IMPLEMENTACIÓN DE UN ALGORITMO PARA LA CLASIFICACIÓN AUTOMÁTICA DE LENGUAJE DE SEÑAS COLOMBIANO EN VIDEO USANDO APRENDIZAJE PROFUNDO.

YURY PAOLA MUÑOZ REINA - 625536
LUIS FELIPE MORENO - 625367

UNIVERSIDAD CATÓLICA DE COLOMBIA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
MODALIDAD TRABAJO DE INVESTIGACIÓN TECNOLÓGICA
BOGOTÁ D.C.
2020

IMPLEMENTACIÓN DE UN ALGORITMO PARA LA CLASIFICACIÓN AUTOMÁTICA DE LENGUAJE DE SEÑAS COLOMBIANO EN VIDEO USANDO APRENDIZAJE PROFUNDO.

YURY PAOLA MUÑOZ REINA - 625536
LUIS FELIPE MORENO - 625367

ESTE TRABAJO DE GRADO ES PRESENTADO COMO REQUISITO PARA OPTAR AL TÍTULO DE: INGENIERO DE SISTEMAS

ASESOR:
ROGER ENRIQUE GUZMÁN AVENDAÑO MSC. INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

ALTERNATIVA:
TRABAJO DE INVESTIGACIÓN TECNOLÓGICA

UNIVERSIDAD CATÓLICA DE COLOMBIA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
MODALIDAD TRABAJO DE INVESTIGACIÓN TECNOLÓGICA
BOGOTÁ D.C.
2020

Nota de aceptación

Aprobado por el comité de grado en cumplimiento de los requisitos exigidos por la Facultad de Ingeniería y la Universidad Católica de Colombia para optar al título de ingeniero de sistemas.

John Velandia Prof. M.Sc. Eng.

Jurado 1

Roger Enrique Guzman Avendaño, Msc

Asesor

BOGOTÁ D.C. JUNIO 13 DE 2020

DEDICATORIA

A nuestras familias, nuestros padres y hermanos que siempre nos apoyaron y nos dieron una voz de aliento para seguir adelante en momentos difíciles y a la memoria de aquellas personas que ya no están con nosotros.

AGRADECIMIENTOS

A la universidad católica de Colombia y a todos nuestros profesores, en especial a nuestro tutor de trabajo de grado Roger Guzmán, por su comprensión, apoyo, enseñanzas y quien compartió sus conocimientos a lo largo de nuestra carrera.

A nuestro compañero David quien nos facilitó herramientas técnicas y conocimientos que hicieron posible llevar a cabo este proyecto.

A nuestras familias por creer en nosotros y apoyarnos de todas las maneras posibles a lo largo de nuestra vida profesional.

TABLA DE CONTENIDO

ÍNDICE DE ILUSTRACIONES

ÍNDICE DE TABLAS

RESUMEN

Este trabajo de grado nace de la necesidad de apoyar a la población sordomuda de la ciudad de Bogotá, que requiere interactuar con la sociedad logrando disminuir la brecha de comunicación existente entre las personas con discapacidad auditiva y la sociedad. Por este motivo se realiza un experimento de un método sustentado en aprendizaje profundo, con el cual se implementan distintos algoritmos computacionales, que permitieron la identificación de movimientos y características emuladas en los videos compuestos por frames (imágenes extraídas de los videos); el entrenamiento y aprendizaje de este set de datos se realiza por medio de una red neuronal convolucional la cual clasifica los videos, determinando a que clases pertenecen las palabras del lenguaje de señas colombiano grabadas.

Se plantea una metodología sustentada en el estado del arte, que contiene seis etapas principales las cuales son: construcción del conjunto de datos, pre procesamiento, muestreo, implementación de la red neuronal, medición y clasificación. Al llegar a la etapa final se obtiene un método de clasificación automático de las cinco palabras utilizadas en el proyecto, que tienen resultados superiores al setenta por ciento (70%) para las métricas de desempeño generadas en la ejecución del experimento.

Palabras Claves: Aprendizaje de máquina, algoritmo, aprendizaje profundo en videos, lengua de señas colombiana (LSC), red neuronal, GPU.

