

**MODELO DE RECONOCIMIENTO PARA LA LENGUA DE SEÑAS:
APROXIMACIÓN COMPARATIVA ENTRE MÉTODOS DE
RECONOCIMIENTO DE PATRONES POR INTELIGENCIA
ARTIFICIAL.**

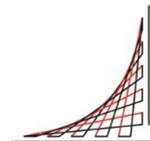
Simon Felipe Corredor Camargo

Tutor(es)

**PhD Alvaro David Orjuela Cañon
PhD Oscar Julian Perdomo Charry**



**Universidad del
Rosario**



**ESCUELA
COLOMBIANA
DE INGENIERÍA
JULIO GARAVITO**

**UNIVERSIDAD DEL ROSARIO
ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO
PROGRAMA DE INGENIERÍA BIOMÉDICA
BOGOTÁ D.C
2022**

**MODELO DE RECONOCIMIENTO PARA LA LENGUA DE SEÑAS:
APROXIMACIÓN COMPARATIVA ENTRE MÉTODOS DE
RECONOCIMIENTO DE PATRONES POR INTELIGENCIA
ARTIFICIAL.**

Simon Felipe Corredor Camargo

Trabajo final de maestría presentado como requisito para optar al título de:
Magister en Ingeniería biomédica

**UNIVERSIDAD DEL ROSARIO
ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO
PROGRAMA DE INGENIERÍA BIOMÉDICA
BOGOTÁ D.C
2022**

AGRADECIMIENTOS

Este trabajo es esfuerzo en el que, directa o indirectamente, participaron distintas personas opinando, corrigiendo, teniéndome paciencia, dando ánimo, acompañando en los momentos de crisis y en los momentos de felicidad. En especial, agradezco a mi familia por siempre ser el motor de cada uno de mis logros. Particularmente a mi hermano que fue la mayor motivación para trabajar en la temática de mi tesis. Agradezco a mis tutores por guiarme durante todo este proceso, y finalmente a mis amigos en especial a Luisa y Luis Felipe por apoyarme y nunca dejarme rendir.

RESUMEN

La lengua de señas es la herramienta de comunicación más utilizadas entre la comunidad de personas con discapacidad auditiva, debido a que permite a sus usuarios comunicarse mediante gestos y movimientos. Aun así, en Colombia y en otros países como India y Estados Unidos se evidencia un reto ante la educación, estandarización y enseñanza de esta lengua, como por ejemplo de cada una de sus variaciones entre zonas geográficas y culturales. Es por esto que metodologías que permitan automatizar el proceso de enseñanza y comunicación de los usuarios de esta lengua ya bien sean sordos u oyentes, son de alta relevancia para lograr la inclusión de las personas sordas o con algún tipo de discapacidad auditiva dentro de un contexto educativo y social. Es así como este trabajo busca estudiar alternativas como algoritmos basados en redes neuronales y aprendizaje automático, con el objetivo de generar un modelo inteligente que reconozca y clasifique diferentes señas del abecedario de la Lengua Americana de Señas (ASL). Todo lo anterior se hará entrenando y validando tres modelos ya utilizados en otros problemas de clasificación de imágenes basados en Redes Neuronales Convolucionales (CNN) a los cuales se explorara sistemáticamente ajustes en su estructura e hiper-parámetros para buscar el modelo que mejor se adapte a la correcta clasificación de cada una de los 27 tipos de imágenes parte de las señas del abecedario de la ASL.

Palabras Clave - Redes neuronales, Deep Learning, Convolutional Neural Networks (CNN), Lengua de Señas

ABSTRACT

The sign language is the communication tool that is most used within the hearing-impaired people community, as it allows their users to communicate through gestures and movements. Even though, in Colombia and some other countries in the world as India and the U.S, the challenge with education, standardization and teaching of this language is evident, an example of this are the multiple variations on these languages between the different cultures and geographic zones. For this reason, the methodologies that allows the automatization of the teaching and communication process within the users of this language (even if they are hearing impaired or not), are relevant to accomplish the inclusion within a social and educational context for the deaf people and those with any type of hear impairing. In this order this investigation aims to study alternatives as algorithms based on Neural Networks and Machine Learning, to generate a model that can recognize and classify different hand gestures part of the alphabet from the American Sign Language (ASL). All the mentioned before will be done training and validating three initial models based on Convolutional Neuronal Networks (CNN) which will be explored systematically with adjustments on structure and hyper-parameters to identify the model structure that adapts the better to the appropriate classification of each of the 27 types of images part of the signs on the ASL alphabet.

Palabras Clave - Neuronal Networks, Deep Learning, Convolutional Neural Networks (CNN), Sign Language

TABLA DE CONTENIDOS

RESUMEN	1
ABSTRACT	2
1. INTRODUCCIÓN	7
2. PROBLEMA DE INVESTIGACIÓN	9
2.1. Antecedentes	9
2.2. Formulación de la pregunta	10
3. DESCRIPCIÓN DEL PROBLEMA	11
4. JUSTIFICACIÓN	13
5. OBJETIVOS	16
5.1. General	16
5.2. Específicos	16
6. MARCO TEÓRICO	17
7. METODOLOGÍA	21
7.1. Tipo de investigación a realizar	21
7.2. Fases de la investigación	21
7.3. Técnica usada para la recolección y procesamiento de la información	21
7.4. Carga y preprocesamiento de imágenes	24
7.5. Definición y Entrenamiento de los modelos	25
7.6. Análisis y conclusiones de los resultados	28
8. RESULTADOS	30
8.1. Resultados entrenamiento Conjunto de Datos A	30
8.1.1. Resultados exploración de parámetros	30
8.1.2. Resultados de entrenamiento de los modelos	33
8.1.3. Evaluación de la predicción con modelos entrenados	37
8.2. Resultados entrenamiento Conjunto de Datos B	38
8.2.1. Resultados exploración de parámetros	38
8.2.2. Resultados de entrenamiento de los modelos	41
8.2.3. Evaluación de la predicción con modelos entrenados	44
9. DISCUSIÓN	46
9.1. Conjunto entrenamiento Base A	46
9.2. Conjunto entrenamiento Base B	47

10.RECOMENDACIONES Y TRABAJOS FUTUROS	50
11.CONCLUSIONES	51

LISTA DE TABLAS

6.1.	Resumen de los sistemas de reconocimiento de la Lengua de señas Americana (ASL) según revisión sistemática de literatura por Wadhwan y Kumar [3].	20
6.2.	Resumen de los sistemas de reconocimiento de la Lengua de señas India (ISL) según revisión sistemática de literatura por Wadhwan y Kumar [3].	20
6.3.	Resumen de los sistemas de reconocimiento de la Lengua de Señas Mexicana LSM según revisión sistemática de literatura por Wadhwan y Kumar [3].	20
7.1.	Distribución de imágenes y etiquetas para las bases de datos utilizadas.	22
7.2.	Variación de Hiperparámetros utilizada para la exploración en cada modelo.	27
7.3.	Variables para la evaluación de los modelos basados Deep Learning y otros modelos de redes neuronales	29
8.1.	Mejores hiper hiper-parámetros encontrados durante Exploración Conjunto de imágenes Base A.	33
8.2.	Evaluación de la predicción de los modelos entrenados a partir de Base A en la Base A.	37
8.3.	Evaluación de la predicción de los modelos entrenados a partir de Base A en Base B.	38
8.4.	Mejores hiper hiper-parámetros encontrados durante Exploración Conjunto de imágenes Base B.	40
8.5.	Evaluación de la predicción de los modelos entrenados a partir de Base B en la Base B.	44
8.6.	Evaluación de la predicción de los modelos entrenados a partir de Base B en la Base A.	45
9.1.	Variables con mejor desempeño dentro de la comparación de modificaciones en arquitectura e hiper-parámetros descritos en cada característica . .	48

LISTA DE FIGURAS

3.1. Comparación del abecedario de la Lengua de Señas Colombiana LSC y la Lengua de Señas Americana ASL	12
4.1. Distribución territorial de la población sorda colombiana, obtenido de [20].	14
4.2. Distribución por edades de la población sorda colombiana, obtenido de [20].	14
6.1. Estructura de una ANN, modificado de [24].	18
6.2. Estructura de una CNN, modificado de [25].	19
7.1. Diagrama general de actividades	22
7.2. Muestra del conjunto de imágenes Base A	23
7.3. Muestra del conjunto de imágenes Base B	23
7.4. Diagrama del proceso de exploración y evaluación de las combinaciones de modelos de Redes neuronales	28
8.1. Resultados de la exploración de hiper-parámetros para la modelo utilizando ResNet50V2 Base A (izquierda) y MobileNetV2 Base A (derecha) .	31
8.2. Resultados de la exploración de hiper-parámetros para la modelo utilizando InceptionResNetV2 Base A	32
8.3. Comportamiento durante el entrenamiento redes basadas en ResNet50V2 Base A	34
8.4. Comportamiento durante el entrenamiento redes basadas en MobileNetV2	35
8.5. Comportamiento durante el entrenamiento redes basadas en Inception-ResNetV2	36
8.6. Resultados de la exploración de hiper-parámetros para la modelo utilizando ResNet50V2 Base B (izquierda) y MobileNetV2 Base B (derecha) .	39
8.7. Resultados de la exploración de hiperparametros para la modelo utilizando InceptionResNetV2 Base B	40
8.8. Comportamiento durante el entrenamiento redes basadas en ResNet50V2 Base B	41
8.9. Comportamiento durante el entrenamiento redes basadas en MobileNetV2 Base B	42
8.10. Comportamiento durante el entrenamiento redes basadas en Inception-ResNetV2 Base B	43

1. INTRODUCCIÓN

La diversidad lingüística es algo característico de los seres humanos frente a otras especies. Los humanos han desarrollado una capacidad única para comunicarse a través de diferentes formas. Los medios por los cuales se comunican, pueden llegar a ser tan exclusivos y particulares para una comunidad que resultan inentendibles para otras. El concepto principal de este documento es la lengua y debe entenderse como el modo de expresión propio de una comunidad. Es el conjunto o sistema de formas o signos orales y escritos que les sirven a las personas de una misma comunidad para comunicarse [1]. El lenguaje, por otro lado, puede entenderse como la capacidad humana que permite conformar el pensamiento. Es decir, es la capacidad innata y abstracta de los seres humanos para comunicarse a través de diferentes lenguas.

En la mayoría de los contextos humanos, la comunicación de las diferentes comunidades se basa en las lenguas fundadas en signos orales. Sin embargo, no todas las comunidades humanas se comunican mediante lenguas habladas. Personas con algún tipo de discapacidad han desarrollado lenguas propias para comunicarse con su comunidad y con el resto del entorno. En el caso de personas con discapacidades auditivas, dentro de las que se encuentran las personas sordas, estas lenguas usan signos y señales visuales para reemplazar la comunicación oral y auditiva. Cada una de estas lenguas se conoce como una Lengua de Señas (LS).

Uno de los focos de investigación en tecnología para el aprendizaje de LS son los sistemas de reconocimiento automático de gestos y movimientos[2]. Estos sistemas de reconocimiento usualmente tienen como objetivo identificar patrones de movimiento y gestualización para poder asignar el significado de cada conjunto de movimientos y relacionarlos con las señas de una palabra o expresión específica [3]. Estas tecnologías buscan acercarse a una interacción humano computadora, donde mediante diferentes técnicas de programación un sistema automático pueda generar un mecanismo de comunicación entre personas oyentes y sordas o viceversa. Adicionalmente, estas tecnologías pueden ser usadas como herramientas educativas útiles para el aprendizaje de esta lengua.

A nivel mundial varios artículos de investigación han propuesto sistemas de reconocimiento aplicado a LS usando diferentes tecnologías. Un ejemplo de estos son guantes que usando sensores permiten identificar el movimiento de la mano [4]. Sin embargo, este tipo de tecnología puede resultar incómoda para el usuario, ya que requiere múltiples conexiones a elementos electrónicos propios del dispositivo y puede obstaculizar el movimiento natural de la seña [2, 5]. Es por esto que los métodos que usan técnicas y algoritmos de análisis de imágenes o vídeos son preferidos, dado que no son invasivos con los usuarios y permiten identificar los gestos y movimientos mediante algoritmos de diferentes tipos [2, 6].

Dentro de los diferentes trabajos de la literatura actual que investigan las tecnologías de reconocimiento por captura de imágenes o vídeos, se identifican tres parámetros principales: (i) el tipo o variación de LS en el cual se desenvuelve cada investigación, (ii) el modo de adquisición de imágenes o vídeos que se utilizó y (iii) el tipo de técnica o

algoritmo implementado.

En cuanto a los tipos de LS se ve una mayor frecuencia en los trabajos que usan como base la LS Americana (ASL, del inglés *American Sign Language*) seguido por la LS India (ISL, del inglés *Indian Sign Language*) y otras LS de otros países del mundo tales como China y Arabia, y son pocos los trabajos que se ven relacionados con estas tecnologías aplicadas a la LS Colombiana (LSC). En términos de dispositivos para la adquisición de imágenes usados en los diferentes trabajos previos, predominan las cámaras de dispositivos móviles seguidas por dispositivos de captura de movimiento como el Kinect de Microsoft y en menor proporción dispositivos como guantes y brazaletes con sensores de movimiento [3].

Por último, las técnicas y algoritmos usados para el análisis de imágenes en estos métodos están usualmente clasificados en tres categorías. En primer lugar, están las que usan la extracción de características y basan su clasificación exclusivamente en el análisis de estas en cada imagen o vídeo. Algunas características son las distancias y vectores de movimiento, el color y bordes en las imágenes, además análisis de movimiento por marcadores óseos entre otros [7]. Por otro lado, están las metodologías que utilizan algoritmos de auto aprendizaje tales como las redes neuronales artificiales (ANN, del inglés *Artificial Neural Networks*) que son comúnmente utilizadas en diferentes aplicaciones tales como análisis de imágenes y reconocimiento de patrones, análisis bioquímico, además del diseño de medicamentos entre otras áreas en la salud [8]. Finalmente, se encuentran las metodologías misceláneas y mixtas que usan una mezcla del análisis de características y aprendizaje automático, unido con otras técnicas para la clasificación de cada una de las imágenes o vídeos [9].

En Colombia son pocos los artículos que exploren este tipo de tecnologías. Además, hay un evidente falta de literatura que soporte los sistemas de reconocimiento en la LSC y la gran oportunidad que ofrecen métodos de clasificación como las redes neuronales convolucionales (CNN, del inglés *Convolutional Neural Networks*) y otros también con estructuras de redes neuronales simples, para la clasificación automática de este tipo de lengua. Este trabajo pretende desarrollar un algoritmo y sistema que procese y clasifique señas estáticas. Lo anterior se realizará inicialmente sobre una base de datos de ASL para encontrar un modelo apropiado y explorar en futuros trabajos su aplicación en la LSC y así determinar la mejor configuración de arreglos de redes neuronales. Se tendrán como principales diseños tres modelos usados comúnmente en problemas de clasificación de imágenes que están basados en CNN, a los cuales se harán variaciones en sus hiper-parámetros y estructura que serán explorados de manera experimental. De esta manera, se busca soportar la amplia necesidad de tecnología que ayude a los procesos de aprendizaje de la lengua de señas, para que a futuro dicho método pueda ser utilizado como parte de instrumentos de comunicación de la mayoría de la población sorda colombiana.

2. PROBLEMA DE INVESTIGACIÓN

2.1. Antecedentes

En primer lugar, y de acuerdo con la revisión de literatura por parte de Wahdawan y Kumar [3] además de la realizada por Oudah y Javaan [6] en relación con este tipo de tecnologías de clasificación de LS, los algoritmos que usan ANN son los que han reportado mayor exactitud. Soportados por distintas técnicas de extracción de características, pueden ser más eficaces, así esto implique en algunos casos, un gasto computacional mayor. Es así como existen metodologías como la usada por Cui y Weng para ASL, en los que hace un énfasis en la extracción de características como el movimiento de la mano y su forma, para después utilizar un algoritmo de árboles de decisión para la clasificación, logrando para un total de 28 señas una exactitud del 93.2 % utilizando cámaras regulares como método de adquisición [10]. Además, se tienen trabajos recientes como el de Quesada y Lope donde utilizaron cámaras Intel RealSense especializadas en el seguimiento de movimientos, combinadas con una técnica de clasificación basado en Máquinas de Soporte Vectorial (SVM, del inglés Support Vector Machines) para hacer la clasificación de las letras del abecedario del ASL obteniendo en la mayoría de las predicciones más del 90 % de exactitud en el proceso de clasificación [11].

En los últimos años se ha visto una tendencia a implementar sistemas de adquisición de imágenes más sencillos y enfocar en el gasto computacional en la clasificación, mediante redes neuronales robustas y complejas. Estructuras enfocadas hacia el aprendizaje profundo o *Deep Learning*, son arquitecturas de ANN que han mostrado efectividad y oportunidades para modelar, representar y aprender de datos complejos y de recursos diversos [12]. Un ejemplo de este tipo de trabajos fue el realizado por Mustafa [13] para la Lengua de Señas Árabe. Allí implementaron y compararon varios modelos junto a arquitecturas de clasificadores, encontrando que la CNN tienen una mejor precisión y exactitud frente a otros métodos frecuentemente utilizados en el área de reconocimiento de imágenes y movimientos. Esto se ve secundado por trabajos como el de Nakjai y Katanyukul [14], que se centran en el uso de CNN para plantear un sistema para el reconocimiento de señas de la Lengua de Señas Tailandesa, logrando una exactitud del 98.82 % en la clasificación de señas del alfabeto de dicha lengua.

