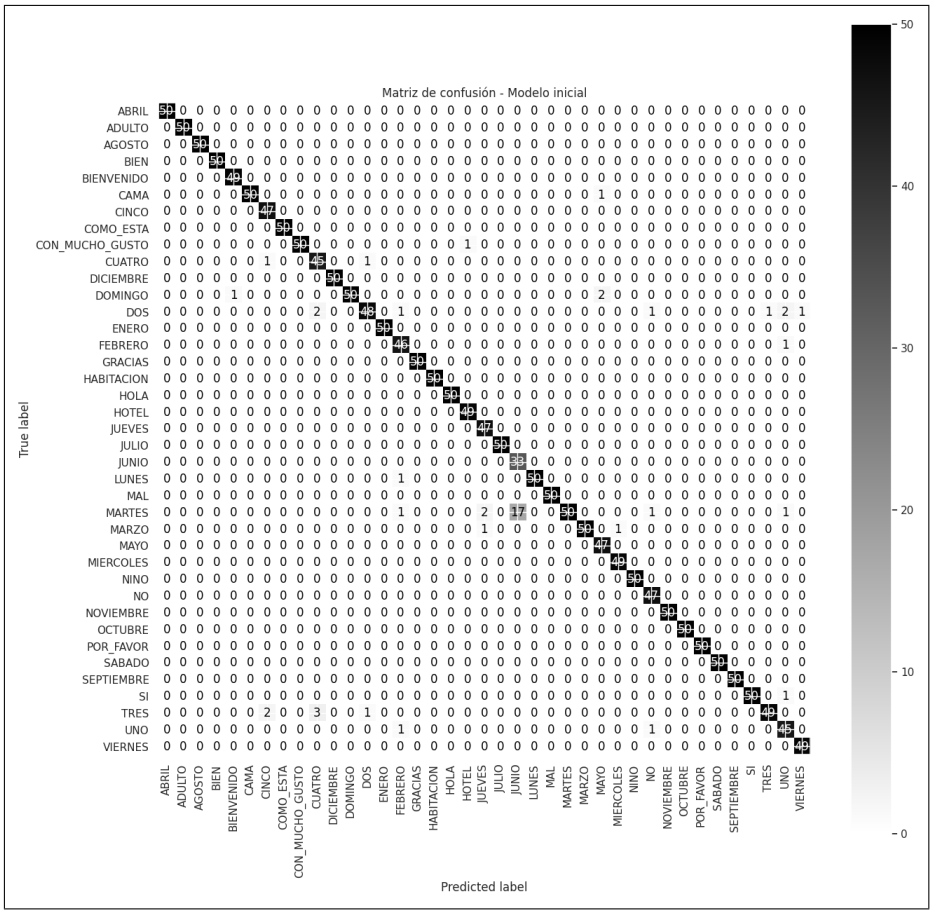


Figura 5: Matriz de confusión Modelo Inicial

A simple vista, en la figura 5 es notable que la mayoría de datos se predicen correctamente y algunas predicciones son erróneas, pero tampoco representan una cantidad significativa y no cambia el hecho de que el modelo tuvo una asertividad alta. De la matriz se puede deducir que:

- Para 25 de las clases (64.10 %), se hace una predicción correcta de las 50 muestras en cada una.
- Para 13 de las clases (33.33 %) se hacen predicciones correctas del 90 % de las muestras.
- Solo un gesto tiene un porcentaje de predicciones correctas del 66 % de la muestra, donde se clasifica erróneamente 17 muestras del gesto en LSC de la palabra **“JUNIO”** y se confunde con el gesto de **“MARTES”** esa misma cantidad de veces.

	Precision	Recall	F1-score	Support
Accuracy			0.97	1950
Macro avg	0.98	0.97	0.97	1950
Weighted avg	0.98	0.97	0.97	1950

Cuadro 3: Métricas Modelo Inicial

Como se evidencia en la tabla 3, estos valores indican que el modelo ha logrado una alta precisión en la clasificación de los gestos en las imágenes, es decir, que rara vez comete errores prediciendo los casos positivos. La precisión promedio es de alrededor del 98 %, lo que sugiere un buen rendimiento en la mayoría de las clases. El modelo parece generalizar bien y tiene un equilibrio entre precisión y sensibilidad, demostrando que tiene buena capacidad para detectar los casos positivos y es confiable, ya que el valor del F1-Score también es alto. Los resultados globales de exactitud, el promedio simple (Macro avg) y ponderado (Weighted avg) son altos e indican que el modelo tiene un rendimiento perfecto en la mayoría de las clases y un buen rendimiento en general con el 98 % de predicciones correctas; por lo tanto, incluso con su sencillez, este modelo es efectivo para clasificación de imágenes de este conjunto de datos.

6.2. Análisis Modelo Inicial con búsqueda de hiperparámetros

Siguiendo la misma estructura utilizada en el modelo inicial, se tuvieron en cuenta los hiperparámetros de tamaño de filtros y cantidad de filtros para ambas capas convolucionales (Conv2D I y Conv2D II), como se describe en la Tabla 4. El tamaño de filtro se ajustó en un rango de 3x3 a 5x5, medidas comunes que se adecúan al uso de imágenes de tamaño moderado (200 x 200 píxeles) y permiten definir el tamaño de recepción para la captura de características y detalles. Además, se ajustó la cantidad de filtros de 32 a 64, teniendo en cuenta que no se trataba de un conjunto de datos demasiado extenso. Esto se realizó con el propósito de optimizar el rendimiento y la eficacia en la captura de características y en la regulación de los datos.

Después de configurar estos hiperparámetros, se aplicó un algoritmo de búsqueda. En este caso, se implementó el método de Random Search, que se encarga de evaluar diferentes combinaciones y seleccionar el resultado más óptimo. Este enfoque resulta eficiente en términos de tiempo y recursos, como también una exploración más amplia gracias a la aleatoriedad.