ABSTRACT

This undergraduate work arises from the need to support the deaf and dumb population of the city of Bogotá, which requires interacting with society, reducing the existing communication gap between people with hearing disabilities and society. For this reason, we did to experiment of a method based on deep learning was carried out, with which different computational algorithms were implemented, which allowed the identification of movements and characteristics emulated in the videos composed of frames (images extracted from the videos); The training and learning of this data set was carried out by means of a convolutional neural network which classified the videos, determining to which classes the recorded Colombian sign language words belonged.

A methodology based on the state of the art is proposed, which contains (6) fundamental stages, these are: construction of the data set, pre-processing, sampling, implementation of the neural network, measurement and classification. Upon reaching the final stage, an automatic classification method is obtained for the five words used in the project, which have results greater than seventy percent (70%) for the performance metrics generated in the execution of the experiment

Key Words: Machine learning, algorithm, deep video learning, Colombian Sign Language (LSC), Neural Red, GPU.

INTRODUCCIÓN

El número de personas con limitaciones de la voz y el habla registradas por el DANE en el censo de población realizado en el año 2010 es de 317.195, de los cuales 153.335 son mujeres y 163.860 hombres[1]. Si bien el porcentaje de población sordomuda no supera el 2% en el país es necesario garantizar el desarrollo integral de esta minoría permitiéndoles una integración a la sociedad de manera efectiva con el resto de colombianos, en el mes de noviembre del año 2016 del total de la población con discapacidad auditiva mayor de 18 años, que corresponde a un total de 136.498 personas, solo el 12% se encuentra trabajando de manera formal, cifra que equivale a un total de 16.966 personas.

Aproximadamente el 80% de las personas sordas se encuentran sin contrato, lo que refleja una grave problemática en relación a la formalidad de las condiciones idóneas socio-laborales de la población en condición de discapacidad auditiva.[2] En el ámbito académico la población sordomuda registrada con matrículas formales en instituciones de educación superior IES según cifras oficiales del INSOR de un total de 168 IES encuestadas, 62 IES reportaron contar actualmente con estudiantes sordos matriculados, lo que corresponde al 37%.[3] Basados en las estadísticas mencionadas se evidencia una problemática social que afecta directamente a la población sordomuda al no poder acceder a los servicios de educación y trabajo digno por no lograr comunicarse efectivamente con su interlocutor, por este motivo se realiza una implementación de un algoritmo basado en aprendizaje profundo en fotogramas de videos fundamentado en inteligencia artificial que permite clasificar cinco palabras tomadas del diccionario de lenguaje de señas colombiano disponible en la página del INSOR de la categoría “Hombre” ya que como lo indica el diccionario “Este primer campo recoge aquellos aspectos que determinan al hombre en su desarrollo físico, psíquico y sentimental”[4], adicional a este criterio se tuvo en cuenta que las palabras escogidas no fueran representadas en el LSC con más de un movimiento ya que el alcance del proyecto así lo determina, de igual forma se eligieron solo cinco palabras teniendo en cuenta la alta complejidad computacional que con lleva procesar por GPU los fotogramas extraídos y parametrizados de los videos obtenidos.

[1] Tomado de la pagina https://www.dane.gov.co/index.php/estadisticas-por-tema/salud/discapacidad

[2] INSOR instituto nacional para sordos. *INSOR instituto nacional para sordos.* [En línea] 2017. [Citado el: 21 de 09 de 2019.] http://www.insor.gov.co/bides/info-general/.

[3] INSOR. 2017. INSOR. INSOR. [En línea] 02 de 2017. [Citado el: 21 de 10 de 2019.] http://www.insor.gov.co/bides/wp-content/uploads/archivos/caracterizacion_acceso_perm_grad_estudiantes_sordos_ies.pdf.

[4] 2011**.** INSOR Instituto Nacional Para Sordos. *INSOR Instituto Nacional Para Sordos.* [En línea] Instituto Caro y Cuervo, Instituto Nacional para sordos (Insor), 2011. [Citado el: 21 de 10 de 2019.] http://www.insor.gov.co/descargar/diccionario_basico_completo.pdf.