De manera similar pero no con el mismo volumen de investigaciones, en Colombia se han estudiado algunas aproximaciones similares a las antes descritas, con el objetivo de aportar y contribuir a la automatización de la enseñanza de la LSC durante los últimos 5 años. Estos están enfocados en la clasificación de señas estáticas y dinámicas en diferentes escenarios y con diferentes configuraciones de redes neuronales profundas. Aun así, comparten la misma limitante que es la falta de un repositorio robusto de LSC que permita hacer comparaciones entre los resultados de los métodos usados en investigaciones, ya que todos estos algoritmos se entrenan y funcionan ante bases de datos creadas por cada autor [15].

Uno de los métodos explorados que se encuentran dentro de la limitada literatura son algoritmos de aprendizaje automático mediante cadenas de Markov [16]. Sin embargo, su evaluación se basa en métricas de perspectiva del usuario, lo cual hace poco comparable. Adicionalmente, son pocos los trabajos que comparan diferentes métodos de procesamiento y clasificación en un contexto colombiano y aplicado al LSC. López [17], en su trabajo hizo una comparación entre diferentes combinaciones de 3 algoritmos descriptores y 4 de clasificación sobre un total de 7 señas sin incluir clasificadores basados en CNN. El autor encontró que las máquinas de soporte vectorial son uno de los métodos eficaces de clasificación. Uno de los trabajos más recientes y con mejores resultados es el presentado en 2020 por Ortiz-Farfan y Camargo Mendoza [15]. Basado en un modelo de CNN, los autores hicieron varios ajustes en las capas de redes neuronales de la estructura inicial, y a través de la modificación de los hiper-parámetros que describen la red, se logró encontrar una exactitud de más del 98%. Sin embargo, en la ejecución del modelo completo ante nuevos datos solo se llegó a el reconocimiento de 15 de un total de las 22 señas dinámicas parte de la LSC que fueron captadas por cámara en formato de video.

2.2. Formulación de la pregunta

Teniendo en cuenta el panorama antes descrito se define como pregunta de investigación:

¿Cuáles son las características en términos de estructura e hiper-parámetros para desarrollar un algoritmo basado en modelos de Redes Neuronales Convolucionales, que permita clasificar automáticamente, imágenes de las señas parte del abecedario de Lengua de señas americana de mejor manera?

3. DESCRIPCIÓN DEL PROBLEMA

En primer lugar, se puede definir el problema a través de los trabajos revisados, en los que se evidencia una tendencia por estudios e investigaciones enfocadas en el desarrollo de un solo modelo de CNN y como al hacer ciertos ajustes se obtienen mejores resultados de clasificación usando una base de datos específica. Sin embargo no hay gran número de trabajos en los que desarrollen de manera permanente un marco comparativo que permita identificar el mejor clasificador entre varios modelos desarrollados basados en redes previamente creadas. Sumado a esto la mayoría de las investigaciones que usan CNN, se desarrollan en diferentes locaciones geográficas y usan diferentes bases de datos como fuente para entrenar cada uno de los modelos artificiales, haciéndolas difícil comparar entre sí, ya que las bases que pueden usarse para entrenar estos modelos pueden variar en tamaño y variedad de las imágenes.

Adicional a esto, dentro de las investigaciones desarrolladas que utilizan CNN usualmente se utilizan redes creadas por los investigadores basados en arquitecturas usualmente usadas y entrenadas en problemas de clasificación de imágenes, o por otro lado se utiliza metodologías de *Transfer Learning* para traer los pesos e hiper-parámetros en redes previamente entrenadas y que de esta manera generar la clasificación deseada. No obstante, no existen trabajos que hablen desde una perspectiva comparativa de estas metodologías y compare redes neuronales que sean generadas por ambos métodos y así llegar a conclusiones que determinen ruta al entrenar este tipo de modelos.

Por otro lado, el número de artículos utilizando estas tecnologías provenientes de Colombia es reducido, implementar investigaciones que aborden dichos problemas de clasificación de señas enmarcados en un entorno como el colombiano pueden ser clave para futuros desarrollos de tecnología de inclusión de la población que utiliza la LSC y la expansión de esta área de aplicación. Además teniendo en cuenta que el ámbito educativo para esta población resulta ser un reto, dependiendo del contexto social y cultural de la región, investigaciones que aporten a la base del desarrollo de tecnologías que faciliten procesos educativos son importantes para abrir paso al proceso de inclusión en la sociedad colombiana.

Dadas todas las condiciones antes descritas, se enmarca el problema específico que busca resolver dentro de los objetivos y resultados de esta investigación. En donde busca desarrollar modelos de clasificación basado en Inteligencia Artificial que provea una clasificación y reconocimiento de las señas del ASL para así comparar combinaciones a nivel de su arquitectura y de hiper-parámetros con la finalidad de definir el modelo que haga una buena clasificación dentro de la base de datos a utilizar.

Esta investigación es importante ya que si bien se enfoca en crear sistemas de clasificación de imágenes que sean la base para la predicción de letras y expresiones de la LSC, su objetivo no es la una apropiación del conocimiento o implementación directa con la población Sorda Colombiana. Este trabajo busca establecer las bases para investigaciones y clasificaciones futuras de la LSC. Una razón de esto es que al hacer una comparación individual de cada una de las señas del abecedario parte del ASL y LSC,

tienen el 76 % de similitud y estos solo difieren en las expresiones de 7 letras (F-P-Q-T-S-U-Z), haciendo que las conclusiones de este trabajo respecto a cada uno de los modelos evaluados, sea de utilidad en investigaciones futuras aplicadas al abecedario de la LSC.

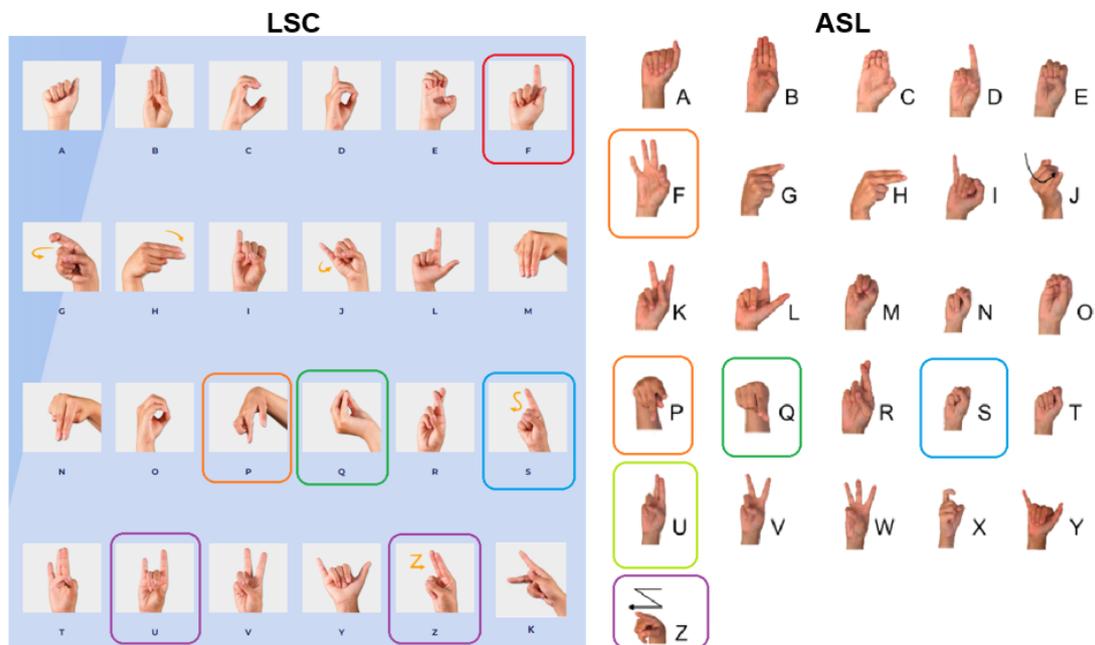


Figura 3.1: Comparación del abecedario de la Lengua de Señas Colombiana LSC y la Lengua de Señas Americana ASL

4. JUSTIFICACIÓN

William Stokoe, quien fue el primer investigador de la ASL, demostró en 1960 que la LS es una lengua natural y puede estudiarse en todos los niveles lingüísticos: fonológico, morfológico, semántico y pragmático. Además, demostró que esta puede ser estudiada desde las diferentes disciplinas lingüísticas, especialmente desde la psicolingüística y la sociolingüística. De esta manera, se puede asegurar que las lenguas de señas no son lenguas universales, pues responden a las diferencias idiolectales, diafásicas, diastráticas, diatópicas entre regiones y países. Siguiendo los estudios de Stokoe, con el tiempo diversos trabajos e investigaciones se han desarrollado en países europeos como Suecia, España, Alemania, Italia, Francia [18]. Del origen de la LS en Colombia hay pocos testimonios escritos y por tanto no se tiene precisión de sus comienzos, sin embargo hay registros de que en el año 1957 aparece la primera asociación de sordos en Bogotá y un año después otra en la ciudad de Cali.

Según la Federación Mundial de Sordos, existen aproximadamente 70 millones de personas sordas en todo el mundo [19]. En Colombia, el Plan Estratégico Institucional INSOR (Instituto Nacional para Sordos) del Ministerio de Educación proyectó un número de cerca de 560.000 personas sordas para el año 2020 basado en el último censo de 2005, [20] ubicadas en el territorio colombiano como se presenta en la Figura 4.1. La mayor concentración de personas sordas se encuentra hacia el interior del país en los departamentos de Antioquia y Bogotá. Sin embargo, se pueden apreciar poblaciones de gran número en la costa pacífica, en el Valle del Cauca y en la costa Norte en Bolívar y el Atlántico. En cuanto a la distribución por edades que tienen las personas sordas en Colombia, el informe reporta que la mayoría se encuentra entre los 70 y 79 años, como se aprecia en la Figura 4.2. En general, la distribución muestra que la mayoría de la población sorda se encuentra de los 50 años en adelante.

Las realidades de la inclusión de estas personas en nuestro país se ven altamente relacionadas con el poder de comunicación que tiene la sociedad que los rodea y las facilidades que esta ofrece para la plena realización de sus actividades de la vida diaria. En Colombia el 70% de las personas sordas se encuentran en un nivel de pobreza o vulnerabilidad y con frecuencia encuentran dificultades de acceso a los servicios del estado por las barreras comunicativas. Se estima que el 89% de las personas sordas se desempeñan en actividades que requieren una baja cualificación, mientras que el 11% lo hace en actividades que requieren algún tipo de cualificación, esto se refleja en que el 58% recibe un bajo ingreso por su desempeño en el trabajo y alrededor del 28% sea dependientes económicamente [20].

Aún con los esfuerzos que son llevados a cabo por parte de entidades gubernamentales y no gubernamentales por incluir a la población sorda del país en la realidad del mismo [18, 21], es evidente que se requiere de desarrollos desde diferentes ámbitos para acelerar el proceso. La educación en la LSC es sin duda un pilar para los mecanismos de inclusión de la población sorda del país. Sin embargo, la variación en ciertas señas entre regiones del país [22] hacen difícil la estandarización de los términos en esta lengua. Tra-

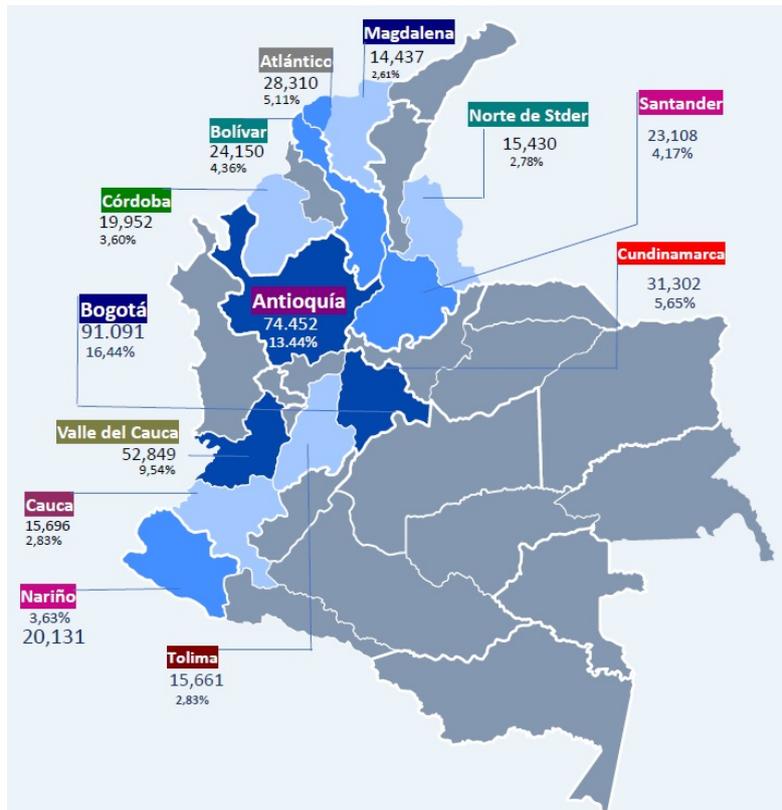


Figura 4.1: Distribución territorial de la población sorda colombiana, obtenido de [20].



Figura 4.2: Distribución por edades de la población sorda colombiana, obtenido de [20].

bajar en métodos de clasificación automática para imágenes de gestos de la LSC es una investigación clave en la búsqueda por soportar esta necesidad. A través de desarrollos tecnológicos como este se puede mejorar la calidad de vida de las personas sordas en el Colombia y el mundo, y dar pasos sólidos hacia la construcción de una sociedad más inclusiva y equitativa.

5. OBJETIVOS

5.1. General

Desarrollar un modelo de clasificación y reconocimiento de señas ASL basado en inteligencia artificial

5.2. Específicos

- Desarrollar modelos basados en aprendizaje de máquina y aprendizaje profundo para extracción de características de gestos y señas del ASL.
- Identificar una estrategia para clasificar automáticamente gestos y señas del alfabeto de la ASL usando imágenes.
- Evaluar sistemáticamente el modelo en conjuntos de datos de señas del alfabeto de la ASL.

6. MARCO TEÓRICO

El foco de investigación en tecnología para el aprendizaje de LS en el que se enmarca este trabajo son los sistemas de reconocimiento automático de gestos y movimientos. La idea detrás de estos sistemas es identificar patrones de movimiento y asignarle el significado que representa en términos de palabras o expresiones específicas.

Una de las maneras óptimas de estudiar los avances desarrollados en esta área de investigación es empezar por identificar el tipo o variación de la LS con la que se trabajó. Seguido, corresponde entender las diferentes tecnologías que se han implementado para llevar a cabo el proceso de reconocimiento de los diferentes movimientos de la determinada LS. Y, por último, se deben determinar y analizar los algoritmos de *Machine Learning* o métodos implementados para procesar y clasificar los movimientos.

De acuerdo con la revisión de literatura presentada por Wadhwan y Kumar [3], en la que se seleccionaron 117 artículos científicos que investigaban sistemas de reconocimiento para 25 diferentes lenguas de señas, las lenguas que han sido más estudiadas son la ASL y la ISL. Si bien LS de otros países del mundo, tales como China y Arabia también son ampliamente usadas, en este documento decidió estudiarse además la LS Mexicana (LSM) por su cercanía cultural a la LSC.

En relación a las tecnologías de captura de señales o imágenes para este tipo de investigación con LS, usualmente implementadas de acuerdo a la revisión, las cámaras de vídeo son los dispositivos más usados para el reconocimiento de señas. Esto se debe a que existe una mayor variedad de métodos que se pueden aplicar al análisis de imágenes o vídeos, en relación a otro tipo de señales extraídas directamente del cuerpo humano [6]. Además, se trata de sistemas no invasivos. Sin embargo, dispositivos como el Kinect (Microsoft, EEUU) y guantes sensorizados [4] también representan una porción significativa de los estudios. Otros sistemas como los de electro-miografía (EMG), electroencefalografía (EEG) y el de Leap Motion (Motion Control, EEUU) son menos usados debido a la complejidad de adaptación y la extracción de la señales biológicas que añaden al procesamiento de los movimientos.

En cuanto a las técnicas para el procesamiento utilizadas, los algoritmos de redes neuronales (ANN y CNN) son los utilizados más frecuentemente con las diferentes tecnología para el reconocimiento de patrones de movimiento. Sin embargo, otros algoritmos destacados son las máquinas de soporte vectorial (SVM, del inglés Support Vector Machines), el algoritmo de los K vecinos más cercanos (KNN, del inglés K Nearest Neighbors), el análisis de discriminante lineal (LDS, del inglés Linear Discriminant Analysis), los modelos ocultos de Márkov (HMM, del inglés Hidden Markov Model) y la deformación dinámica del tiempo (DTW, del inglés Dynamic Time Warping).

Las redes neuronales son algoritmos inspirados en cómo funciona el cerebro humano, de manera que programas informáticos puedan reconocer patrones y resolver problemas. Estás hacen parte del aprendizaje profundo (Deep Learning, en inglés), un subconjunto de Machine Learning (que a su vez es parte de la Inteligencia Artificial) donde los algoritmos aprenden de grandes cantidades de datos de manera progresiva al realizar

una tarea repetitiva a través de “capas”. El Deep learning ha tenido un gran impacto varias industrias ya que se puede utilizar para el análisis avanzado de imágenes, la investigación, el descubrimiento de medicinas y, en general, tareas de predicción [23].