Busqueda de hiperparámetros			
Capa	**Parámetros**	**Valores**	**Ajuste hiperparámetros**
Conv2D I	Filters	41	Sí
	Kernel size	3x3	Sí
	Activation	ReLu	No
	Input Shape	(200, 200, 3)	No
	Padding	Valid	No
MaxPooling2D I	Pool size	2x2	No
	Stride	2	
Conv2D II	Filters	42	Sí
	Kernel size	3x3	Sí
	Activation	ReLu	No
	Input Shape	(200, 200, 3)	No
	Padding	Valid	No
MaxPooling2D II	Pool size	2x2	No
	Stride	2	
Flatten	NA	NA	No
Dropout	Rate	0.25	No
Dense	Unidad	39	No
	Dimensionalidad		
	Activation	Softmax	No

Cuadro 4: Arquitectura Modelo Inicial

Luego de la búsqueda, se presentan en el cuadro 4 los parámetros óptimos identificados para el conjunto de datos según las especificaciones establecidas. Se asigna un número de filtros de 41 para la capa Conv2D I, con un tamaño de kernel de 3x3. Asimismo, para la capa Conv2D II, se establece un número de filtros de 42 con un tamaño de kernel de 3x3.

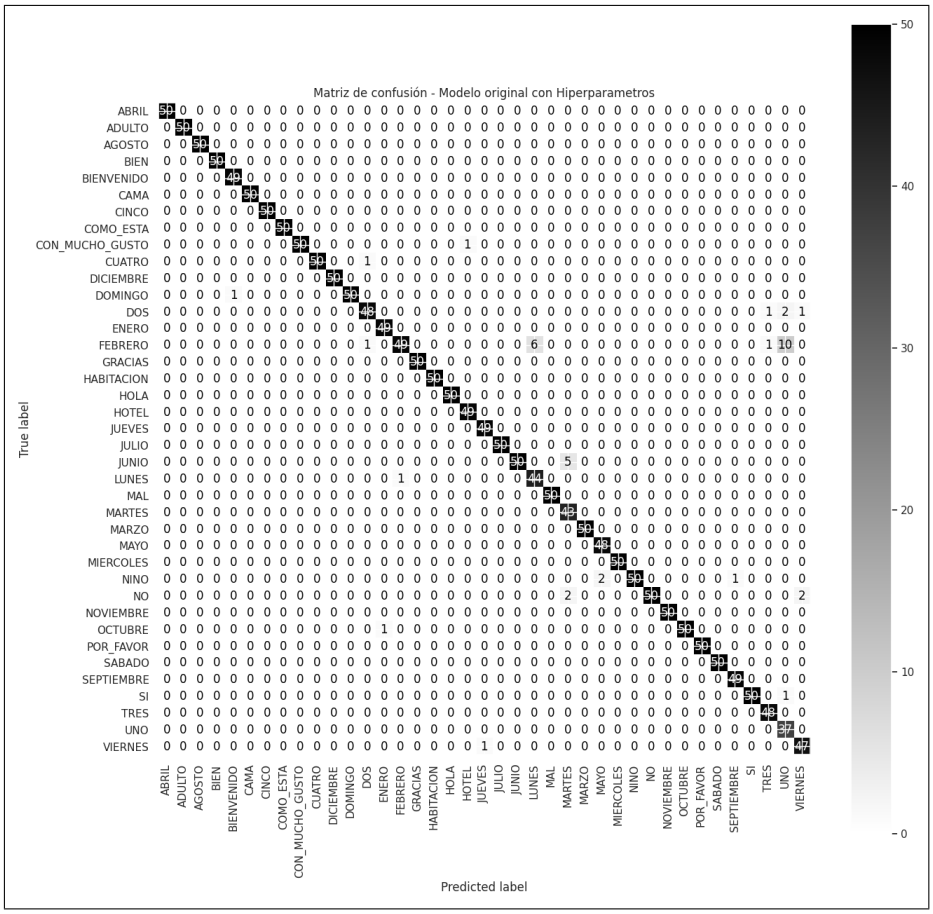


Figura 6: Matriz de confusión Modelo Inicial con búsqueda hiperparámetros

Con base en la matriz de confusión que se muestra en la Figura 6, se puede ver que:

- Para 26 de las clases (66.67 %), se hace una predicción correcta de las 50 muestras en cada clase.
- Para 12 de las clases (30.77 %), se hacen predicciones correctas del 86 % de las muestras.
- Solo una clase tiene un porcentaje de predicciones correctas del 74 % de la muestra, donde se clasifica erróneamente el gesto en LSC de la palabra **"UNO"** y lo confunde mayoritariamente con el gesto de **"FEBRERO"** con 10 desaciertos.
- Otros gestos como **"FEBRERO"** también son clasificados erróneamente en algunos casos y lo confunden con **"LUNES"**.

	Precision	Recall	F1-score	Support
Accuracy			0.98	1950
Macro avg	0.98	0.98	0.98	1950
Weighted avg	0.98	0.98	0.98	1950

Cuadro 5: Métricas Modelo Inicial con hiperparámetros

Conforme se puede apreciar en la tabla 5, el resultado con hiperparámetros no difiere mucho del modelo original 3, pues aunque el promedio simple y ponderado en la precisión de las predicciones positivas no cambia, hay una leve mejora de en la sensibilidad y la puntuación F1. Ambas métricas subieron sus valores de un 97 % a un 98 %, para la exactitud y los promedios. A pesar de que la optimización no fue significativa en comparación a sus primeros resultados, no quita el hecho de que siga siendo una elección óptima para la clasificación de este conjunto de datos, pues demuestra tener un alto rendimiento al igual que su primera versión.

Otro detalle a destacar es como en la matriz de confusión 5 del modelo inicial, notamos casos de rendimiento inferior en la clase **"JUNIO"**, mientras que en la matriz de confusión 6 del modelo inicial con hiperparámetros, la clase **"UNO"** presenta dificultades de clasificación. Estas diferencias se deben a los parámetros definidos y han permitido lograr una ligera mejora en la clasificación de forma general.