Apoyados en esta información suministrada por el INSOR las primeras palabras de la categoría “Hombre” que cumplen los criterios mencionados son: *anciano, bebe, hombre, homosexual y joven.*[5]

En este documento se presenta una estrategia metodológica sustentada en el estado de arte que contiene las siguientes fases de ejecución: construcción del conjunto de datos en videos, procesamiento de fotogramas y/o imágenes, entrenamiento de la red neuronal, muestreo y clasificación, finalizando con la medición del desempeño del algoritmo desarrollado.

[5] 2011**.** INSOR Instituto Nacional Para Sordos. *INSOR Instituto Nacional Para Sordos.* [En línea] Instituto Caro y Cuervo, Instituto Nacional para sordos (Insor), 2011. [Citado el: 21 de 10 de 2019.] http://www.insor.gov.co/descargar/diccionario_basico_completo.pdf.

1. GENERALIDADES

1.1 LÍNEA DE INVESTIGACIÓN

Trabajo de investigación tecnológica.

1.2 PLANTEAMIENTO DEL PROBLEMA

1.2.1 Descripción del problema

La población sordomuda actualmente presenta dificultades de inclusión social al acceder a los servicios estatales como la salud, educación, trabajo entre otros. La discapacidad auditiva es contemplada como una condición integral de la persona, la cual excede cualquier ámbito sectorial, al momento de su atención. Se evidencia que actualmente la problemática que presenta la comunidad sordomuda adicional a los aspectos sociales como la educación o el acceso a trabajos o atención primaria de salud, también radica en su diario vivir, las personas sordomudas por su limitación al escuchar y hablar no logran comunicarse en una tienda para adquirir algún producto, o un restaurante, inclusive solicitar alguna ayuda en caso de emergencia.

A pesar del trabajo realizado por el gobierno nacional en disminuir las brechas de comunicación que tiene la comunidad sordomuda en el país aún queda bastante trabajo por realizar, en los últimos años diversas instituciones privadas y públicas han realizado esfuerzos para mitigar esta problemática, un ejemplo de estas iniciativas por parte de universidades privadas como la ECCI que ha trabajado de la mano con el ministerio de educación y la comunidad sordomuda permitiendo la graduación de la primera promoción de pregrado en la carrera de mercadeo y publicidad[6]. Por otra parte las instituciones públicas del estado como el SENA que ha integrado en sus cursos de formación a la comunidad sordomuda en Colombia[7].

Esto ha permitido ubicar a la sociedad sordomuda como actores principales y participantes activos en el diseño e implementación de desarrollos tecnológicos realizando aplicaciones las cuales permiten una interacción entre el sujeto y su medio, como lo son: el software “Hablando con Julis, creadora de un software que permite a personas con discapacidad comunicarse sin intérprete, la cual fue seleccionada como finalista del MassChallenge, el Mundial del Emprendimiento que se celebra cada año en

[6] INSOR Instituto nacional para sordos. *INSOR Instituto nacional para sordos.* [En línea] INSOR , 01 de 2012. [Citado el: 21 de 10 de 2019.] http://www.insor.gov.co/bides/experiencia-universidad-ecci/.

[7] SENA, S. N. (10 de 2017). Periódico SENA. (Servicio Nacional de Aprendizaje SENA) Recuperado el 11 de 10 de 2019, de Periódico SENA: http://periodico.sena.edu.co/inclusion-social/noticia.php?t=comunidad_sorda&i=86

la ciudad estadounidense de Boston"[8], y SIEL: Software colombiano para comprender el lenguaje de signos disponible para hospitales, centros de salud, EPS y consultorios médicos que les ayudará a comunicarse con personas con discapacidades auditivas,[9] si bien se han realizado de manera exitosa el desarrollo de estas aplicaciones basadas en un marco tecnológico, es necesario seguir impulsando la innovación de nuevas herramientas las cuales faciliten la libre comunicación de la comunidad sordomuda.