Las ANN, como algoritmo, están formadas por capas de neuronas artificiales, que contienen: (i) una capa de entrada, (ii) una o varias capas ocultas y (iii) una capa de salida. De la manera cómo funciona la red, cada neurona se conecta a otra y tiene un peso y un umbral asociados, de manera que si la salida de una neurona individual está por encima del valor de umbral especificado, dicha neurona se activa y envía datos a la siguiente capa de la red. De lo contrario, si se activa, no se pasan datos a la siguiente capa de la red [24]. En la Figura 6.1 se muestra la arquitectura clásica de una ANN con una capa de entrada de 5 neuronas como entrada (en azul), varias capas ocultas de procesamiento (en verde) y una capa de salida de 3 neuronas (en lila).

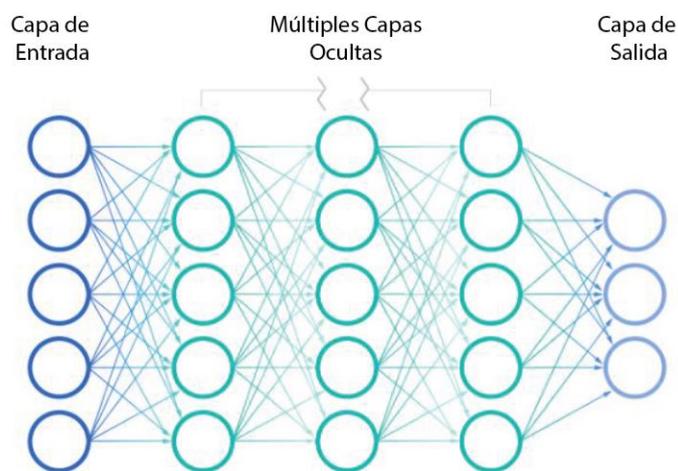


Figura 6.1: Estructura de una ANN, modificado de [24].

Cada neurona funciona como un modelo de regresión lineal, conformado por datos de entrada x , ponderaciones w , un sesgo (*bias*- b , en inglés) o umbral y una salida como se muestra en las ecuaciones 6.1 y 6.2:

$$\sum_{i=1}^m w_i x_i + bias = w_1 x_1 + w_2 x_2 + w_3 x_3 + bias \quad (6.1)$$

$$Salida = f(x) = \begin{cases} 1 & \sum w_1 x_1 + b \geq 0 \\ 0 & \sum w_1 x_1 + b < 0 \end{cases} \quad (6.2)$$

Primero se determina una capa de entrada y a esta se le asignan ponderaciones que permiten determinar la importancia de cualquier variable. Como es de esperarse entre

más grandes las ponderaciones, contribuyen más significativamente a la salida respecto a otras entradas. Una vez las entradas se multiplican por sus respectivas ponderaciones y se suman, la salida se expresa a través de una función de activación. Si esta supera un umbral determinado, activa la neurona y los datos pasan a la siguiente capa de la red. De esta forma es como la salida de una neurona se convierte en la entrada de la siguiente.

Las redes neuronales se clasifican en diferentes tipos y dependen de los fines que se buscan con ellas. El perceptrón es la red neuronal más antigua, creada por Frank Rosenblatt en 1958 y consiste en una sola neurona, siendo la forma simple de una red neuronal. Por otro lado están las redes neuronales de propagación hacia delante o perceptrones multicapa (MLP, del inglés *Multilayer Perceptron*), formadas por una capa de entrada, una capa o varias capas ocultas y una capa de salida, como la que se presenta en la Figura 6.1. Esta es la base para la visión artificial, el procesamiento del lenguaje natural y otras redes neuronales. También existen las redes neuronales recurrentes (RNN, del inglés *Recurrent Neural Networks*) que constan de bucles de retroalimentación y se utilizan principalmente con datos de series temporales para hacer predicciones sobre resultados futuros.

Si bien existen otros tipos de ANN, las últimas redes mencionadas en este documento son las CNN, son los algoritmos en los que se centrará este trabajo de investigación. Son redes neuronales similares a las redes de propagación hacia delante, pero que aprovechan los principios de la multiplicación de matriz (matrices bidimensionales) para identificar patrones dentro de una imagen y son ampliamente utilizadas para el reconocimiento de imágenes, el reconocimiento de patrones y/o la visión artificial. La Figura 6.2 muestra la arquitectura clásica de una CNN que consta de una imagen de entrada, varias capas ocultas de convolución matricial y una capa de salida neuronal similar la previamente presentada en otras redes.

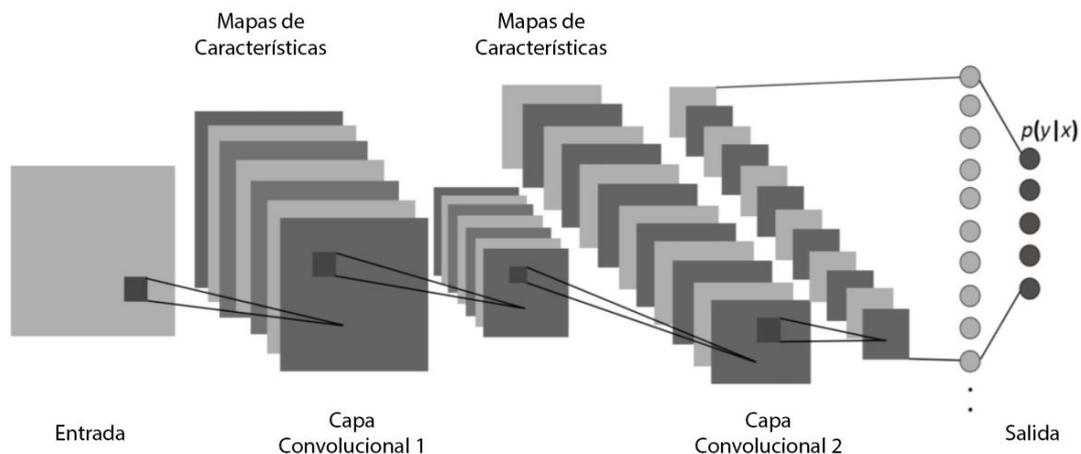


Figura 6.2: Estructura de una CNN, modificado de [25].

En las Tablas 6.1, 6.2, 6.3 se muestra un resumen de los sistemas de reconocimiento usados para países como Estados Unidos (ASL), India (ISL) y Mexico (LSM) según revisión sistemática de literatura por Wadhwan y Kumar [3], siendo la Lengua de señas en estos países representativas dentro de la investigación de estas tecnologías de reconocimiento de imágenes. En las Tablas se muestra la tecnología usada y los algoritmos de *Machine Learning* o métodos utilizados para el clasificación de los movimientos a través de cada una de las tecnologías. Además el número de artículos que implementan cada tecnología de manera descendente para observar tendencias tanto en la tecnología misma, como el algoritmo de *Machine Learning* o método utilizado.

N. de Artículos	LS	Tecnología de Adquisición	Algoritmos de Machine Learning o Métodos Empleados
10	ASL	Cámara de Vídeo	SVM, KNN, ANN, CNN
6	ASL	Kinect	SVM, KNN, CNN
3	ASL	Sistema de EMG	SVM
2	ASL	Guantes	ANN
1	ASL	Sistema de EEG	LDA y SVM
1	ASL	Sistema Leap Motion	SVM y KNN
1	ASL	Sensor de radio impulso	CNN

Tabla 6.1: Resumen de los sistemas de reconocimiento de la Lengua de señas Americana (ASL) según revisión sistemática de literatura por Wadhwan y Kumar [3].

N. de Artículos	LS	Tecnología de Adquisición	Algoritmos de Machine Learning o Métodos Empleados
13	ISL	Cámara de Vídeo	SVM, KNN, ANN, DTW
4	ISL	Kinect	SVM, HMM, CNN
4	ISL	Sistema Leap Motion	ANN, HMM

Tabla 6.2: Resumen de los sistemas de reconocimiento de la Lengua de señas India (ISL) según revisión sistemática de literatura por Wadhwan y Kumar [3].

N. de Artículos	LS	Tecnología de Adquisición	Algoritmos de Machine Learning o Métodos Empleados
2	LSM	Kinect	ANN, DTW
1	LSM	Cámara de Vídeo	ANN

Tabla 6.3: Resumen de los sistemas de reconocimiento de la Lengua de Señas Mexicana LSM según revisión sistemática de literatura por Wadhwan y Kumar [3].

7. METODOLOGÍA

7.1. Tipo de investigación a realizar

El tipo de investigación de este trabajo de grado es del tipo descriptivo, ya que busca describir y encontrar las mejores características para diferentes modelos de CNN y seleccionar la mejor combinación de estas características que permita desarrollar un modelo para la clasificación de imágenes de señas del abecedario del ASL. Además al llevar a cabo esta investigación sobre bases de datos públicas para encontrar los parámetros útiles para futuros clasificadores, esta investigación se puede considerar del tipo transversal, ya que dicha exploración se hará basada sobre datos ya recolectados buscando describir la metodología de investigación representada en la arquitectura de la red neuronal que tenga mejores resultados en el proceso de clasificación [26].

7.2. Fases de la investigación

En la figura 7.1 muestra las actividades que se deben desarrollar para cumplir con los objetivos planteados en este proyecto de investigación. Como primera etapa se encuentra la colección e identificación de bases de datos de imágenes correspondientes a las 26 letras del abecedario de la ASL, para esto se utilizarán dos bases de datos publicadas para el estudio de reconocimiento de imágenes parte del abecedario en la ASL [27, 28].

Posteriormente, se cargarán las bases de datos y se hará un preprocesamiento de las imágenes para aplicar técnicas normalización y reajuste que permitan un análisis sencillo por el sistema de clasificación basado en modelos de CNN. Después de esto se hará una definición y construcción de cada una de las diferentes estructuras de CNN que se utilizarán para hacer una exploración de su funcionamiento ante diferentes condiciones dadas por sus hiper-parámetros, para así encontrar la mejor combinación de estos y compararlos para identificar la mejor estrategia para la clasificación automática de imágenes de señas del alfabeto de la ASL.

7.3. Técnica usada para la recolección y procesamiento de la información

Este trabajo utiliza bases de datos de libre acceso, las cuales han sido dispuestas para la evaluación de diferentes sistemas de clasificación de imágenes mediante el uso de redes neuronales, cada uno de los data sets esta compuesto por 29 clases que clasifican las señas realizadas en cada imagen capturada. Estas clases hacen referencia a las señas de 26 letras del abecedario del idioma inglés (A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z). Además, las dos clases referentes a las señas de SPACE o espacio (representa un espacio entre dos conjuntos de letras que conforman

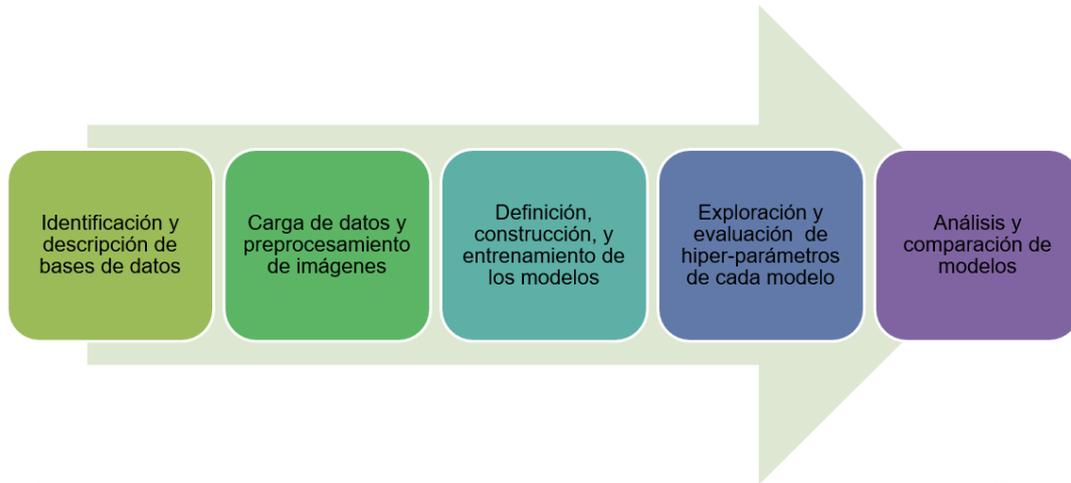


Figura 7.1: Diagrama general de actividades

dos palabras diferentes) de DELETE o borrar (representa el borrado o eliminación del último carácter usado durante el uso de ASL) y una última clase llamada NOTHING o nada que es un conjunto de imágenes sin la presencia de ningún gesto y diferentes formas de fondo [27, 28].

En cada uno de los conjuntos de datos, las imágenes fueron tomadas por diferentes dispositivos móviles como computadores portables y celulares con diferentes resoluciones, sin embargo fueron preprocesadas por sus autores para solo enfocarse en la sección de la mano en el gesto de cada una de las señas capturadas y almacenarlas en formato jpg y con un tamaño total de 200X200 pixeles cada una. La tabla 7.1 muestra para cada conjunto de datos el autor, la cantidad de imágenes y su distribución por clases y así mismo una etiqueta con la que se le referenciará durante toda la investigación; además, en las figuras 7.2 y 7.3 se observa una muestra aleatoria de las imágenes que componen cada una de las bases de datos usadas, y en la parte superior la etiqueta que representa la letra o palabra de la seña mostrada en cada imagen.

Autor	Tamaño (cantidad de imágenes)	Etiqueta
D. Rasband	870 imágenes en total. 30 por cada clase	Base A
A. Akash	87000 imágenes en total 3000 por cada clase	Base B

Tabla 7.1: Distribución de imágenes y etiquetas para las bases de datos utilizadas.

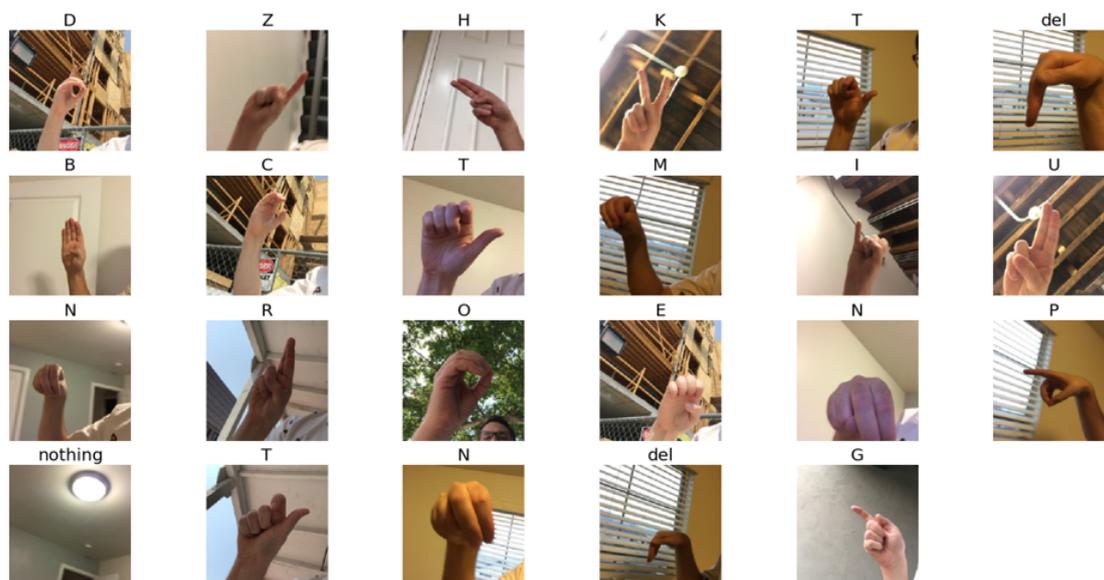


Figura 7.2: Muestra del conjunto de imágenes Base A

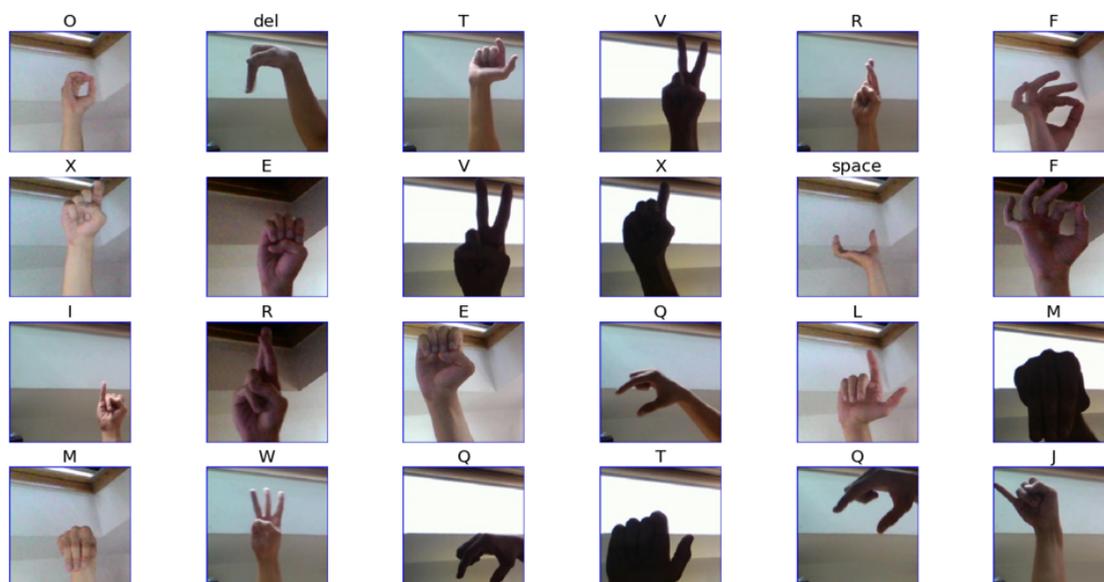


Figura 7.3: Muestra del conjunto de imágenes Base B

7.4. Carga y preprocesamiento de imágenes

El lenguaje de programación utilizado para el desarrollo de los algoritmos de procesamiento de imágenes para la construcción de las redes neuronales profundas convolucionales fue Python®[®], esto debido a su interpretación intuitiva y múltiples librerías de funciones [29], en especial aquellas que permiten el procesamiento de imágenes además de la creación, diseño y evaluación de redes neuronales para modelos de clasificación de imágenes. Un ejemplo de las librerías usadas para lo anteriormente descrito son *TensorFlow* [30], *ScikitLearn*[31] y *Keras* [32], en especial esta última que permite descargar, entrenar y usar modelos previamente creados para otros ejercicios de clasificación de imágenes como los que se emplearan en este trabajo.