6.3. Análisis Modelo Simplificado

Este modelo tiene una arquitectura ligeramente más profunda al incluir más capas convolucionales; sin embargo, es muy similar al modelo inicial. Una estructura con más capas no necesariamente significa un mejor rendimiento, como se puede ver en el siguiente caso:

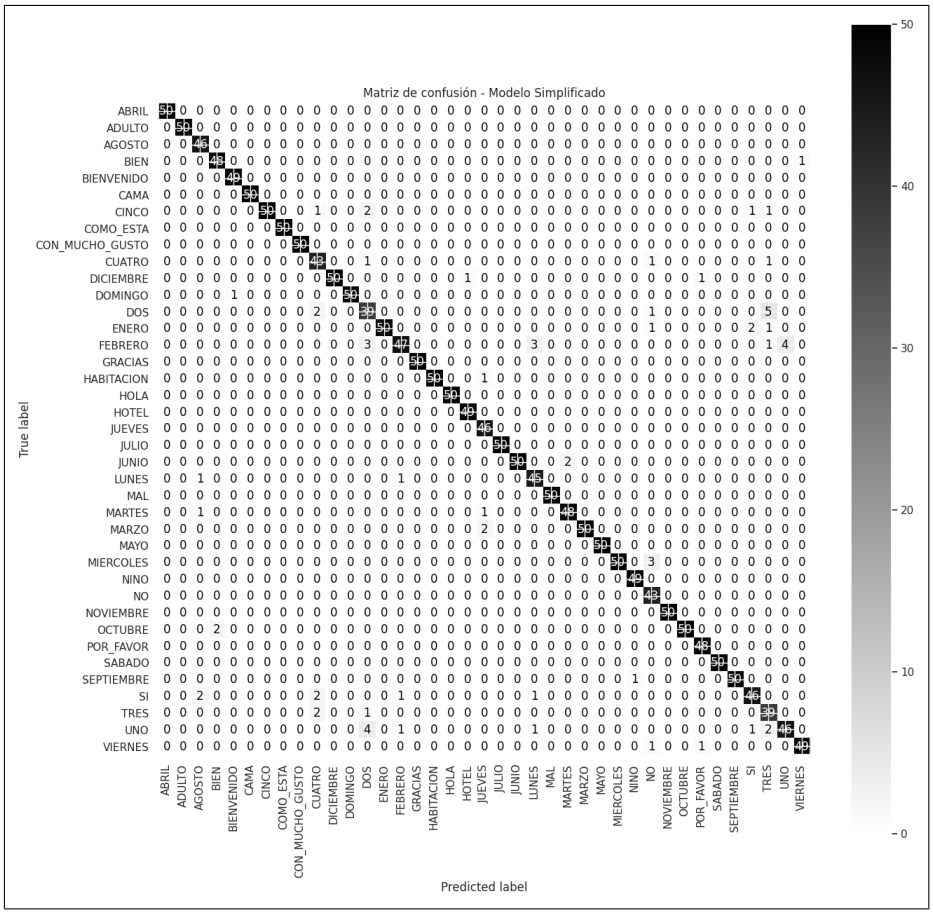


Figura 7: Matriz de confusión Modelo Simplificado

A juzgar por la matriz de confusión de la figura 7, se puede apreciar que:

- Para 22 de las clases (56.41 %), se hace una predicción correcta de las 50 muestras en cada clase.
- Para 15 de las clases (38.46 %), se tienen predicciones correctas que aciertan el 86 % las muestras.
- En dos de las clases, se tiene un porcentaje de predicciones del 78 %, donde se clasifica erróneamente 11 muestras, siendo el gesto en LSC de la palabra **"TRES"** con **"DOS"** mayoritariamente.

	Precision	Recall	F1-score	Support
Accuracy			0.96	1950
Macro avg	0.97	0.96	0.96	1950
Weighted avg	0.97	0.96	0.96	1950

Cuadro 6: Métricas Modelo Simplificado

Como se evidencia en la tabla 6, este modelo tiene resultados muy similares al modelo inicial (figura 3), pues también cuenta con un alto rendimiento, aunque es ligeramente peor a comparación, y aun así, con un alto promedio del 97 % en los datos de prueba en promedio simple y ponderado. Respecto a la sensibilidad se tiene un porcentaje del 96 %. Además, el balance entre precisión y sensibilidad, tiene un valor alto con una puntuación F1 del 96 %. Lo anterior, indica que predice correctamente en la mayoría de los casos positivos. Este modelo demuestra eficacia en la clasificación de imágenes para este conjunto de datos, aunque no se considera como la opción más óptima.

6.4. Análisis Modelo Simplificado con búsqueda de hiperparámetros

Este modelo ha sido configurado utilizando Hyperparameter Tuning mediante la biblioteca Keras Tuner. Se ha habilitado la selección de valores para los filtros en las capas convolucionales en un rango de 32 a 64, y simultáneamente, se puede elegir el tamaño del kernel entre 3 y 5. Estos hiperparámetros son ajustados automáticamente durante el proceso de entrenamiento para encontrar la combinación óptima que maximice el rendimiento del modelo en la tarea específica. A continuación, se muestra la combinación de valores en hiperparametros que obtienen el mejor resultado en exactitud:

Estructura del modelo simplificado			
Capa	**Parámetros**	**Valores**	**Ajuste Hiper-parámetros**
Conv2D I	Filters	36	Si
	Kernel size	5x5	Si
	Input shape	(200, 200, 3)	No
	Padding	Same	No
	Activation	ReLu	No
Conv2D II	Filters	61	Si
	Kernel size	5x5	Si
	Input shape	(200, 200, 3)	No
	Padding	Same	No
	Activation	ReLu	No
MaxPooling2D I	Pool size	2x2	No
Dropout I	Rate	0.25	No
Conv2D III	Filters	63	Si
	Kernel size	3x3	Si
	Input shape	(200, 200, 3)	No
	Padding	Same	No
	Activation	ReLu	No
Conv2D IV	Filters	36	Si
	Kernel size	3x3	Si
	Input shape	(200, 200, 3)	No
	Padding	Same	No
	Activation	ReLu	No
MaxPooling2D II	Pool size	2x2	No
Dropout II	Rate	0.25	No
Flatten	NA	NA	NA
Dense I	Unidad	256	No
	Dimensionalidad		
	Activation	ReLu	
Dropout III	Rate	0.5	No
Dense II	Unidad	39	No
	Dimensionalidad		
	Activation	Softmax	No

Cuadro 7: Arquitectura Modelo Simplificado

Después de la búsqueda, se presentan en el cuadro 7 los parámetros óptimos identificados. Se asigna un número de filtros de 36 para la capa Conv2D I, con un tamaño de kernel de 5x5. Además, para la capa Conv2D II, se establece un número de filtros de 61 con un tamaño de kernel de 5x5, mientras que para la capa Conv2D III se utilizan 63 filtros con un tamaño de kernel de 3x3, y para la capa Conv2D IV se emplean 36 filtros con un tamaño de kernel de 3x3.