Mediante la ley 324 de 1996 se establece el término lengua manual colombiana definiéndose como: es la que se expresa en la modalidad viso-manual, un código cuyo medio de transmisión es visual no auditivo; como cualquier otra lengua tiene su propio vocabulario, expresiones idiomáticas, gramáticas, sintaxis diferentes del español. Los elementos de esta lengua (las señas individuales) son la configuración, la posición y la orientación de las manos en relación con el cuerpo y con el individuo. La lengua también utiliza el espacio, dirección y velocidad de movimientos, así como la expresión facial para ayudar a transmitir el significado del mensaje, esta es una lengua viso gestual[10]. El principal reto es lograr la clasificación de señales viso gestuales de manera automática ya que tienen distintas gesticulaciones realizadas por el emisor que no solo se ejecutan con las manos y brazos si no con distintas partes del cuerpo para lograr comunicarse efectivamente con el receptor. La señal expresada por la persona contiene gestos que son representados en video, dado que estos movimientos efectuados por el emisor están sujetos a la subjetividad del entendimiento por parte del receptor y al ser representados en videos se complica la interpretación de las señas recibidas por la máquina.

[8] Barros**,.** El Universal. *El Universal.* [Online] El Universal, 28 mayo 2014. [Cited: 20 04 2019.] https://www.eluniversal.com.co/tecnologia/software/software-para-sordos-finalista-en-mundial-de-emprendimiento-160945-DXEU253808.

[9] Bécares, Bárbara. 2015. siliconweek. *siliconweek.* [En línea] 20 de 05 de 2015. [Citado el: 15 de 01 de 2020.] https://www.siliconweek.com/cloud/siel-software-colombiano-para-comprender-el-lenguaje-de-signos-59195.

[10] Colombia, El Congreso de. LEY 324 DE 1996. Bogotá: Congreso, 1996.

1.2.2 Formulación del problema

De acuerdo con el problema explicado anteriormente, se plantea la siguiente pregunta de investigación:

¿Cómo puede la tecnología apoyar la clasificación de palabras en lenguaje de señas colombiano grabadas en video de forma automática utilizando algoritmos de aprendizaje profundo?

1.3 OBJETIVOS

1.3.1 Objetivo general

Implementar un algoritmo basado en aprendizaje profundo para la clasificación de palabras del lenguaje de señas colombiano aplicado en videos.

1.3.2 Objetivos específicos

- Construir un conjunto de datos de videos de las cinco primeras palabras del diccionario básico de la lengua señas colombiana.
- Diseñar una estrategia metodológica sustentada en el estado de arte para la clasificación de las cinco primeras palabras del diccionario básico de la lengua señas colombiana usando aprendizaje profundo.
- Desarrollar un algoritmo basado en aprendizaje profundo para la clasificación de las cinco primeras palabras del diccionario básico de la lengua señas colombiana grabadas en video.
- Medir el desempeño del algoritmo de aprendizaje profundo para evaluar el rendimiento de clasificación de las palabras, utilizando las métricas de exactitud, precisión y exhaustividad.

1.4 JUSTIFICACIÓN

El ministerio de educación en su proyecto de inclusión social orientado a la comunidad sordomuda ha realizado cursos interactivos y presenciales con instituciones privadas y públicas, fomentando el aprendizaje del lenguaje de señas en Colombia, estas formaciones tienen un porcentaje bastante alto de aceptación por parte de la sociedad, pero como todo proceso de formación presenta una curva de aprendizaje alta por la complejidad del lenguaje.

Usando asertivamente la tecnología la curva de aprendizaje de los cursos impartidos por las entidades privadas y públicas a los individuos que desean aprender el lenguaje de señas para comunicarse con la comunidad sordomuda disminuyen considerablemente, ya que el algoritmo se encargará de la clasificación de las palabras del lenguaje de señas realizando una clasificación a texto en español. Usando aprendizaje profundo se evidencia que el rendimiento de estos algoritmos en redes neuronales se comporta de manera eficiente en la identificación de objetos y movimientos en videos por su arquitectura de capas, simulando el funcionamiento biológico del cerebro compuesto por la interconexión entre neuronas (capas).

El impacto social que se quiere lograr con la ejecución de este proyecto es establecer una base computacional para el desarrollo de futuros trabajos, en los cuales se realicen clasificaciones del lenguaje de señas colombiano en video para oraciones y palabras compuestas, sin mencionar que estos futuros trabajos afectaran positivamente la calidad de vida de la comunidad sordomuda permitiendo optimizar su comunicación con la sociedad, sin que su contraparte sea experto en el lenguaje de señas.