Además como plataforma para ejecutar y realizar el código se utilizara Google Colab que es un producto de Google Research que permite escribir y ejecutar código de Python directamente en el navegador de internet, también ofrece funcionalidades que permiten analizar altos volúmenes de datos como es el caso de este trabajo, en especial para procesos de entrenamiento de redes neuronales donde ofrece el uso de unidades de procesamiento gráficas (GPU) para el procesamiento de altos volúmenes de información, para el caso de este trabajo se utilizó *Persistence – M Tesla T4 y P100* [33].

En la etapa inicial el algoritmo buscará cargar las imágenes al entorno de programación mediante funciones contenidas en la librería de *Keras* que permiten identificar cada uno de los directorios donde están almacenadas las imágenes, son ordenadas en diferentes carpetas de acuerdo con el significado de su seña en el ASL. Para después extraer la información de las dichas imágenes y almacenar en forma matricial donde cada píxel es representado por tres canales de información dados en la escala RGB y que varían en un rango de 0 a 255, es por esto que se hace una normalización y estalación de cada uno de estos canales para que tomen un valor entre 0 y 1 y las imágenes en su representación matricial sean fácilmente procesadas durante la fase de entrenamiento de los modelos utilizados.

Una vez obtenida la información de todas las imágenes cargadas en los diferentes arreglos, se realiza un proceso de creación y codificación de las etiquetas de cada una de estas señas para vincular cada imagen con las 29 posibles clasificaciones que representa. Para el proceso de codificación, primero se lleva a cabo un proceso de codificación utilizando etiquetas para codificar cada una de las señas como un número que va del 0 al 28, siendo así A codificada como 0, B codificada como 1, C codificada como 3 y así sucesivamente. Después con el objetivo de cumplir con etiquetas que sean compatibles con los algoritmos de aprendizaje automático y profundo, se lleva a cabo el segundo proceso de codificación en donde se transformarán los valores numéricos ya convertidos en una matriz bidimensional binaria, de tal manera que en dicha matriz contenga una columna para cada posible etiqueta creada previamente, en donde es marcada con un valor de 1 de acuerdo a la columna que pertenezca la seña que se representa en cada imagen procesada.

Finalmente, se harán dos particiones en cada una de las bases de datos, la primera se hará en una proporción del (90-10) % del total de las imágenes, donde el 10 % se utilizará

como testeo final después de que los modelos sean entrenados y configurados con cada una de las combinaciones de hiper-parámetros para cada base de datos. El otro 90 % se usara para el proceso de entrenamiento y validación individual de cada modelo como se hace generalmente para los problemas de clasificación de imágenes[34].

Generalmente, para ejercicios de clasificación se ha encontrado que una partición del (70-30) % de los datos es mas que óptima al entrenar modelos usando bases de datos de imágenes[35, 36]. Teniendo en cuenta lo anterior y la primera división para el testeo general de los modelos en cada base de datos, en este caso se utilizo una proporción (80-20) % donde el primer conjunto de estos con la mayor proporción de imágenes será llamado “*train*” o conjunto de entrenamiento el cual se usará para entrenar a nuestra red para reconocer e identificar patrones en los datos que representan las imágenes de cada set de datos, que a su vez se validará durante este proceso de entrenamiento con el conjunto del 20 % que actuará como validación del modelo.

7.5. Definición y Entrenamiento de los modelos

Para la definición de los modelos que serán comparados fueron definidas tres arquitecturas predefinidas disponibles en Keras, una biblioteca de código abierto con varias aplicaciones de redes neuronales escritos en Python. Para esta investigación las tres arquitecturas base han sido utilizadas en múltiples ejercicios y problemas de clasificación de imágenes y en especial en la clasificación de imágenes de señas dentro de diferentes lenguas de señas en el mundo con resultados prometedores, además de compartir características útiles como el tamaño de las imágenes que usa cada red. Todas estas redes neuronales han sido desarrolladas, entrenadas y utilizadas en un principio en torno a resolver el desafío ImageNet para el reconocimiento de objetos y detección de imágenes a larga escala de la Universidad de Stanford [37].

- **ResNet50V2:** Es una versión modificada de la red inicial ganadora del primer lugar en el concurso de clasificación ILSVRX 2015, esta cuenta con un total de 152 capas distribuidas con 50 bloques convolucionales, tiene aproximadamente 25.5 millones de parámetros dentro de su configuración. Esta utiliza una serie de de bloques con redes residuales profundas (Deep residual Network) que mediante conexiones entre cada uno de los bloques convolucionales ayuda a mejorar la precisión del modelo evitando incrementar el número de capas y parámetros [38].
- **MobileNetV2:** Es uno de los modelos de la familia MobilNets los cuales son usados para aplicaciones de visión móviles e integradas. Estas arquitecturas se basan en optimizar su funcionamiento manteniendo su número de parámetros bajo uniendo capas convolucionales de expansión, profundidad y proyección en bloques con conexiones residuales invertidas sencillas. Tiene un total de 53 capas organizadas en un total de 16 bloques y compuesta por untotal de 4.5 millones de parámetros dentro de la configuración de su red. [39].

- **InceptionResNetV2:** Este modelo tiene un total de 164 capas y un total de 55.4 millones de parámetros, combina el modelo robusto de su red predecesora InceptionV3 la cual pese a su robustez tiene un costo computacional, junto con la estrategia de redes residuales profundas de las redes de la familia de modelos ResNet mejorando considerablemente sus resultados en múltiples ejercicios de clasificación, ya que al ser el diseño del modelo InceptionV3 de una arquitectura compuesta de redes muy profunda las redes residuales ayudan a mejorar sus resultados [40].

Ante cada una de estas configuraciones base se hacen modificaciones a su arquitectura donde se agregaron capas de neuronas adicionales con características y funciones específicas que permitan adaptar y mejorar la clasificación que realiza cada modelo a la cantidad de clases e imágenes del problema. Estas capas son consideradas como un “*finetunning*” o un ajuste fino al modelo para que pueda clasificar la cantidad de clases esperadas y estará compuesto por dos capas de una capa ‘Densa’ y otra de ‘Caída’ conocidas en el idioma inglés como *Dense* y *DropOut* respectivamente, configuradas de tal manera que la salida de la clasificación se ajuste a las 29 clases que representan el abecedario de ASL.

Después de realizar el proceso de “*finetunning*” en cada uno de los modelos a usar, se procederá a entrenar cada uno de los tres modelos con cada uno de los conjuntos de imágenes usados durante la investigación (Base A y Base B). Mediante diferentes funciones se cargara la información de cada imagen dentro de cada conjunto de imágenes y cada uno de los modelos, de esta manera cada modelo aprendió las características de cada conjunto de imágenes. Luego se ajustaron sus pesos para poder acomodar su modelo de clasificación a las 29 clases en las que se dividen las imágenes dentro de cada conjunto de datos de acuerdo a sus características.

Además, se realizó una exploración de dos de los hiper-parámetros dentro del entrenamiento y ajuste del modelo a las imágenes de cada conjunto de datos, estos son el tamaño swl lote o “*Batch size*” (BS) por su nombre en inglés, que representa el tamaño de la muestra o cantidad de imágenes que el modelo usa en cada ciclo de entrenamiento para poder aprender las características de cada clase, y por otro lado la tasa de aprendizaje o “*Learning rate*” (LR) que se puede interpretar como que tan rápido el modelo se adapta al problema que se quiere abordar, ya que representa en que cantidad se actualizan los pesos dentro de cada red neuronal en cada iteración dentro del proceso de aprendizaje del modelo. Para este trabajo se variaron de manera experimental los valores del BS y LR guiados en los valores frecuentemente usados en estos problemas de clasificación y el tamaño de las bases de datos estudiadas, los valores en los que se variarán se describen en la Tabla 7.2.

Esta exploración se hará evaluando los valores mínimos de pérdida del modelo entre las diferentes combinaciones de los hiper-parámetros anteriormente descritos, ya que este representa el error generado en la clasificación en general de todas las clases durante la clasificación al emplear el modelo entrenado en la porción del conjunto de datos destinados para la validación. Esto se hará de manera iterativa usando todas las posibles combinaciones contempladas en la variación de estos parámetros hasta encontrar para

cada combinación de Base de datos y modelo los valores de BS y LR.

Dado que inicialmente cada uno de los modelos se entrenan con los pesos predeterminados en cada modelo de acuerdo con el uso en su base de datos inicial (Reto *ImageNet* [37]), se buscará hacer una segunda exploración a través de un segundo proceso de “finetuning” el cual consistirá en permitir que algunas de las capas finales de cada red puedan entrenarse con a información determinada de acuerdo a las imágenes de cada conjunto de datos de ASL. Este proceso es comúnmente conocido como hacer transferencia de aprendizaje o “*Transfer Learning*” en ingles aplicado en las ultimas capas del modelo, y el objetivo de esta exploración es comparar el comportamiento de los modelos cuando no tienen modificación contra la manera que trabajan teniendo el último o los últimos dos bloques de redes convolucionales entrenados de esta manera, en donde en estos dos últimos escenarios se espera los modelos se puedan ajustar de mejor manera a las características de cada uno de los conjuntos de imágenes de ASL.

Finalmente, cada iteración dentro de las exploraciones propuestas se correrá durante 20 ciclos de entrenamiento ya que experimentalmente se ve que el comportamiento de los modelos no varía después de esta cantidad de ciclos de entrenamiento. Teniendo en cuenta todo lo antes mencionado la Figura 7.4 muestra el esquema general de las exploraciones y evaluaciones que se realizarán durante el proceso de evaluación descrito en la metodología.

Hiper-parámetro	Variación
BS	16 - 32 - 64 - 128
LR	0,01 - 0,001 - 0,0001 - 0,00001 - 0,000001

Tabla 7.2: Variación de Hiperparámetros utilizada para la exploración en cada modelo.

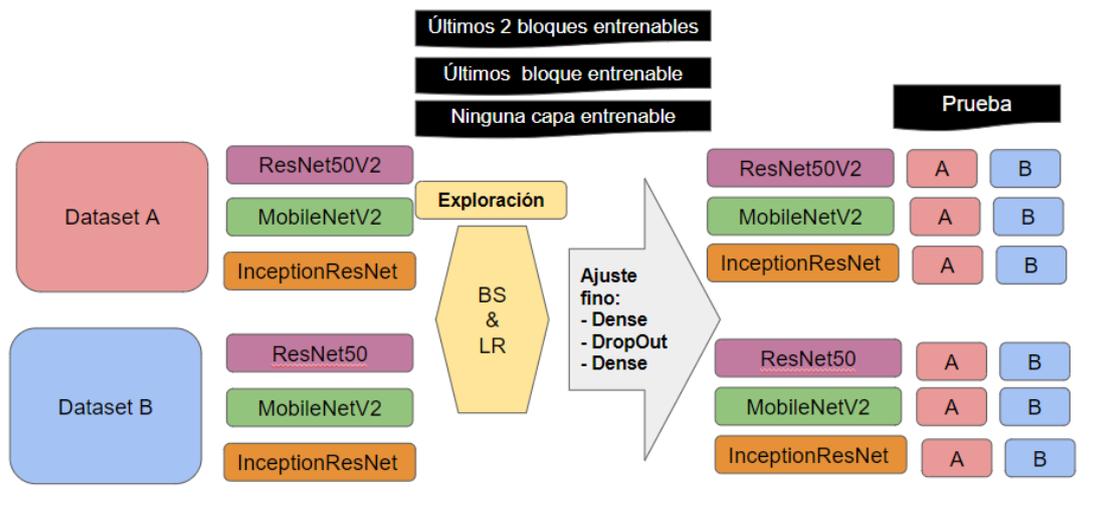


Figura 7.4: Diagrama del proceso de exploración y evaluación de las combinaciones de modelos de Redes neuronales

7.6. Análisis y conclusiones de los resultados

Por último, la evaluación de los modelos de clasificación a como de sus ajustes, se hará mediante el análisis de las variables que se muestran en la Tabla 7.3, estas son métricas comúnmente utilizadas para evaluar modelos de redes neuronales para clasificación y son calculados a través del conteo de casos clasificados como Falsos Positivos (FP), Falsos Negativos (FN), Verdaderos Positivos (TP) y Verdaderos Negativos (TN). Después de obtener los valores de eficiencia para la clasificación en cada modelo, se hará un análisis comparativo para determinar cuál de los modelos puede dar mejor respuesta al problema de clasificación de las señas analizadas.

Además de esto se analizará el comportamiento durante el entrenamiento de cada uno de los modelos después de su exploración para observar el comportamiento de la precisión y pérdida durante cada uno de los ciclos de entrenamiento. Finalmente se capturará el tiempo de corrida de cada uno de los sistemas de clasificación y sus exploraciones para identificar si alguna de las modificaciones realizadas pueda no ser adecuada para mantener el tiempo normal de corrida esperado de cada clasificación e identificar el posible tiempo que se pueda emplear en investigaciones futuras usando este tipo de modelos de CNN.

Variable	Definición	Formula
Exactitud (Accuracy)	Total de clasificaciones correctas frente al total de muestras clasificadas	$Acc = \frac{TP + TN}{TP + FP + FN + TN}$
Precisión	Representa el número de clasificaciones correctas en cada clase clasificada correctamente	$Precision = \frac{TP}{TP + FP}$
Sensibilidad (Recall)	Numero de clasificaciones verdaderas que fueron correctamente clasificadas	$Recall = \frac{TP}{TP + FN}$
F1-Score	Cálculo de la relación entre la medición de Precisión y Sensibilidad.	$F1score = 2 * \frac{Precision * Recall}{Precision + Recall}$

Tabla 7.3: Variables para la evaluación de los modelos basados Deep Learning y otros modelos de redes neuronales

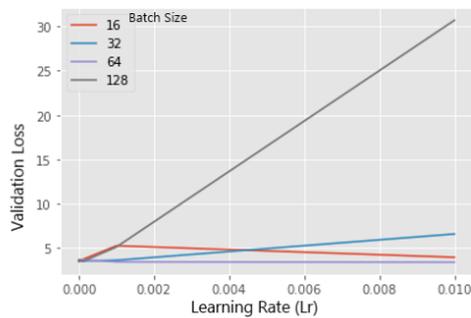
8. RESULTADOS

A continuación se muestran los resultados obtenidos a partir del entrenamiento de cada uno de los modelos para los diferentes conjuntos de imágenes. Para cada uno de los conjuntos de datos se mostrará los resultados de la exploración de hiper-parámetros, en cada modelo bajo los diferentes tipos de estructuras, es decir sin ninguna capa entrenable (distinguida como NTL "*No Trainable Layer*"), con el último bloque de capas entrenable (TL1) y con los últimos dos bloques de capas entrenables (TL2). Adicional a esto se reportará el comportamiento durante el entrenamiento, para mostrar la variación en precisión y pérdida del modelo. Después de que los modelos se entrenaron se almacenaron los pesos y configuración de cada red neuronal bajo los mejores hiper-parámetros encontrados durante la exploración; se evaluará el desempeño de cada modelo al clasificar las imágenes del conjunto de prueba extraído de cada conjunto de datos A y B. Para esto se usará las métricas descritas en la sección 7.6 de la clasificación del desempeño global de cada modelo, además de una observación del comportamiento de la clasificación en cada una de las clases que representan de manera individual cada letra del abecedario de ASL.

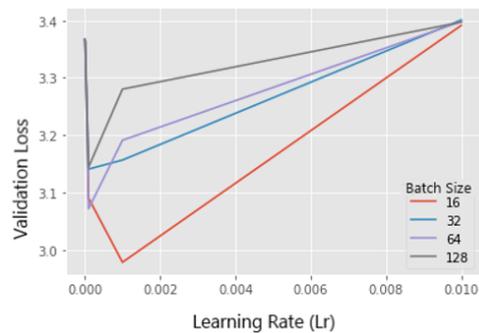
8.1. Resultados entrenamiento Conjunto de Datos A

8.1.1. Resultados exploración de parámetros

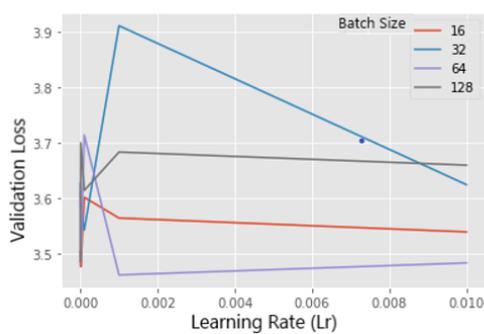
Al realizar la exploración de los hiper-parámetros y evaluar la mínima pérdida en validación para cada una de las combinaciones mencionados en la metodología, entrenados con conjunto datos Base A , se obtuvieron los resultados mostrados en las Figuras 8.1 y 8.2 así mismo como en la tabla 8.1 donde se muestra el resumen de la exploración.