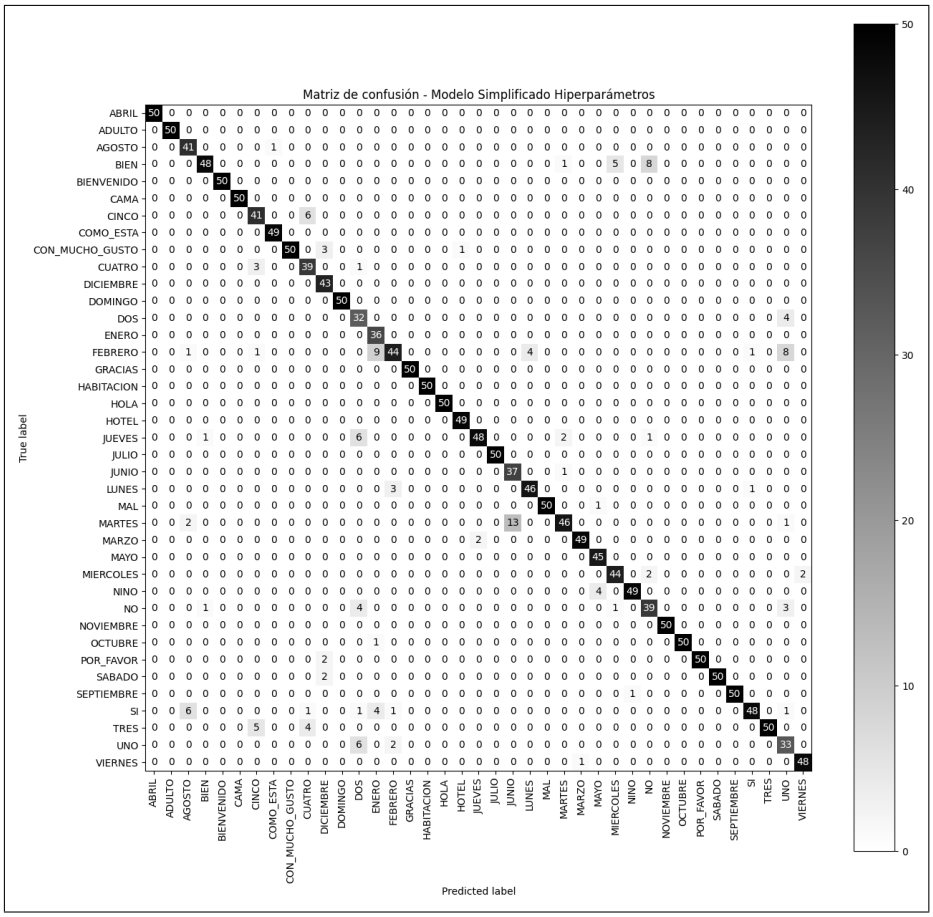


Figura 8: Matriz de confusión Modelo Simplificado con hiperparámetros

Considerando la matriz de confusión que se muestra en la figura anterior 6:

- Para 17 de las clases (43.59 %), se hace una predicción correcta de las 50 muestras en cada clase.
- Para 16 de las clases (41.03 %), se tienen predicciones correctas en el 82 % de las muestras.
- Para 6 de las clases (15.38 %), se tienen predicciones correctas en el 62 % de las muestras.

	Precision	Recall	F1-score	Support
Accuracy			0.93	1950
Macro avg	0.93	0.93	0.92	1950
Weighted avg	0.93	0.93	0.92	1950

Cuadro 8: Métricas Modelo Simplificado con hiperparámetros

Tal como revela la tabla 6, se puede observar que este modelo presenta resultados notablemente inferiores en comparación con todos los modelos previamente

presentados. El promedio de precisión en los datos de prueba, tanto simple como ponderado, es del 93 %. En cuanto a la sensibilidad, se alcanza un porcentaje del 93 %. Además, al considerar el equilibrio entre precisión y sensibilidad, la puntuación F1 se sitúa en el 92 %.

6.5. Análisis Modelo VGG-16

Al ser un modelo preentrenado, usar aprendizaje por transferencia y tener varias capas convolucionales, tiene una arquitectura que le permite capturar mejor detalles y patrones. A continuación se muestra los resultados de esta:

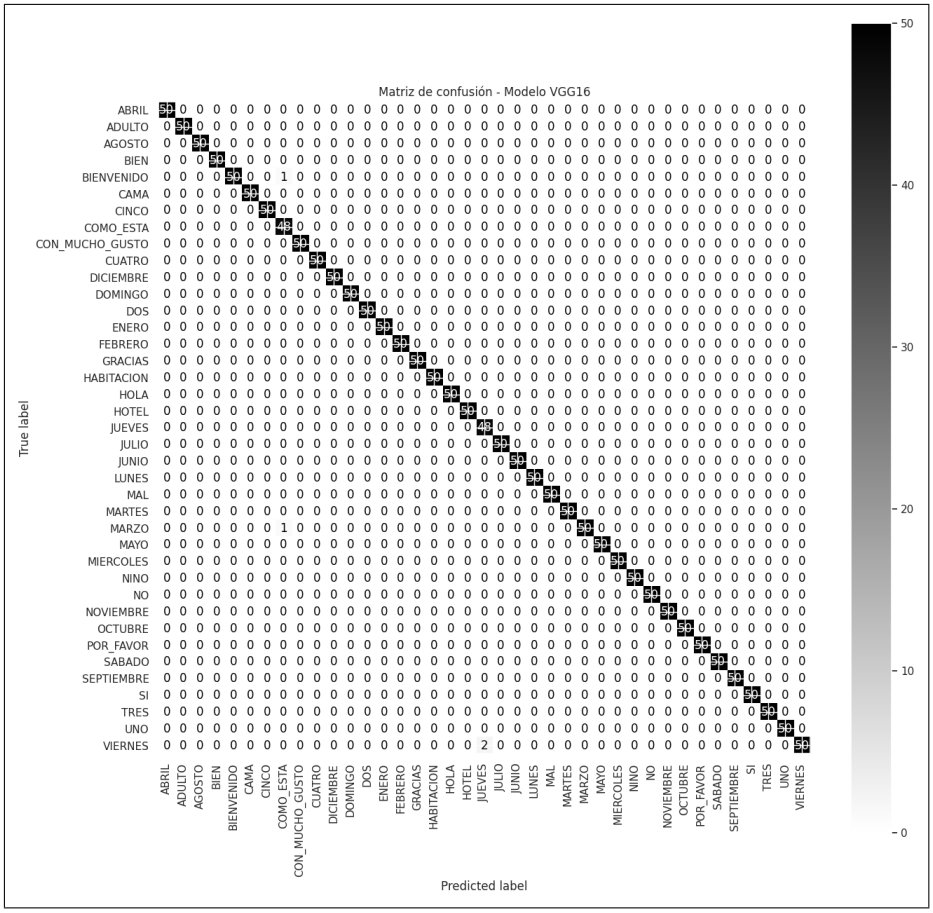


Figura 9: Matriz de confusión Modelo VGG-16

Con base en la matriz de confusión del modelo (Figura 9), se aprecia una notable mejora en la asertividad de predicciones positivas del conjunto de datos, en casi todas las clases se predicen correctamente en las 50 muestras. Analizando en profundidad se tiene lo siguiente:

- Para 37 de las clases (94.87 %), se hacen predicciones correctas de las 50 muestras en cada clase, siendo el más alto de los modelos en este caso.

- Para dos de las clases (5.13 %), se tienen predicciones correctas que aciertan ambas con el 96 % de las muestras.
- En esas dos clases, se clasifican erróneamente los gestos en LSC de la clase que pertenece a **"JUEVES"**, se confunde con **"VIERNES"** y también la palabra **"COMO ESTA"** con **"BIENVENIDO"** o con **"MARZO"**. Sin embargo, son casos muy específicos y con solo dos coincidencias en el error, por lo que es poco significativo.

	Precision	Recall	F1-score	Support
0				
1	1.00	1.00	1.00	50
2	1.00	1.00	1.00	50
3	1.00	1.00	1.00	50
4	0.98	1.00	0.99	50
5	1.00	1.00	1.00	50
6	1.00	1.00	1.00	50
7	1.00	0.96	0.98	50
8	1.00	1.00	1.00	50
9	1.00	1.00	1.00	50
10	1.00	1.00	1.00	50
11	1.00	1.00	1.00	50
12	1.00	1.00	1.00	50
13	1.00	1.00	1.00	50
14	1.00	1.00	1.00	50
15	1.00	1.00	1.00	50
16	1.00	1.00	1.00	50
17	1.00	1.00	1.00	50
18	1.00	1.00	1.00	50
19	1.00	0.96	0.98	50
20	1.00	1.00	1.00	50
21	1.00	1.00	1.00	50
22	1.00	1.00	1.00	50
23	1.00	1.00	1.00	50
24	1.00	1.00	1.00	50
25	0.98	1.00	0.99	50
26	1.00	1.00	1.00	50
27	1.00	1.00	1.00	50
28	1.00	1.00	1.00	50
29	1.00	1.00	1.00	50
30	1.00	1.00	1.00	50
31	1.00	1.00	1.00	50
32	1.00	1.00	1.00	50
33	1.00	1.00	1.00	50
34	1.00	1.00	1.00	50
35	1.00	1.00	1.00	50
36	1.00	1.00	1.00	50
37	1.00	1.00	1.00	50
38	0.96	1.00	0.98	50
Accuracy			1.00	1950
Macro avg	1.00	1.00	1.00	1950
Weighted avg	1.00	1.00	1.00	1950

Cuadro 9: Métricas Modelo VGG-16

En este modelo se emplea transferencia de aprendizaje a partir de un modelo preentrenado, el cual muestra un rendimiento destacado. En la tabla (véase la Tabla 9), se observa una tasa de asertividad del 100 %. Sin embargo, al considerar cifras no redondeadas, se pueden identificar 4 desaciertos en algunas clases (véase la Figura 10), lo que ajustaría la asertividad a un 99.79 %. Destacando con un porcentaje alto para la precisión, la sensibilidad y la puntuación F1, lo que indica que tiene un rendimiento sobresaliente, evidenciando que la implementación de modelos preentrenados con transferencia de aprendizaje es altamente efectiva en este contexto. Aunque los modelos anteriores también tuvieron una alta asertividad en la clasificación de los datos, el resultado del modelo preentrenado es indiscutiblemente mejor.

6.6. Análisis Modelo VGG-16 con búsqueda de hiperparámetros

Ya que el modelo VGG 16 es una arquitectura bastante profunda, para la optimización solo se modificaron las últimas tres capas del modelo, es decir, las capas completamente conectadas. Además, solo se tuvieron en cuenta dos rutas de entrenamiento: en la primera solo añadiendo las últimas capas con 64 y 32 filtros con activación ReLu y la segunda, añadiendo solo capas densas de 64 filtros cada una con la misma activación. Con esto se quería experimentar cómo afectaría el resultado ajustando solo las últimas capas de esta estructura.

Al igual que los modelos anteriores, se utilizó Random Search para la experimentación, pero debido a la cantidad de combinaciones posibles y al costo computacional que eso implica, se hizo solo una búsqueda para establecer las dos rutas que se mencionaron anteriormente.