1.5 DELIMITACIÓN

1.5.1 Limitaciones

Para alcanzar el objetivo planteado en la presente investigación se debe hacer uso de una Unidad de procesamiento gráfico (GPU) para el procesamiento de gráficos y videos. Se deben recolectar aproximadamente 100 videos por palabra para conformar el conjunto de datos. Se limitará al uso del lenguaje de señas colombiano (LSC) en la ciudad de Bogotá. Se debe contar con el consentimiento de las personas que serán grabadas para la construcción del conjunto de datos. Las palabras a analizar serán: anciano, bebé, hombre, homosexual y joven.

1.5.2 Alcances

Inicialmente se realizará un trabajo de campo y búsqueda en bases de datos públicas para obtener los videos de diferentes personas realizando las señas de únicamente las cinco palabras seleccionadas en el lenguaje de señas colombiano en la ciudad de Bogotá con el fin de crear el conjunto de datos. posteriormente se desarrollará la aplicación de un algoritmo de aprendizaje profundo a partir del conjunto de datos haciendo uso de Python donde se implementará métodos para las etapas de reprocesamiento, aplicación de una red neuronal, además se realizará el análisis del desempeño del algoritmo implementado. dicho experimento se desarrollará en el lenguaje de programación Python 3.7 o superior durante el periodo académico de primer semestre del 2020.

2. MARCO DE REFERENCIA

2.1 Marco Teórico

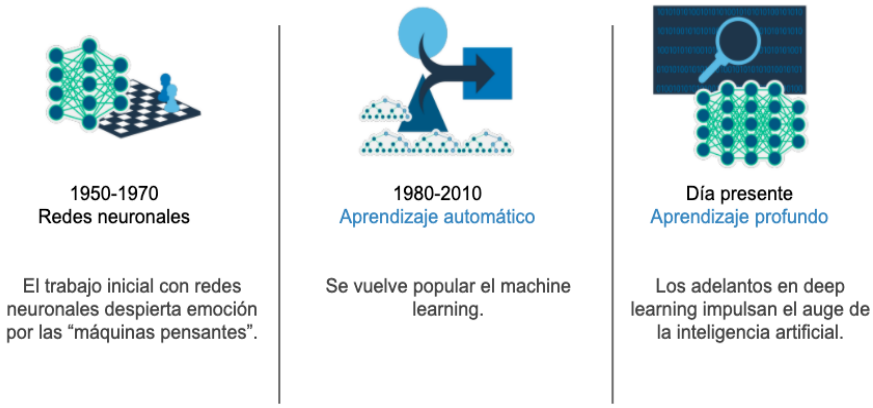
2.1.1 Inteligencia artificial

En los laboratorios del Dartmouth College en los estados unidos un grupo de científicos inició el proyecto de investigación “Inteligencia Artificial”, al inicio del proyecto el objetivo era que la inteligencia humana pudiera ser descrita de forma tan precisa que una máquina fuera capaz de simularla, como se evidencia en la ilustración No1 vemos el avance de la inteligencia artificial en el tiempo.

La Inteligencia artificial es el campo científico de la informática que se centra en la creación de programas y mecanismos que pueden mostrar comportamientos considerados inteligentes. En otras palabras, la IA es el concepto según el cual “las máquinas piensan como seres humanos”.[11] Normalmente un sistema de IA es capaz de analizar datos en grandes cantidades (big data), identificar patrones y tendencias y, por lo tanto, formular predicciones de forma automática, con rapidez y precisión.

En la línea de tiempo visualizada en la ilustración No 2 se pueden evidenciar las tres principales técnicas de IA las cuales son: redes neuronales, aprendizaje de máquina o automático y aprendizaje profundo, las cuales se entrará en detalle más adelante.