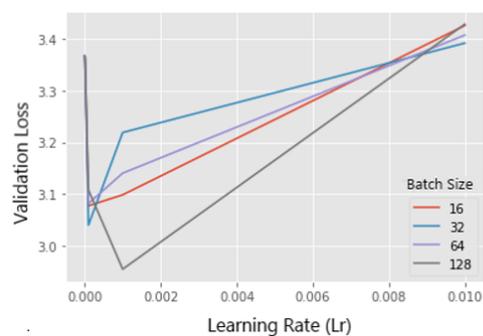
(a) Exploración NTL utilizando ResNet50V2 Base A



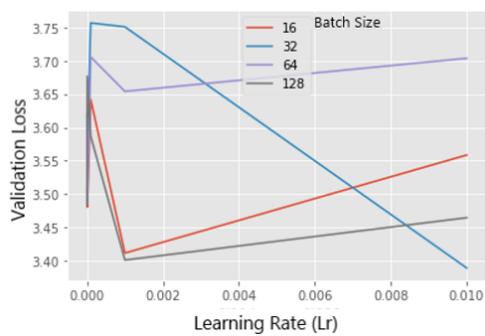
(b) Exploración NTL utilizando MobileNetV2 Base A



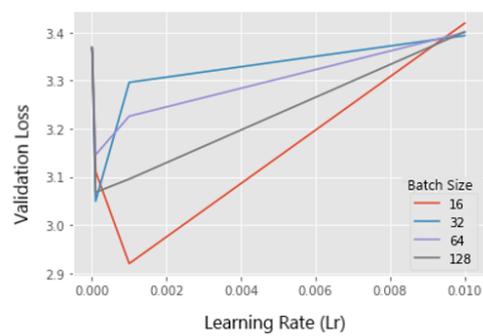
(c) Exploración TL1 utilizando ResNet50V2 Base A



(d) Exploración TL1 utilizando MobileNetV2 Base A

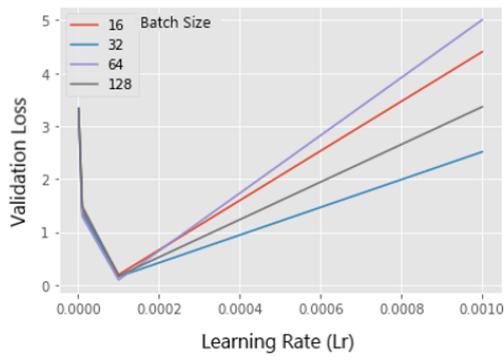


(e) Exploración TL2 utilizando ResNet50V2 Base A

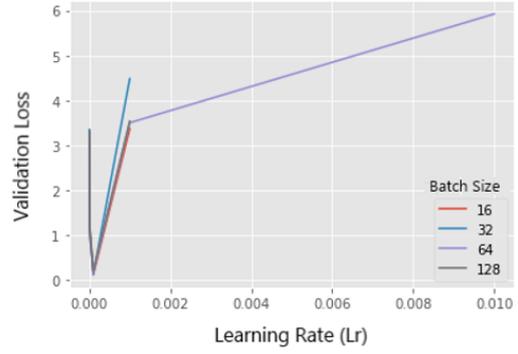


(f) Exploración TL2 utilizando MobileNetV2 Base A

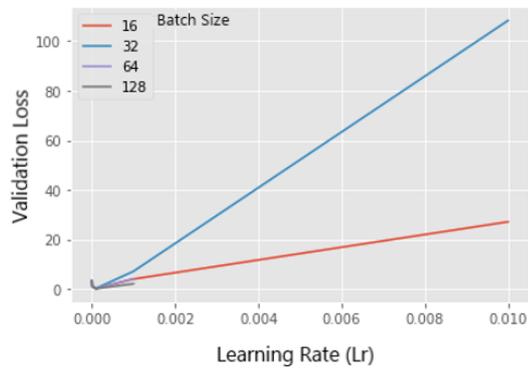
Figura 8.1: Resultados de la exploración de hiper-parámetros para la modelo utilizando ResNet50V2 Base A (izquierda) y MobileNetV2 Base A (derecha)



(a) Exploración NTL



(b) Exploración TL1



(c) Exploración TL2

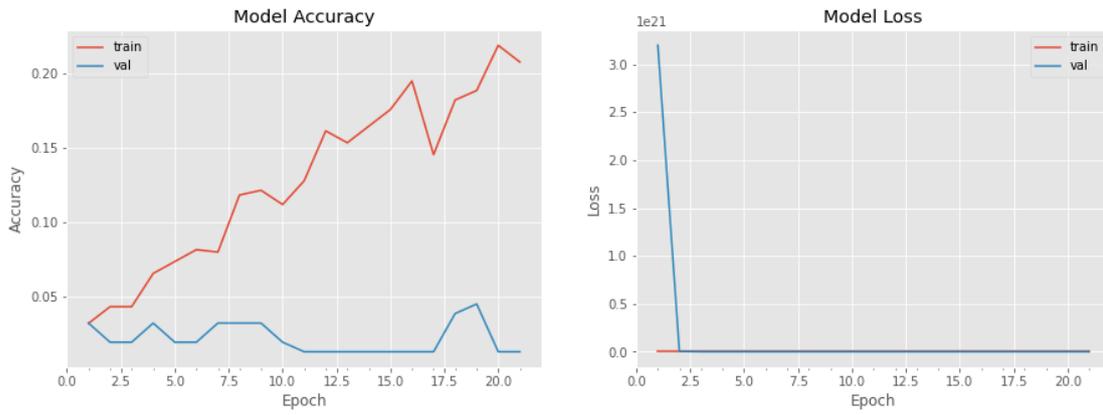
Figura 8.2: Resultados de la exploración de hiper-parámetros para la modelo utilizando InceptionResNetV2 Base A

Capas entrenables	Modelo	BS	LR	Mínima pérdida validación	Tiempo de exploración (horas)
Ninguna capa entrenable	ResNet50V2	64	0.01	3.41	4.5
	MobileNetV2	16	0.001	2.97	3.3
	InceptionResNetV2	64	0.0001	0.09	7
Último bloque entrenable	ResNet50V2	64	0.001	3.46	7.5
	MobileNetV2	128	0.001	2.95	7.3
	InceptionResNetV2	16	0.0001	0.13	16.3
Últimos dos bloques entrenables	ResNet50V2	32	0.01	3.38	8.3
	MobileNetV2	16	0.001	2.93	8
	InceptionResNetV2	64	0.0001	0.15	20

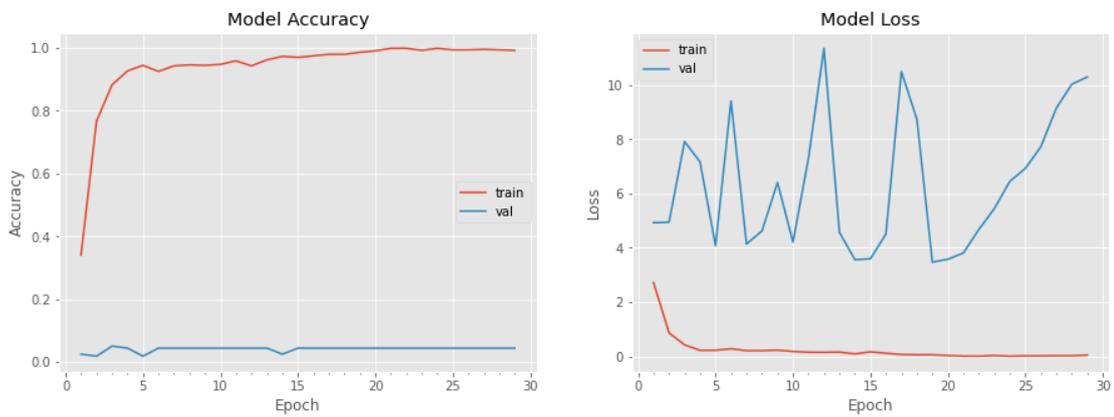
Tabla 8.1: Mejores hiper hiper-parámetros encontrados durante Exploración Conjunto de imágenes Base A.

8.1.2. Resultados de entrenamiento de los modelos

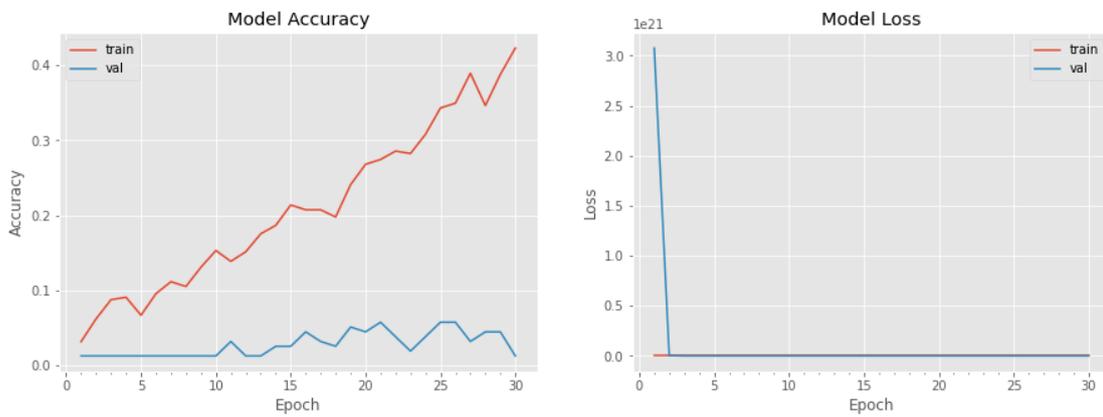
Después de identificar la mejor combinación de hiper-parámetros para cada una de las estructuras y modelos, se procedió a observar el comportamiento de la precisión *Accuracy* y la pérdida *Loss* durante el entrenamiento de cada modelo bajo sus mejores hiper-parámetros lo cual se ve en las Figuras 8.3, 8.4 y 8.5.



(a) Entrenamiento NTL

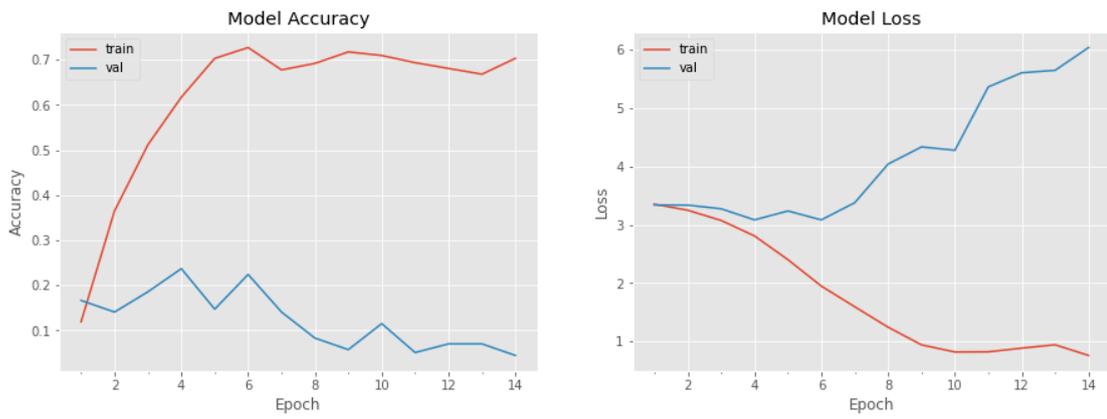


(b) Entrenamiento TL1

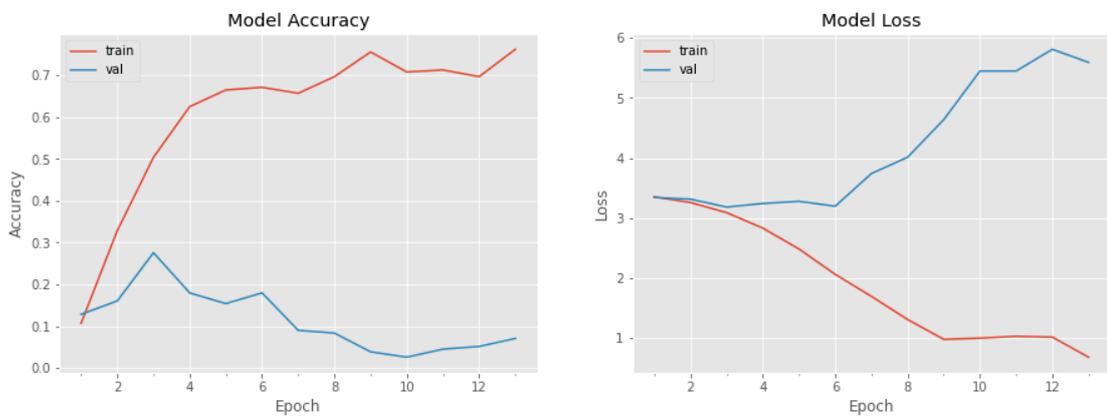


(c) Entrenamiento TL2

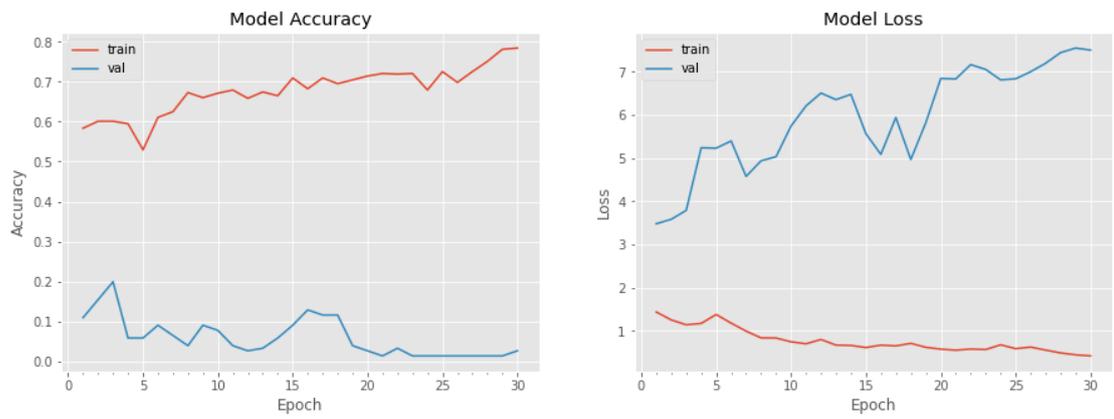
Figura 8.3: Comportamiento durante el entrenamiento redes basadas en ResNet50V2 Base A



(a) Entrenamiento NTL

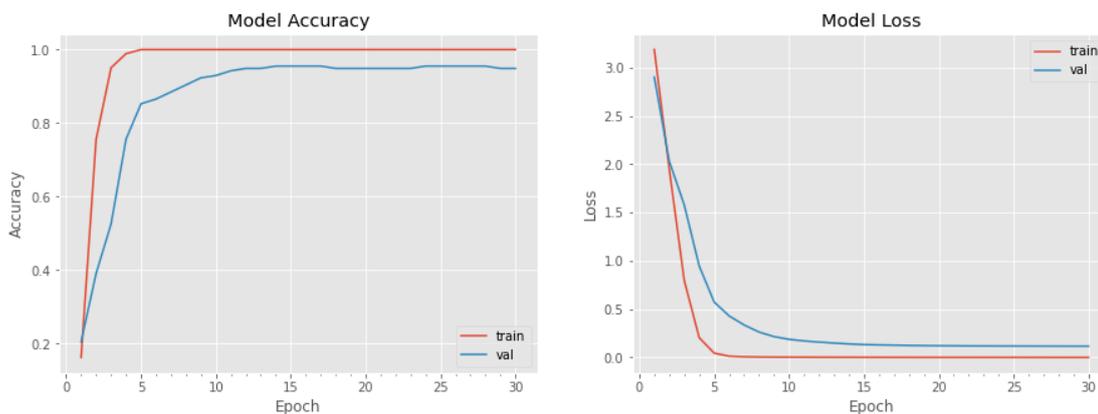


(b) Entrenamiento TL1

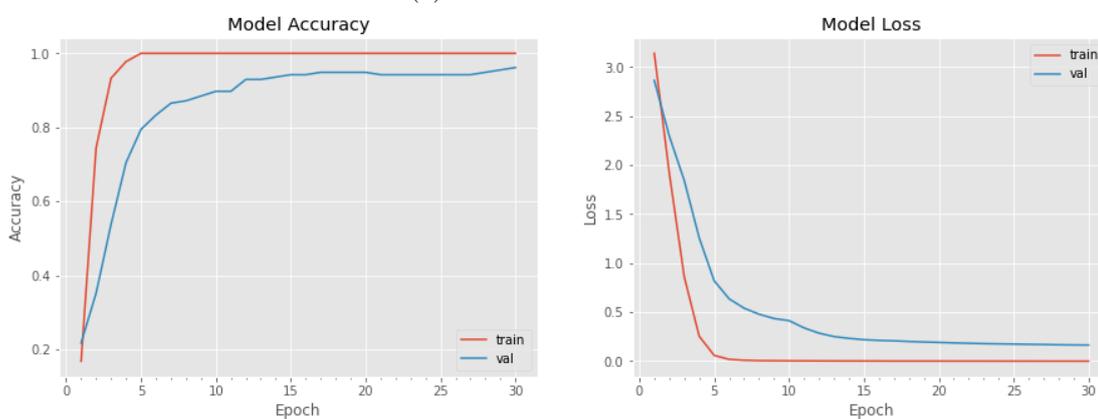


(c) Entrenamiento TL2

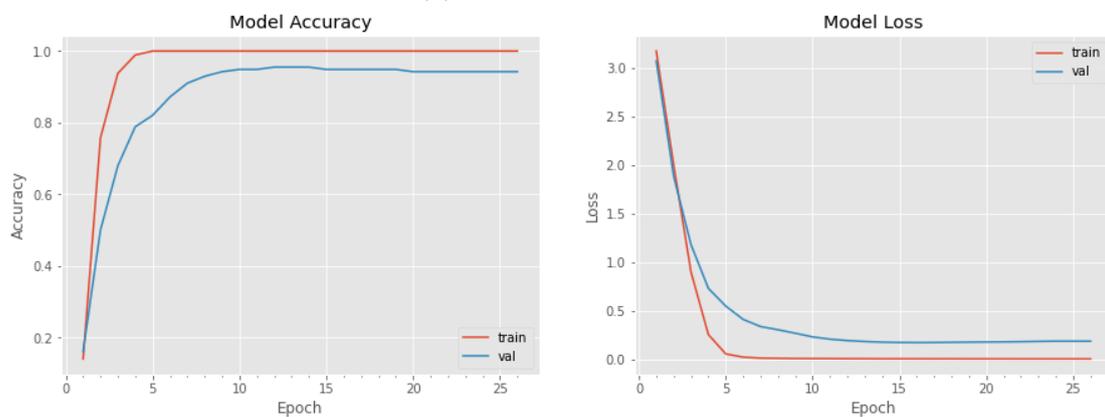
Figura 8.4: Comportamiento durante el entrenamiento redes basadas en MobileNetV2



(a) Entrenamiento NTL



(b) Entrenamiento TL1



(c) Entrenamiento TL2

Figura 8.5: Comportamiento durante el entrenamiento redes basadas en InceptionRes-NetV2

8.1.3. Evaluación de la predicción con modelos entrenados

Por último, los modelos ya entrenados fueron evaluados realizando la predicción del grupo de imágenes de testeo de los conjuntos de imágenes Base A y Base B, como lo muestran las tablas 8.2 y 8.3 para las métricas anteriormente mencionadas en la metodología.