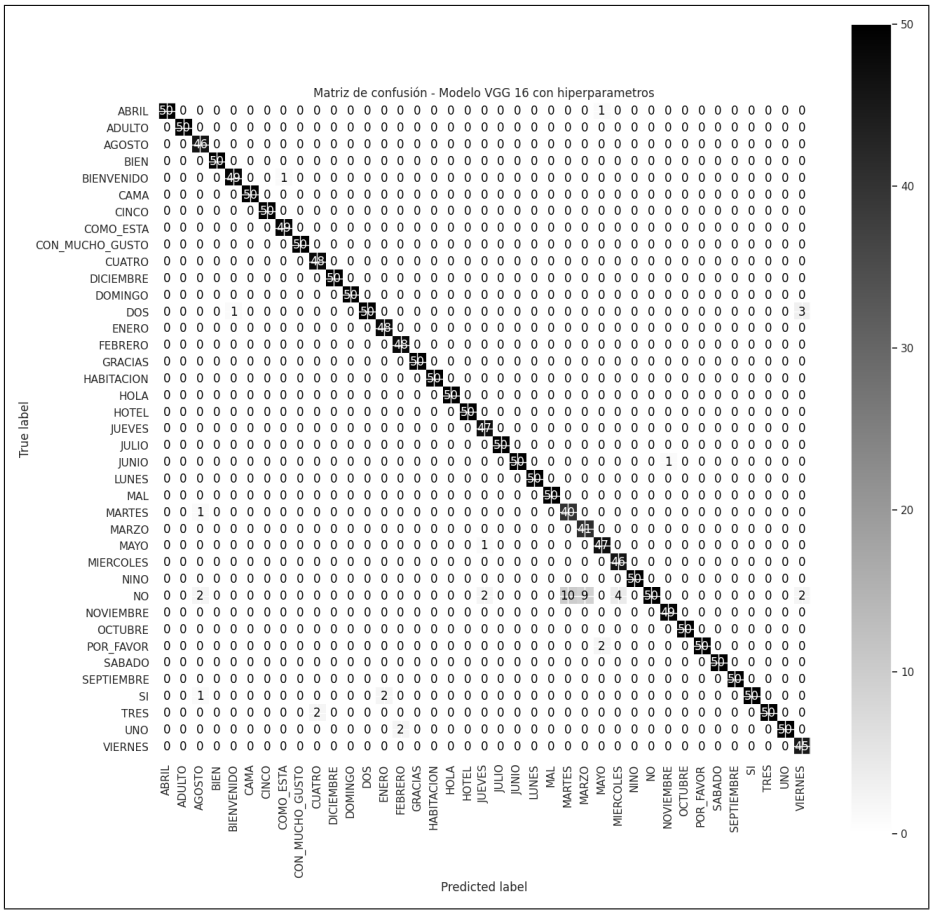


Figura 10: Matriz de confusión Modelo VGG-16 con hiperparámetros

- Para 26 de las clases (66,6 %), se hacen predicciones correctas de las 50 muestras en cada clase, siendo mostrando una reducción de asertividad en comparación al primer resultado del modelo preentrenado.
- Para 11 de las clases (28,2 %), se tienen predicciones correctas que aciertan el 90 % de las muestras.
- En dos de las clases (5,1 %) se tiene un porcentaje de predicciones del 80 % y 82 % respectivamente.
- En estas dos clases se clasifican erróneamente los gestos en LSC de las clases **"MARTES"** y **"MARZO"** que se confunden con **"NO"**.

	Precision	Recall	F1-score	Support
Accuracy			0.98	1950
Macro avg	0.98	0.98	0.98	1950
Weighted avg	0.98	0.98	0.98	1950

Cuadro 10: Métricas Modelo VGG 16 con hiperparámetros

De la misma manera que se demostró antes, aunque los hiperparámetros son cruciales para la optimización del rendimiento del modelo, puede llegar a tener resultados inesperados, como sería empeorar el rendimiento del modelo VGG 16. En este caso, el rendimiento bajó del 99.79 % al 98 % en promedio. Lo mismo ocurre con todas las métricas evaluadas de exactitud, sensibilidad y puntuación F1, las cuales empeoraron en la misma cantidad. Esto no quiere decir que este modelo no sirva, de hecho sigue siendo bastante alto y obtiene un resultado similar al modelo inicial con dos convoluciones, siendo apto para aplicar en este conjunto.

7. Evaluación final

Los tres modelos de redes neuronales convolucionales presentados ofrecen sólidas soluciones para la clasificación de gestos en imágenes, el modelo inicial y simplificado son más flexibles en términos de hiperparámetros, permitiendo la capacidad de ajustar y optimizar el modelo de acuerdo con las necesidades específicas. Estos podrían considerarse como una opción para el prototipo si los recursos computacionales fueran muy limitados.

Los modelos, basados en los respectivos hiperparámetros, demostraron ser menos viables en algunos casos. en el Modelo Inicial, con los hiperparámetros específicos (véase la Tabla 5), se observó una mejora en comparación con el Modelo Inicial estándar (véase la Tabla 3). Sin embargo, en el caso del modelo simplificado con hiperparámetros (véase la Tabla 8) y en estructuras más complejas, como en el Modelo VGG 16 con sus hiperparámetros correspondientes (véase la Tabla 10), se observó una menor precisión en comparación con sus modelos base (véase las Tablas 6 y 9). Estos datos pueden ser el resultado de las limitaciones en la búsqueda de hiperparámetros. Como también a una cantidad insuficiente de iteraciones para la búsqueda de hiperparámetros.

Los modelos con una disminución de rendimiento en comparativa con su versión estándar, podría atribuirse a un espacio de búsqueda reducido; teniendo en cuenta que los rangos definidos para la búsqueda hiperparametros no fueron demasiado extensos, por ejemplo: solo se toca una cantidad de filtros (o kernels) entre 32 y 64 y con un tamaño del kernel entre 3x3 y 5x5. La cantidad de kernels afecta en la diversidad de las características que se extraen y el tamaño de estos influye en la captura de características más grandes o más finas. Por otro lado, en la búsqueda de hiperparámetros se define una cantidad de ensayos para escoger la mejor combinación de esos ensayos, para este proyecto se realizaron tres iteraciones en cada modelo. Es importante tener en cuenta que aumentar el rango de búsqueda de hiperparametros y la cantidad de iteraciones también implica una mayor complejidad carga computacional que afecta el uso de recursos y el tiempo de espera de ejecución.