Capas entrenables	Modelo base	Base A			
		Acc	recall	f1 score	Observación inter clases
Ninguna capa entrenable	ResNet50	0.03	0.03	0	Clasifica todas las muestras como la clase L
	MobileNetV2	0.17	0.28	0.17	No llega a una clasificación clara
	InceptionResNetV2	0.92	0.92	0.92	Baja precisión y sensibilidad en las letras M y N
Último bloque entrenable	ResNet50	0.03	0.03	0	Clasifica todas las muestras como la clase Nothing
	MobileNetV2	0.17	0.28	0.17	No llega a una clasificación clara
	InceptionResNetV2	0.93	0.93	0.93	Baja precisión y sensibilidad en las letras V y K
Últimos dos bloques entrenables	ResNet50	0.03	0.03	0.01	No llega a una clasificación clara
	MobileNetV2	0.19	0.2	0.2	No llega a una clasificación clara
	InceptionResNetV2	0.95	0.95	0.95	Predice la clase M como N

Tabla 8.2: Evaluación de la predicción de los modelos entrenados a partir de Base A en la Base A.

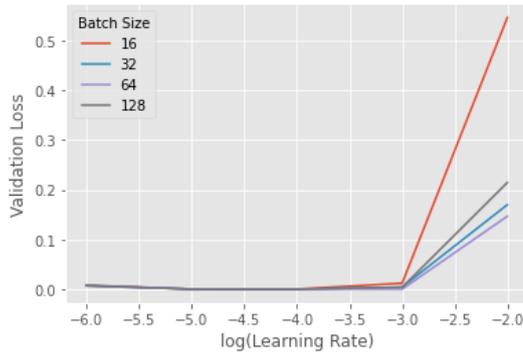
Capas entrenables	Modelo base	Base B			
		Acc	recall	f1 score	Observación inter clases
Ninguna capa entrenable	ResNet50V2	0.03	0.03	0	Clasifica todas las muestras como la clase L
	MobileNetV2	0.1	0.1	0.1	No llega a una clasificación clara
	InseptionResNetV2	0.54	0.52	0.54	Alta sensibilidad y precisión en la clase Nothing
Último bloque entrenable	ResNet50V2	0.03	0.03	0	Clasifica todas las muestras como la clase K
	MobileNetV2	0.13	0.13	0.13	No llega a una clasificación clara
	InseptionResNetV2	0.55	0.55	0.55	Alta sensibilidad y precisión en la clase Nothing
Últimos dos bloques entrenables	ResNet50V2	0.03	0.03	0	Clasifica todas las muestras como la clase J
	MobileNetV2	0.13	0.13	0.13	No llega a una clasificación clara
	InseptionResNetV2	0.52	0.52	0.5	Alta precisión en la clase Nothing

Tabla 8.3: Evaluación de la predicción de los modelos entrenados a partir de Base A en Base B.

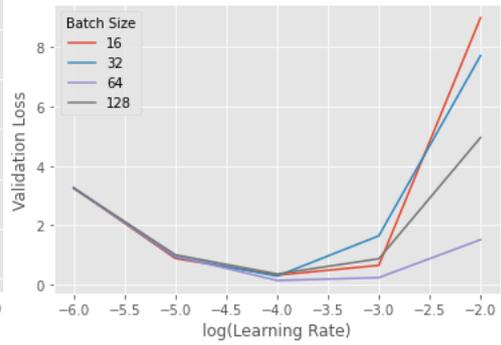
8.2. Resultados entrenamiento Conjunto de Datos B

8.2.1. Resultados exploración de parámetros

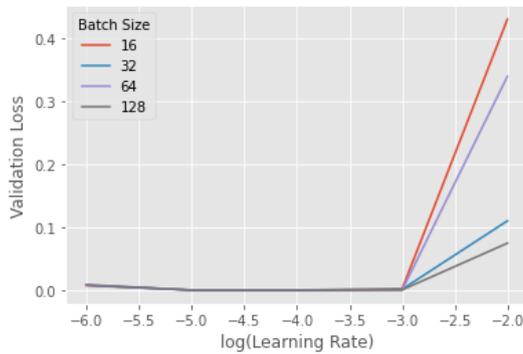
Al realizar la exploración de los hiper-parámetros y evaluar la mínima pérdida en validación para cada una de las combinaciones mencionados en la metodología, entrenados con conjunto datos Base A , se obtuvieron los resultados mostrados en las Figuras 8.6 y 8.7 así mismo como en la tabla 8.4 donde se muestra el resumen de la exploración.



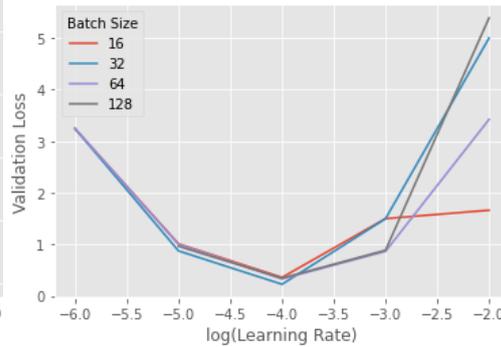
(a) Exploración NTL utilizando ResNet50V2 Base B



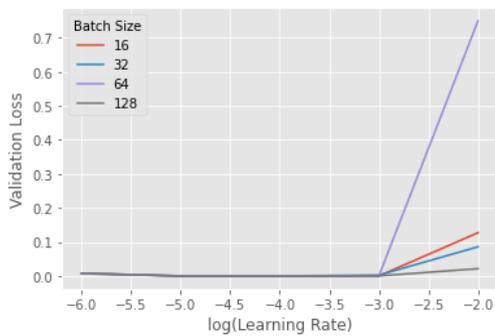
(b) Exploración NTL utilizando MobileNetV2 Base B



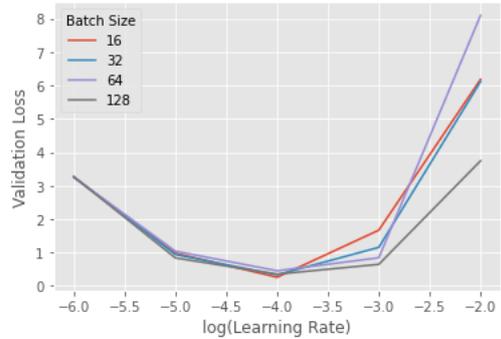
(c) Exploración TL1 utilizando ResNet50V2 Base B



(d) Exploración TL1 utilizando MobileNetV2 Base B



(e) Exploración TL2 utilizando ResNet50V2 Base B



(f) Exploración TL2 utilizando MobileNetV2 Base B

Figura 8.6: Resultados de la exploración de hiper-parámetros para la modelo utilizando ResNet50V2 Base B (izquierda) y MobileNetV2 Base B (derecha)

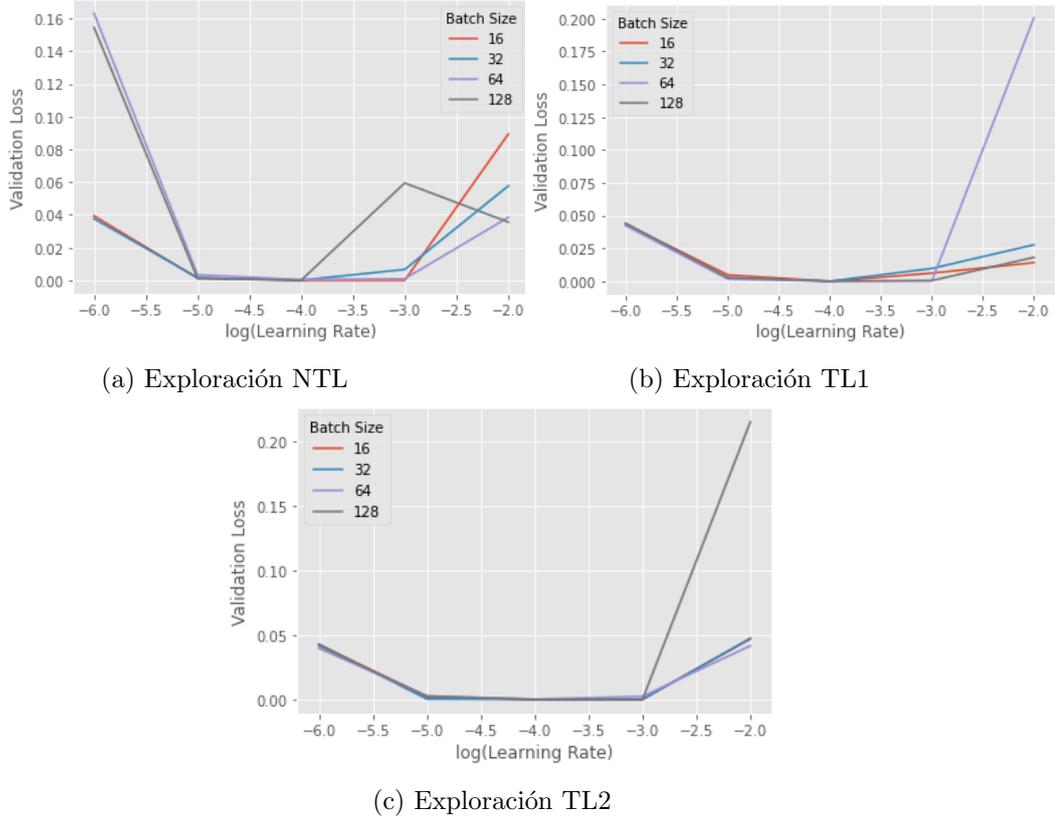
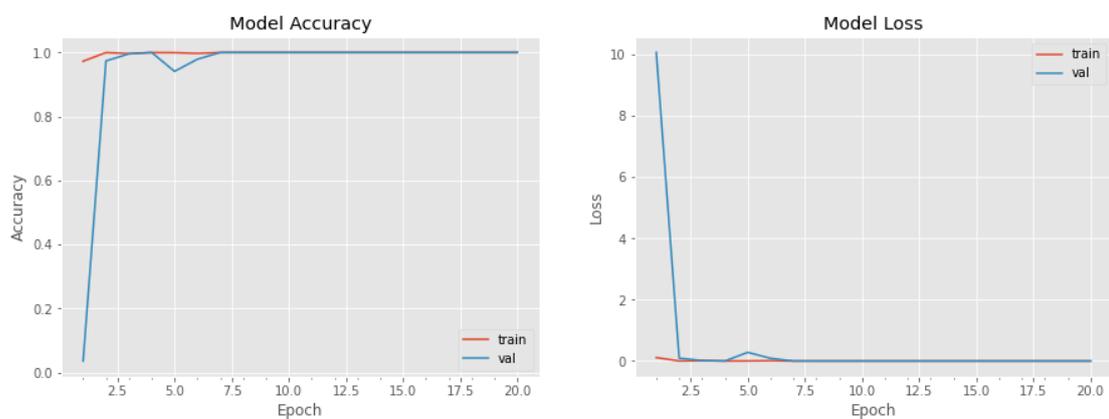


Figura 8.7: Resultados de la exploración de hiperparámetros para la modelo utilizando InceptionResNetV2 Base B

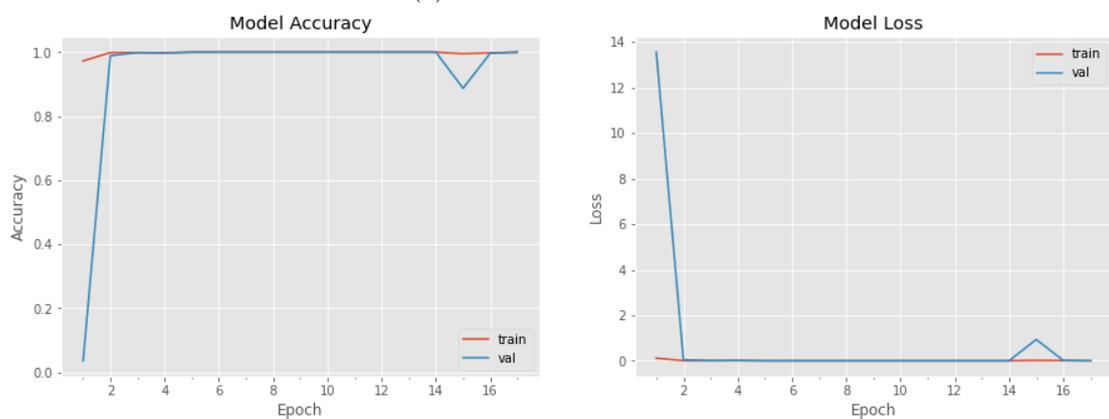
Capas entrenables	Modelo	BS	LR	Mínima pérdida validación	Tiempo de exploración (horas)
Ninguna capa entrenable	ResNet50V2	128	0.0001	$3.14e^{-6}$	27.5
	MobileNetV2	64	0.0001	0.1474	22.3
	InceptionResNetV2	16	0.0001	$0.2e^{-6}$	31.2
Último bloque entrenable	ResNet50V2	128	0.0001	$0.993e^{-6}$	35
	MobileNetV2	32	0.0001	0.2274	24.5
	InceptionResNetV2	16	0.0001	$5.98e^{-6}$	37
Últimos dos bloques entrenables	ResNet50V2	128	0.0001	$1.61e^{-6}$	35
	MobileNetV2	16	0.0001	0.2593	25
	InceptionResNetV2	16	0.0001	$0.24e^{-6}$	37

Tabla 8.4: Mejores hiper hiper-parámetros encontrados durante Exploración Conjunto de imágenes Base B.

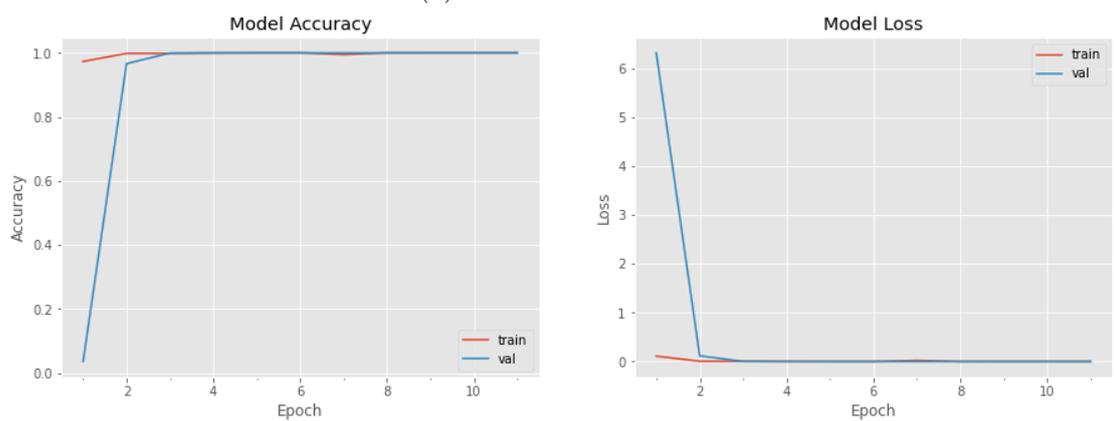
8.2.2. Resultados de entrenamiento de los modelos



(a) Entrenamiento NTL

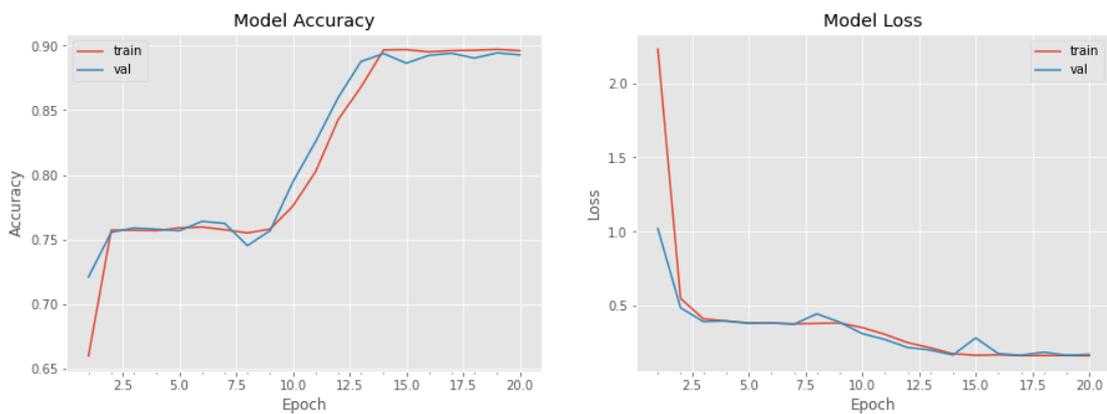


(b) Entrenamiento TL1

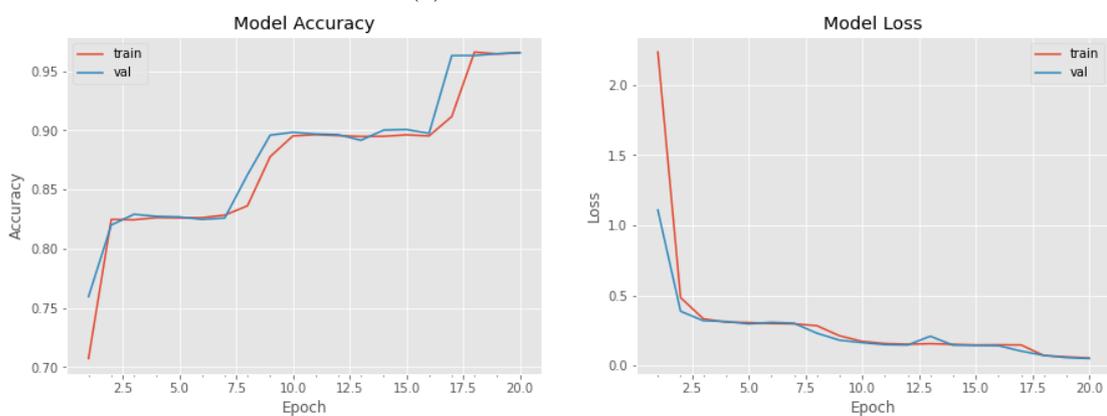


(c) Entrenamiento TL2

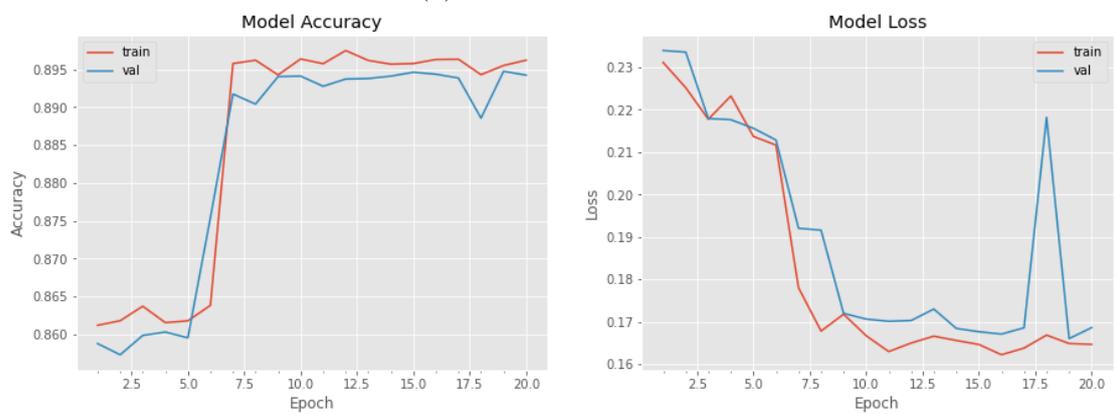
Figura 8.8: Comportamiento durante el entrenamiento redes basadas en ResNet50V2 Base B



(a) Entrenamiento NTL

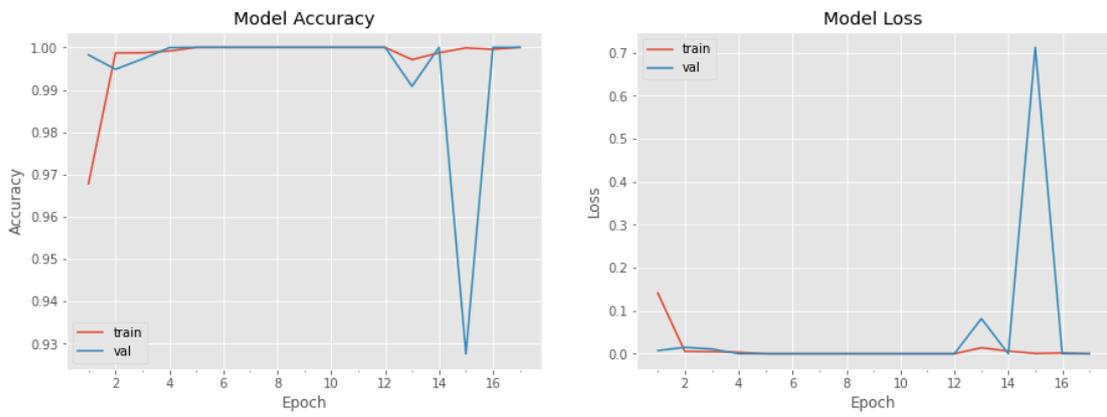


(b) Entrenamiento TL1

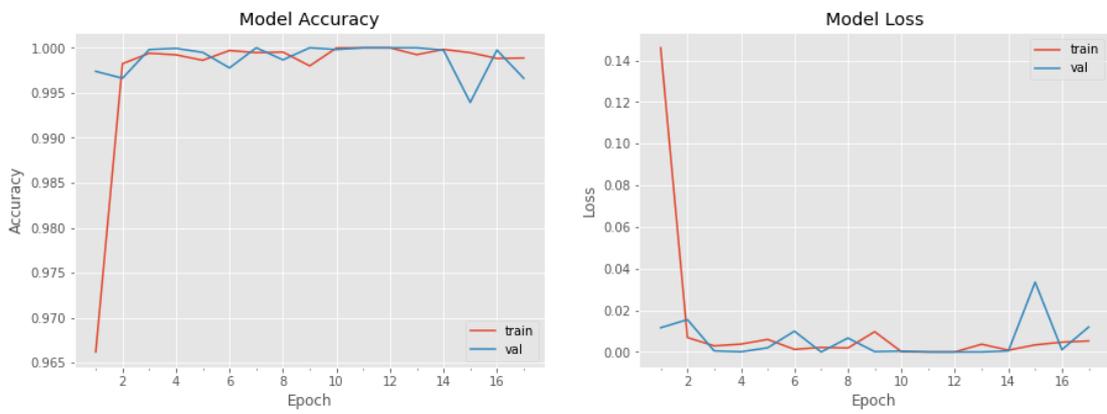


(c) Entrenamiento TL2

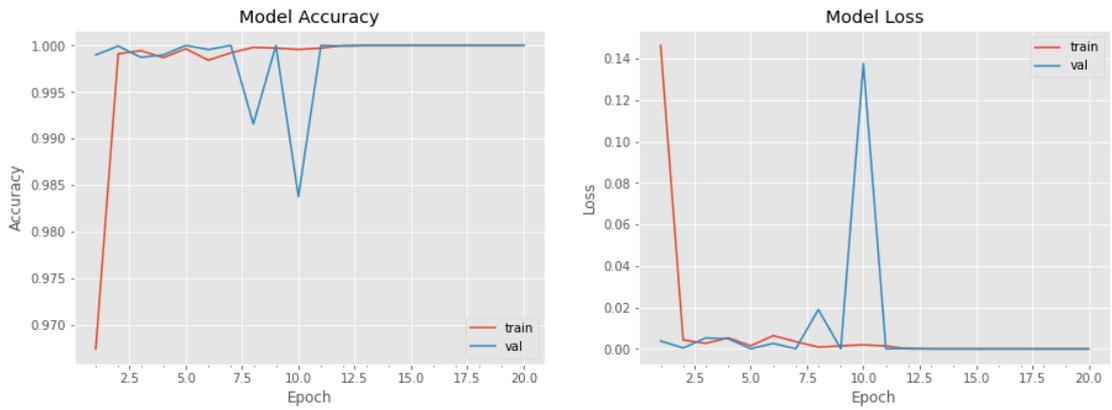
Figura 8.9: Comportamiento durante el entrenamiento redes basadas en MobileNetV2 Base B



(a) Entrenamiento NTL



(b) Entrenamiento TL1



(c) Entrenamiento TL2

Figura 8.10: Comportamiento durante el entrenamiento redes basadas en InceptionRes-NetV2 Base B

8.2.3. Evaluación de la predicción con modelos entrenados

Capas entrenables	Modelo base	Base B			
		Acc	recall	f1 score	Observación inter clases
Ninguna capa entrenable	ResNet50V2	0.99	0.99	0.99	No errores en la clasificación del conjunto de test
	MobileNetV2	0.89	0.9	0.9	No se logra identificar la letra K Confunde la letra U y V
	InseptionResNetV2	0.99	0.99	0.99	No errores en la clasificación del conjunto de test
Último bloque entrenable	ResNet50V2	0.99	0.99	0.99	No errores en la clasificación del conjunto de test
	MobileNetV2	0.95	0.97	0.95	No logra identificar la letra N
	InseptionResNetV2	0.99	0.99	0.99	No errores en la clasificación del conjunto de test
Últimos dos bloques entrenables	ResNet50V2	0.98	0.98	0.98	No errores en la clasificación del conjunto de test
	MobileNetV2	0.96	0.95	0.96	No distingue entre clases K y U
	InseptionResNetV2	0.99	0.99	0.99	No errores en la clasificación del conjunto de test

Tabla 8.5: Evaluación de la predicción de los modelos entrenados a partir de Base B en la Base B.

Capas entrenables	Modelo base	Base A			
		Acc	recall	f1 score	Observación inter clases
Ninguna capa entrenable	ResNet50V2	0.68	0.68	0.67	Logra clasificación correcta, en las letras P, J y T
	MobileNetV2	0.65	0.63	0.63	Logra clasificación correcta, en las letras P
	InseptionResNetV2	0.74	0.74	0.72	Logra clasificación correcta, en las letras P y Q
Último bloque entrenable	ResNet50V2	0.69	0.69	0.69	Logra clasificación correcta, en las letras P
	MobileNetV2	0.69	0.61	0.61	No logra identificar la letra N o Q
	InseptionResNetV2	0.74	0.74	0.74	Logra clasificación correcta, en las letras P, J
Últimos dos bloques entrenables	ResNet50V2	0.76	0.74	0.76	Clasifica correctamente las clases P y J
	MobileNetV2	0.72	0.71	0.72	Clasifica correctamente las clases U y V
	InseptionResNetV2	0.81	0.80	0.81	Predice la clase M como N

Tabla 8.6: Evaluación de la predicción de los modelos entrenados a partir de Base B en la Base A.

9. DISCUSIÓN

De acuerdo con los resultados mostrados en la sección anterior, se discutirán los hallazgos de la exploración de hiper-parámetros y evaluación de los modelos empleados, al ser entrenados con cada una de las bases de datos de imágenes del abecedario del ASL que se utilizaron analizando por separado (Base A y Base B). Todo lo anterior se hará centrado en las diferencias entre cada modelo y las posibles características de los mismos y del proceso de entrenamiento que lleven a una mejor clasificación en general y para cada clase de seña.

9.1. Conjunto entrenamiento Base A

Inicialmente, al realizar la exploración de hiper-parámetros para este data set se evidencio que los “BS” con el que la mayoría de modelos bajo estas primeras condiciones se entrenaban mejor fue el de 64 y 16 y así mismo los valores de “LR” más predominantes son los de 0.001 y 0.0001. Por otro lado, el modelo basado en la arquitectura Inception-ResNetV2 fue el que alcanzó una menor perdida en validación lo cual corrobora después con su uso y prueba en la porción de Prueba de los datos.

A partir de los resultados obtenidos de la evaluación de los modelos usando como base de datos para su entrenamiento el conjunto de Imágenes Base A, se logro identificar que para los modelos ResNet50V2 y MobileNetV2 no se logró generar un proceso de entrenamiento o aprendizaje por parte de la red neuronal de acuerdo con las características de las diferentes clases de imágenes. Esto se interpreta en la manera en la que ambas redes están construidas no logra identificar las características de cada seña ante una muestra de datos que se puede considerar reducida por comparado al numero de clases (30 imágenes por clase, 29 clases en total). Además al evaluar las imágenes de este data set, su diferencia en fondos y texturas dentro de la misma clase genera en cada proceso de aprendizaje las redes aprendan de la manera correcta si no tiene una estructura considerable de capas convencionales para la identificación de sus características.

En el caso de ResNet50V2 al tener un bajo volumen en el conjunto imágenes en cada ciclo de aprendizaje y dado su diseño que usa redes residuales profundas las cuales pueden escapar algunas capas de entrenamiento ante los valores de la salida de cada bloque teniendo en cuenta el error en la salida, de esta manera la diferencia en las imágenes de cada clase consigue que en cada bloque se presente un error alto y que el sistema no pueda lograr un aprendizaje, ya que los valores de entrada en cada bloque se verán influenciados por su salida. Esta misma situación puede pasar con el modelo basado en MobileNetV2, ya que este modelo también usa pequeños bloques convencionales conectados por conexiones residuales inversas dando alta importancia al error durante el proceso de aprendizaje del modelo.

Por el otro lado para este conjunto de datos el modelo basado en InceptionResNetV2

logró tener unos resultados considerablemente mejores comparado con los modelos anteriormente descritos, al lograr un 95 % de exactitud en la evaluación de la porción de prueba del mismo set de datos. Esto quiere decir que este modelo al contemplar estructuras más complejas entre sus bloques convencionales permite una identificación amplia de características ayudando a reducir el error residual que queda en sus interconexiones disminuyendo el error dentro de cada una de sus iteraciones a pesar de la diferencia en las imágenes del conjunto de datos.

Sin embargo, el modelo pre-entrenado presenta algunos problemas en la clasificación de imágenes en contextos y fondos distintos como lo es las del conjunto de datos Base B y logra solo un 52 % de exactitud en su clasificación. Este resultado al ser comparado con las diferentes técnicas expuestas en la literatura se puede determinar como bajo o no satisfactorio, ya que en su mayoría los modelos evaluados en revisiones sistemáticas [3] varían sus resultados de exactitud entre un 80-95 % en los conjuntos de prueba. Esto se da ya que utilizan algunas metodologías de procesamiento de imágenes adicionales antes de emplear los métodos de CNN, además de tener como fuente de entrenamiento bases de datos más robustas en términos del número de imágenes, ya que permite que el entrenamiento de estas redes pueda aprender más características de cada clase de imágenes que representa el ASL.

Adicionalmente, se identificó que los tiempos de exploración usualmente oscilan de 4 a 15 horas dependiendo la complejidad y cantidad de parámetros del modelo para completar las 20 posibles combinaciones de parámetros para cada uno de los modelos, esto ayuda a identificar que pese a que el modelo Basado en InceptionResNetV2 mostró los mejores resultados para este conjunto de datos, los tiempos de ejecución son elevados dada la complejidad de su estructura.

9.2. Conjunto entrenamiento Base B

Para el conjunto de datos Base B durante la exploración de hiper-parámetros inicialmente se evidenció que para el caso de todas las combinaciones que el “LR”, la que mejor funciona bajo esta base de datos es de 0.0001. Sin embargo, en el caso de los modelos basados en ResNet50V2 el valor de “BS” que mantuvo buenos resultados fue 128, indicando que esta red indica un número mayor de imágenes por iteración para poder aprender las características de cada clase. Por el contrario para los modelos basados en InceptionResNetV2, los menores valores de pérdida se encontraron al entrenar el modelo con un número menor de imágenes por iteración ya que el “BS” que se logró menores valores de pérdida fue el de 16. De otra manera, para las redes basadas en los modelos generados utilizando la estructura MobileNetV2 se ve una relación inversamente proporcional en el número de capas entrenables y el “BS” bajo el cual los modelos funcionan mejor, es decir entre más capas entrenables tiene la estructura del modelo es sus últimos bloques convoluciones esta estructura requiere un número menor de imágenes para lograr su entrenamiento durante cada iteración.