En última instancia, la elección del modelo depende de los recursos disponibles, los requisitos de precisión y la flexibilidad necesaria para adaptar el prototipo a las necesidades del proyecto. Con base en esto, cualquiera de los tres modelos podría

servir como base para un sistema de identificación de gestos. No obstante, el modelo preentrenado de VGG 16 sería la elección preferida si se prioriza la precisión, ya que este modelo alcanzó el 99.79 % (véase Tabla 9). Por lo tanto, se selecciona este modelo como la propuesta de solución, dado que es capaz de identificar gestos de manera correcta.

8. Despliegue del modelo

Para la creación de este prototipo, se optó por utilizar Streamlit en Python, una plataforma que permite desarrollar aplicaciones web de manera eficiente con modelos de aprendizaje automático.

Por lo que, es necesario instalar algunas dependencias para que el prototipo pueda ser usado, estas son: Streamlit, TensorFlow y Numpy.

La aplicación está diseñada para cargar imágenes en formato JPG, JPEG y PNG, que luego son procesadas y analizadas. La imagen se muestra una vez que ha sido cargada, junto con el nombre del archivo correspondiente.

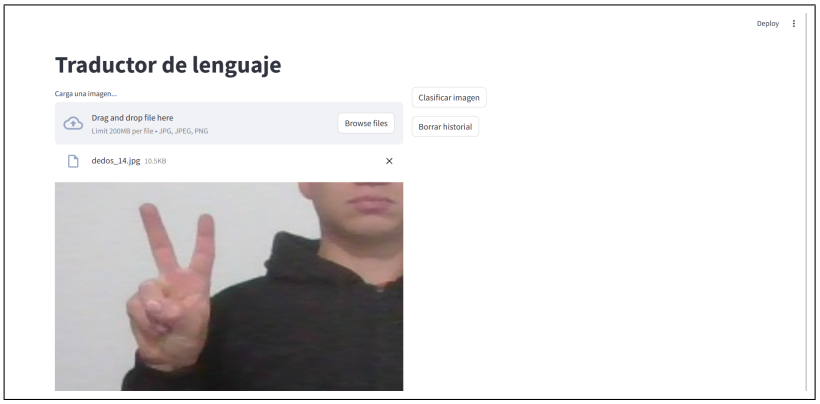


Figura 11: Plantilla inicial e imagen cargada

Al hacer clic en el botón “Clasificar imagen”, la aplicación revela el significado del gesto presente en la imagen.

Figura 12: Clasificación

Además, por cada imagen ingresada, se guarda su significado para generar una frase u oración que el usuario pueda entender con mayor facilidad. Una vez que

la frase u oración esté completa, se puede utilizar el botón “Borrar historial” para eliminar la frase generada y comenzar de nuevo.

A continuación se muestra un ejemplo con las palabras “DOS”, “CAMA” y “POR_FAVOR”:

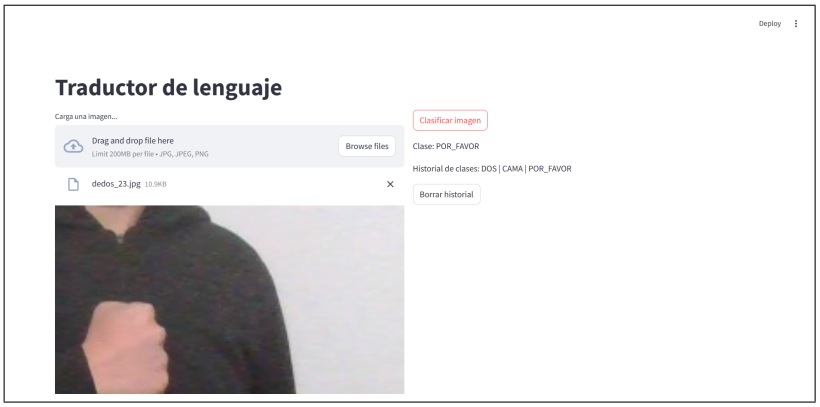


Figura 13: Frase formada

9. Conclusiones

En esta investigación se abordó el problema de comunicación entre las personas con discapacidad sensorial auditiva y el personal de hotel en Colombia. Nuestra propuesta de solución frente a esta situación es la creación de una herramienta con aprendizaje automático programada en Python capaz de identificar gestos manuales en LSC relacionados con el tema del sector hotelero, cuyo objetivo es preparar mejor a estos establecimientos para que brinden una comunicación efectiva con clientes de esa población. Para construir la herramienta en cuestión, se tomó un corpus de imágenes de LSC relacionadas con el ambiente hotelero y se tuvieron en cuenta tres clases de modelos de redes neuronales convolucionales con diferentes niveles de profundidad (modelo inicial, modelo simplificado, modelo preentrenado) con la intención de encontrar la opción más asertiva en la identificación de los gestos. Teniendo en cuenta métricas como: la precisión, la sensibilidad (recall), el F1-Score, la exactitud y la aplicación de búsqueda con hiperparámetros para un análisis más detallado del resultado de los entrenamientos de los modelos propuestos.

Los resultados obtenidos de los modelos son muy similares entre sí, ya que todos alcanzan una tasa de acierto superior al 90 % en la predicción de datos, evidenciando un rendimiento general elevado. Al final del entrenamiento se obtuvo que: el modelo inicial tiene una exactitud del 97 % y un 98 % con hiperparámetros, el modelo simplificado con un 96 % y un 93 % con hiperparámetros, el modelo preentrenado de VGG 16 con un 99.79 % y un 98 % con hiperparámetros.

Como se indicó en la sección de análisis, se anticipaba que el modelo VGG 16 sobresaldría entre los demás debido a su arquitectura más profunda y al empleo

de transferencia de aprendizaje. Esta característica implica que su capacidad para extraer y detallar características es superior, convirtiéndolo en la opción preferida para el aprendizaje de características abstractas y complejas. La transferencia de aprendizaje de redes convolucionales, basándose en los resultados obtenidos (véase la Tabla 9), se implementó con éxito en este proyecto. No obstante, en la búsqueda de hiperparámetros del modelo VGG 16 resulta ser compleja debido a los tiempos involucrados. Junto con esto, la reducción de posibilidades de tiempos considerables conllevó que no se obtuviera una mejora en el rendimiento del modelo.

La elección de utilizar el modelo VGG 16 en el prototipo se basó en su capacidad para lograr una asertividad del 99.79 %. La interfaz simple y fácil de usar permite a los usuarios comprender de manera sencilla el significado de los gestos representados en las imágenes que desean cargar.

10. Trabajos futuros

Este proyecto está enfocado en el reconocimiento de imágenes cargadas en una interfaz, y a su vez, la traducción de estas. La implementación actual ya abre posibilidades para la expansión del proyecto, ya que el LSC no se limita únicamente a gestos estáticos, sino que incluye movimientos manuales que transmiten significados diversos. Con miras al futuro, se contempla el desarrollo de un prototipo de reconocimiento de gestos en tiempo real, capaz de mostrar traducciones exactas en texto, facilitando la traducción de gestos y la comunicación entre los clientes y el personal de hotel que no esté capacitado en LSC.

Adicionalmente, se propone la creación de un corpus de videos como una mejora significativa. Este corpus permitiría un entrenamiento más preciso del modelo, agilizando la interpretación de gestos y mejorando la eficacia de la herramienta. Adaptar el modelo a diversos contextos más allá del ámbito hotelero se presenta como una iniciativa prometedora para ampliar la aplicabilidad de la herramienta. Esta expansión, que abarca a diversos sectores, incluyendo acciones cotidianas, educación, salud y negocios, no solo promovería la inclusividad al satisfacer las necesidades de una audiencia más diversa, sino que también se alinea con la amplitud del LSC.

Referencias

[1] T. B. Méndez, "Rehabilitación auditiva en colombia, un problema de salud pública con poca escucha," in *Periódico UNAL / Salud*, 2023.

[2] C. Mariño, "Barreras de la comunidad sorda en bogotá," El Espectador, May 2022. [Online]. Available: https://www.elespectador.com/bogota/barreras-de-la-comunidad-sorda-en-bogota-y-donde-aprender-lengua-de-senas-en-colombi

[3] L. República, "¿por qué las personas sordas tienen dificultades para leer y escribir el español?" https://larepublica.pe/sociedad/2022/11/30/por-que-las-personas-sordas-tienen-dificultades-para-leer-y-escribir-el-espanol, Nov 2022.

[4] A. Oliverio, "¿es posible aprender a leer sin oír?" *The Conversation*, 2020. [Online]. Available: https://theconversation.com/es-posible-aprender-a-leer-sin-oir-142693

[5] C. I. D. L. H. Montenegro, *Turismo Accesible Análisis de los servicios hoteleros y gastronómicos de la ciudad de Barranquilla*. Educosta, 2017.

[6] S. Subburaj and S. Murugavalli, "Survey on sign language recognition in context of vision-based and deep learning," *Measurement: Sensors*, vol. 23, p. 100385, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2665917422000198

[7] W. Li, H. Pu, and R. Wang, "Sign language recognition based on computer vision," in *2021 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, 2021, pp. 919–922.

[8] L. Sandoval, "Algoritmos de aprendizaje automático para análisis y predicción de datos," *Revista Tecnológica*, vol. 11, no. 22, pp. 1–10, Enero-diciembre 2018. [Online]. Available: http://www.redicces.org.sv/jspui/bitstream/10972/3626/1/Art6_RT2018.pdf

[9] B. Borrella, "Introducción a la visión artificial: procesos y aplicaciones," Universidad Complutense de Madrid. Departamento de Análisis y matemática, Tech. Rep., 2021. [Online]. Available: https://eprints.ucm.es/id/eprint/74914/1/beatriz_borrella_introduction.pdf

[10] D. Calvo, "Clasificación de red neuronal según la tipología de red," Blog post, Julio 2017. [Online]. Available: https://www.diegocalvo.es/clasificacion-de-redes-neuronales-artificiales/

[11] (s. f.) Redes neuronales profundas preentrenadas - matlab & simulink - mathworks españa. [Online]. Available: https://es.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html

[12] ¿qué son las redes neuronales convolucionales? IBM. IBM in Deutschland, Österreich und der Schweiz. [Online]. Available: https://www.ibm.com/es-es/topics/convolutional-neural-networks

[13] Unir. (2023, noviembre 2) ¿qué es el transfer learning y qué ventajas tiene? [Online]. Available: https://www.unir.net/ingenieria/revista/transfer-learning/#:~:text=El%20transfer%20learning%20consiste%20en,se%20puede%20compartir%20entre%20ellos

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[15] ICHIPRO. (2021, septiembre 24) ¿qué es vgg16? — introducción a vgg16. [Online]. Available: https://ichi.pro/es/que-es-vgg16-introduccion-a-vgg16-267001881294357

[16] S. Tammina, "Transfer learning using vgg-16 with deep convolutional neural network for classifying images," *International Journal of Scientific and Research Publications (IJSRP)*, vol. 9, no. 10, 2019.

[17] T. S. S and M. R. I, "A extensive survey on sign language recognition methods," in *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)*, 2023, pp. 613–619.

[18] K. Suri and R. Gupta, "Continuous sign language recognition from wearable imus using deep capsule networks and game theory," *Computers Electrical Engineering*, vol. 78, pp. 493–503, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790619301508

[19] R. Elakkiya, K. Selvamani, and S. Kanimozhi, "A framework for recognizing and segmenting sign language gestures from continuous video sequence using boosted learning algorithm," in *2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, 2014, pp. 498–503.

[20] J. Pu, W. Zhou, and H. Li, "Iterative alignment network for continuous sign language recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4160–4169.

[21] LSC - Conjunto de datos sector hotelero, "LSC - Conjunto de datos sector hotelero," https://www.kaggle.com/datasets/jimmyalejandro/lsc-conjunto-de-datos-sector-hotelero?resource=download, 2022, accedido el 14 de marzo de 2022.

[22] Yassineghouzam. (2017, 8) Introduction to cnn keras - 0.997 (top 6 %). Kaggle. [Online]. Available: https://www.kaggle.com/code/yassineghouzam/introduction-to-cnn-keras-0-997-top-6#Introduction-to-CNN-Keras---Acc-0.997-(top-8%25)