Por otro lado, de acuerdo con lo mostrado en la Tabla 8.5 en la predicción de los

modelos sobre imágenes del mismo tipo, se identifica que con el conjunto mas grande de datos (Base B) todos los modelos lograron una exactitud alta (mayor al 90 %), mostrando que las tres diferentes arquitecturas de redes aprenden de gran manera las características generales del tipo de imágenes en esta base de datos, en especial las arquitecturas basadas en InceptionResNetV2 y Resnet50V2 que llegaron a valores cercanos al 99 %. Además se evidencia que la arquitectura basada en MobilNetV2, con la cual se lograron los valores más bajos en las métricas evaluadas, tiene algunos errores al clasificar imágenes de señas que representan letras como la K y la U, y que entre mayor sea la cantidad de capas que se puedan reentrenar a partir de las características de las imágenes de esta base de datos, la arquitectura basada en MobilNetV2 puede mejorar su precisión y exactitud.

En el caso de la evaluación de los modelos entrenados con la Base B en un conjunto de imágenes diferente como lo es la Base A, se ve que la exactitud de todos los modelos baja en un 20 % al 30 % en comparación a la predicción de imágenes de la Base B anteriormente descrita. Por lo tanto el desempeño en los modelos al predecir imágenes de la base de datos Base A está por debajo del 80 % esperado para todos los modelos, menos el basado en la arquitectura de InceptionResNetV2 con la mayor cantidad de capas configuradas para entrenarse. Esta configuración logra alcanzar una precisión y exactitud mayor igual al porcentaje mencionado, sin embargo tiene problemas al clasificar las imágenes que representan la letra M y N, ya que por su parecido las interpreta como todas bajo la categoría de la letra N. Ya que solo uno de los modelos logro más del 80 %, se puede utilizar la información colectada en términos de estructura y características de entrenamiento, para en futuros diseños poder implementar algunos otros métodos de ajuste fino que permitan ajustar más la predicción de todos los diseños a imágenes que no se hallan visto durante el entrenamiento.

Característica	Valores con mejor desempeño
Arquitectura	InceptionResNetV2
Base de datos para entrenamiento	Base B
BS	16
LR	0.0001
Capas entrenables	Ultimos dos bloques

Tabla 9.1: Variables con mejor desempeño dentro de la comparación de modificaciones en arquitectura e hiper-parámetros descritos en cada característica

Además, para el conjunto de datos Base B durante la exploración de hiper-parámetros inicialmente se evidenció, que los tiempos de ejecución para esta exploración fueron bastante elevados dado la cantidad de imágenes y parámetros en cada uno de los modelos explorados, esta información es de gran importancia para futuros trabajos. Además, se evidencia que el modelo InceptionResNetV2, a pesar de ser el que mejores resultados en las imágenes que logra clasificar, es el que más tiempo demora en entrenarse dada la

complejidad de su estructura.

Finalmente, en el aspecto metodológico se encontró que la estrategia que permitió una mejor clasificación de este tipo de imágenes, es aplicar redes con arquitectura robustas, ya que este tipo de redes pueden adaptarse mejor a los diferentes volúmenes de las bases de datos alimentando el sistema. Adicionalmente, se confirmó la importancia de hacer procesos de exploración para definir las configuraciones de hiper-parámetros y ajuste fino a usar, en particular para la clasificación de imágenes del abecedario de ASL, se encontró que en arquitecturas robustas como InceptionResNetV2 la mejor estrategia es utilizar valores bajos de “BS”, “LR” y mayor cantidad de bloques entrenables. Por último, para encontrar la mejor combinación de parámetros, se realizó la evaluación de las diferentes configuraciones en términos de arquitecturas e hiper-parámetros y se encontró que la estrategia que generaba un mejor desempeño durante la clasificación, para ambos conjuntos de imágenes, esta determinada por los valores descritos en la 9.1

10. RECOMENDACIONES Y TRABAJOS FUTUROS

En primer lugar y a partir de los resultados de la evaluación de los modelos en el conjunto de datos Base A, se recomienda en futuros trabajos que uno de los mayores diferenciadores al momento de escoger un modelo en este tipo de clasificaciones de imágenes es que tan amplio es el conjunto de datos en comparación al número de clases a clasificar, ya que esto indicará la robustez en términos de arquitectura que necesita el modelo para extraer las características de cada imagen de la mejor manera sin generar una propagación del error.

Además se da el caso de una base de imágenes de un volumen bajo se puede sugerir el uso de redes robustas en su número de convoluciones y capas de extracción de características, o usar redes con un modelo más sencillo y simplificado y usar técnicas de "Aumento de datos" para entrenar el modelo, ya que estas técnicas utilizan metodologías para tratar las imágenes (rotaciones de la imagen, cambio de color) para ampliar el conjunto de datos incluyendo ejemplos de las mismas imágenes iniciales con estas modificaciones implementadas [41].

Por otro lado, a partir de los resultados obtenidos de el entrenamiento con la Base B se puede decir que al usar un volumen de imágenes mas elevado los modelos basados en la arquitectura ResNet50V2 y MobileNetV2 logran mejorar su rendimiento y pueden ser utilizados para el problema de clasificación para el lenguaje de ASL. Aun así se recomienda utilizar algunos otros métodos de ajuste fino o capas agregadas que permitan que el modelo de aprendizaje no se sature o genere un sobre aprendizaje y solo se especialice de más en la clasificación de solo las imágenes que pertenecen a el conjunto de datos con el que se ha entrenado.

Finalmente, se identifica que InceptionResNetV2 es el modelo más exacto y sensible dentro de las pruebas realizadas para un modelo de clasificación de ASL, sin embargo, para futuras aplicaciones del tipo móvil este modelo tiene características como su peso computacional, tiempo de entrenamiento y exploración que pueden hacer que este sistema no sea la mejor opción en un sistemas donde se busque un uso mínimo de uso en los recursos. Los otros dos modelos estudiados pueden ser utilizados para dichas aplicaciones que usan menor cantidad de recursos, sin embargo se debe tener en cuenta de acuerdo a los resultados obtenidos, que los dos factores que pueden afectar la predicción de estas arquitecturas ampliamente los cuales son el tamaño del conjunto de datos y su relación con la cantidad de clases, y la cantidad de capas que se entrenan ya que este último factor puede hacer que variar los hiper-parámetros bajo los cuales el modelo mejor trabaja como se observó en la arquitectura MobileNetV2 para el conjunto de datos Base B.

11. CONCLUSIONES

Al evaluar de manera sistemática las arquitecturas propuestas fueron desarrollados varios modelos de clasificación de imágenes utilizando diferentes combinaciones y estructuras de CNN, encontrando que los modelos con mejores resultados a lo largo de la evaluación dados diferentes combinaciones de hiper-parámetros y grupos de datos son los que se basan en la arquitectura InceptionResNetV2. Es por esto que se identifico como la mejor estrategia. Sin embargo, dado del peso computacional que genera este modelo, y su tiempo de exploración y predicción este modelo puede no ser tan útil en aplicaciones móviles o que demanden una velocidad de predicción elevada.

Tras comparar los diferentes modelos bajo cada una de las modificaciones de la arquitectura e hiper-parámetros, se encontró que la mejor estrategia dentro de las desarrolladas en esta investigación, para clasificar imágenes de señas del abecedario de ASL, es utilizar la red InceptionResNetV2 configurado con un valor de (BS) bajo para permitir que el modelo utilice pocas imágenes durante su proceso de entrenamiento, además de un valor de tasa de aprendizaje (LR) de 0.0001 puesto que genero menor perdida para todas las exploraciones de esta arquitectura. Adicionalmente, se identifico que entre mayor sea la cantidad de capas que puedan ser entrenadas dentro de esta arquitectura, los resultados en las métricas son mejores.

Después de hacer una revisión sistemática de las diferencias en los entrenamientos e hiper-parámetros de los modelos dados en contextos donde el proceso de aprendizaje se daba en diferentes conjuntos de datos (Base A y Base B), se encontró que para este tipo de implementaciones donde se hace un proceso de transferencia del aprendizaje de otros modelos antes entrenados, es necesario una muestra representativa de imágenes por cada clase a clasificar ya que al tener pocas muestras para ajustar múltiples parámetros el modelo no podrá aprender de manera correcta patrones significativos en cada clase.

Al evaluar sistemáticamente los modelos estudiados en los dos diferentes conjuntos de datos de señas del alfabeto de la ASL, se concluyo la gran importancia del tamaño del conjunto de datos para entrenamiento y su relación con el numero de clases de clasificación. Esta cantidad de datos puede considerarse dependiendo de la estructura y profundidad de las capas de convolución y extracción de características que compongan la red del modelo base.

11. REFERENCIAS

- [1] Banco de la República. *Lengua y lenguaje - Enciclopedia — Banrepcultural*. 2017. URL: https://enciclopedia.banrepcultural.org/index.php?title=Lengua_y_lenguaje.
- [2] Haitham Badi y Sabah Hussein. “Hand posture and gesture recognition technology”. En: *Neural Computing and Applications* 25 (2014), págs. 871-878. DOI: 10.1007/s00521-014-1574-4.
- [3] Ankita Wadhawan y Parteek Kumar. “Sign Language Recognition Systems: A Decade Systematic Literature Review”. En: *Archives of Computational Methods in Engineering* 28.3 (2021), págs. 785-813. ISSN: 18861784. DOI: 10.1007/s11831-019-09384-2. URL: <https://doi.org/10.1007/s11831-019-09384-2>.
- [4] Laura Dipietro, Angelo M Sabatini y Paolo Dario. “A Survey of Glove-Based Systems and Their Applications”. En: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38.4 (2008), págs. 461-482. DOI: 10.1109/TSMCC.2008.923862.
- [5] G Drew Kessler, Larry F Hodges y Neff Walker. *Evaluation of the CyberGlove as a Whole-Hand Input Device*. Inf. téc.
- [6] Munir Oudah, Ali Al-Naji y Javaan Chahl. “Hand Gesture Recognition Based on Computer Vision: A Review of Techniques”. En: *Journal of Imaging* 6.8 (2020). ISSN: 2313433X. DOI: 10.3390/JIMAGING6080073.
- [7] Qutaishat Munib y col. “American sign language (ASL) recognition based on Hough transform and neural networks”. En: *Expert Systems with Applications* 32.1 (2007), págs. 24-37. ISSN: 09574174. DOI: 10.1016/j.eswa.2005.11.018.
- [8] Alvaro David Orjuela-Cañon y col. “On the Use of Neuroevolutionary Methods as Support Tools for Diagnosing Appendicitis and Tuberculosis”. En: *Springer Nature* (2018), págs. 171-181.
- [9] V S Kulkarni y a D S D Lokhande. “Appearance Based Recognition of American Sign Language Using Gesture Segmentation”. En: *International Journal on Computer Science and Engineering IJCSE* 2.03 (2010), págs. 560-565. ISSN: 09753397. URL: <http://www.enggjournals.com/ijcse/doc/IJCSE10-02-03-33.pdf>.
- [10] Yuntao Cui y Juyang Weng. “Appearance-based hand sign recognition from intensity image sequences”. En: *Computer Vision and Image Understanding* 78.2 (2000), págs. 157-176. ISSN: 10773142. DOI: 10.1006/cviu.2000.0837.
- [11] Luis Quesada, Gustavo López y Luis Guerrero. “Automatic recognition of the American sign language fingerspelling alphabet to assist people living with speech or hearing impairments”. En: *Journal of Ambient Intelligence and Humanized Computing* 8.4 (2017), págs. 625-635. ISSN: 18685145. DOI: 10.1007/s12652-017-0475-7.

- [12] Riccardo Miotto y col. “Deep learning for healthcare: review, opportunities and challenges”. En: *Brief Bioinform* 19.6 (2018), págs. 1236-1246. DOI: 10.1093/bib/bbx044. URL: <https://academic.oup.com/bib/article/19/6/1236/3800524>.
- [13] Mohammed Mustafa. “A study on Arabic sign language recognition for differently abled using advanced machine learning classifiers”. En: *Journal of Ambient Intelligence and Humanized Computing* 12.3 (2021), págs. 4101-4115. ISSN: 18685145. DOI: 10.1007/s12652-020-01790-w. URL: <https://doi.org/10.1007/s12652-020-01790-w>.
- [14] Pisit Nakjai y Tatpong Katanyukul. “Hand Sign Recognition for Thai Finger Spelling: an Application of Convolution Neural Network”. En: *Journal of Signal Processing Systems* 91.2 (2019), págs. 131-146. ISSN: 19398115. DOI: 10.1007/s11265-018-1375-6.
- [15] Nelson Ortiz-Farfán y Jorge E. Camargo-Mendoza. “Modelo computacional para reconocimiento de lenguaje de señas en un contexto colombiano”. En: *TecnoLógicas* 23.48 (2020), págs. 197-232. ISSN: 0123-7799. DOI: 10.22430/22565337.1585.
- [16] Robinson Steven Castro. “Aplicativo para apoyar el proceso de aprendizaje del lenguaje de señas hacia un oyente mediante Microsoft Kinect”. Tesis doct. Universidad Piloto de Colombia, 2015.
- [17] López Triviño Camilo Iván. “Sistema para el aprendizaje del lenguaje de señas colombiano usando visión por computador”. Tesis doct. Universidad de La Salle, 2018.
- [18] Ministerio de Educacion Nacional. *Diccionario Básico de la Lengua de Señas Colombiana*. Inf. téc. Instituto Nacional para Sordos, 1996, pág. 325.
- [19] Naciones Unidas. *Día Internacional de las Lenguas de Señas — Naciones Unidas*. 2022. URL: <https://www.un.org/es/observances/sign-languages-day>.
- [20] Ministerio de Educacion Nacional de Colombia. *Plan Estratégico Institucional IN-SOR*. 2022.
- [21] Ministerio de Tecnologías de la Información y las Comunicaciones de Colombia. *Servicio de Interpretación en línea SIEL*. 2022. URL: <https://centroderelievo.gov.co/632/w3-propertyvalue-15254.html>.
- [22] Yenny Milena Cortés Bello y Alex Giovanny Barreto Muñoz. “Variacion Sociolingüística En La Lengua D Señas Colombiana”. En: *Forma y Función* 26 (2013), págs. 149-170. URL: <http://www.scielo.org.co/pdf/fyf/v26n2/v26n2a07.pdf>.
- [23] IBM Cloud Education. *¿Qué es deep learning? - México — IBM*. 2020. URL: <https://www.ibm.com/mx-es/cloud/deep-learning>.
- [24] IBM Cloud Education. *¿Qué son las redes neuronales? - España — IBM*. 2020. URL: <https://www.ibm.com/es-es/cloud/learn/neural-networks>.

- [25] Towards Data Science. *How To Teach A Computer To See With Convolutional Neural Networks* — by Alex Yu — *Towards Data Science*. 2018. URL: <https://towardsdatascience.com/how-to-teach-a-computer-to-see-with-convolutional-neural-networks-96c120827cd1>.
- [26] Julia García Salinero. *Estudios descriptivos*. 2004. DOI: 10.1016/b978-84-8174-709-6.50009-9.
- [27] Dan Rasband. *ASL Alphabet Test* — *Kaggle*. 2018. URL: <https://www.kaggle.com/datasets/danrasband/asl-alphabet-test>.
- [28] Akash Nagaraj. *ASL Alphabet* — *Kaggle*. 2018. URL: https://www.kaggle.com/datasets/grassknotted/asl-alphabet?select=asl_alphabet_test.
- [29] Python Software Foundation. *Welcome to Python.org*. 2022. URL: <https://www.python.org/>.
- [30] TensorFlow. *Educación sobre aprendizaje automático* — *TensorFlow*. 2022. URL: <https://www.tensorflow.org/resources>.
- [31] Scikit Learn. *scikit-learn: machine learning in Python* — *scikit-learn 1.1.1 documentation*. 2022. URL: <https://scikit-learn.org/stable/>.
- [32] Keras. *Keras Applications*. 2022. URL: <https://keras.io/api/applications/>.
- [33] NVIDIA. *Driver persistence*. 2020.
- [34] Yu Bai y col. *How Important is the Train-Validation Split in Meta-Learning?* *Inf. téc.* 2021, pág. 139.
- [35] Nikunj Saunshi, Arushi Gupta y Wei Hu. *A Representation Learning Perspective on the Importance of Train-Validation Splitting in Meta-Learning*. *Inf. téc.* 2021.
- [36] Rajiv Khosla, Robert J Howlett y Lakhmi C Jain. *Lecture Notes in Artificial Intelligence 3684 Subseries of Lecture Notes in Computer Science*. *Inf. téc.* 2005, págs. 76-83.
- [37] Stanford Vision Lab. *ImageNet*. 2020. URL: <https://www.image-net.org/challenges/LSVRC/>.
- [38] Kaiming He y col. “Deep residual learning for image recognition”. En: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2016-Decem. 2016, págs. 770-778. ISBN: 9781467388504. DOI: 10.1109/CVPR.2016.90.
- [39] Andrew G. Howard y col. “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”. En: *Computer Science*. 2017. URL: <http://arxiv.org/abs/1704.04861>.
- [40] Christian Szegedy y col. “Inception-v4, inception-ResNet and the impact of residual connections on learning”. En: *31st AAAI Conference on Artificial Intelligence, AAAI 2017*. 2017, págs. 4278-4284.

- [41] Connor Shorten y Taghi M. Khoshgoftaar. “A survey on Image Data Augmentation for Deep Learning”. En: *Journal of Big Data* 6.1 (dic. de 2019), pág. 60. ISSN: 2196-1115. DOI: 10.1186/s40537-019-0197-0.