Ilustración 1 Técnicas de IA

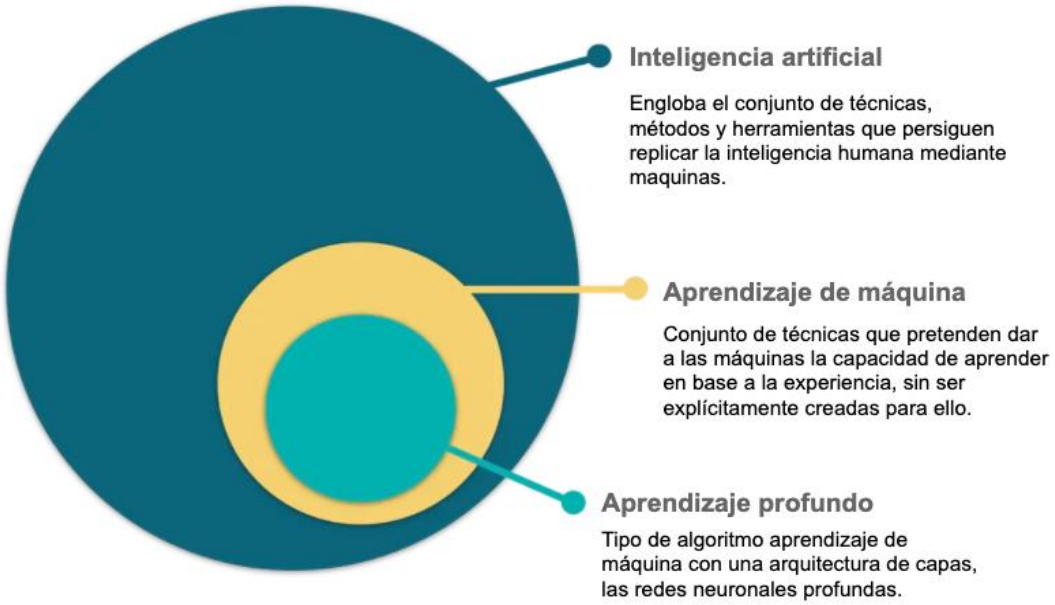


Fuente: SAS. 2019. Tomado de: Inteligencia Artificial. Inteligencia artificial Qué es y por qué es importante. Bogotá : s.n., 2019.

[11] **SalesForce. 2017.** SalesForce Blog. *SalesForce Blog.* [En línea] SalesForce.org, 22 de Junio de 2017. [Citado el: 21 de 10 de 2019.] https://www.salesforce.com/mx/blog/2017/6/Que-es-la-inteligencia-artificial.html.

A su vez la inteligencia artificial (IA) tiene relación directa con el aprendizaje de maquina (ML), y este a su vez con la técnica de aprendizaje profundo (DL). Para que el concepto quede más claro la ilustración No 3 indica la relación directa entre estas tres técnicas de la inteligencia artificial.

Ilustración 2 Relación Entre IA, ML, DL



Fuente: Spot. 2019. La diferencia entre Inteligencia Artificial, Machine Learning y Deep Learning. *La diferencia entre Inteligencia Artificial, Machine Learning y Deep Learning.* 2019

2.1.2 Aprendizaje de máquina

Es una rama de la inteligencia artificial basada en la idea de que los sistemas pueden aprender de datos, identificar patrones y tomar decisiones con mínima intervención humana. El aspecto iterativo de Aprendizaje de máquina es importante porque a medida que los modelos son expuestos a nuevos datos, éstos pueden adaptarse de forma independiente. Aprenden de cálculos previos para producir decisiones y resultados confiables y repetibles.

Existen tres subramas en los cuales se clasifican las técnicas de aprendizaje profundo:

Aprendizaje Supervisado: Se tiene un histórico de datos en el que se dispone de la variable a predecir (o campo objetivo o etiqueta) para entrenar el modelo.

Aprendizaje No Supervisado: Para el entrenamiento del modelo se dispone de un conjunto de datos de históricos que no están etiquetados y el algoritmo descifra la información y clasifica por sí solo.

Aprendizaje Por Refuerzo: Las dos cualidades principales del aprendizaje por refuerzo son prueba-error y la recompensa. Un algoritmo que aprende por refuerzo lo que hace es recibir una recompensa positiva si el resultado de la acción es positiva, y negativa o nula si es malo. El objetivo del aprendizaje se consigue mediante la maximización de la función de recompensa.

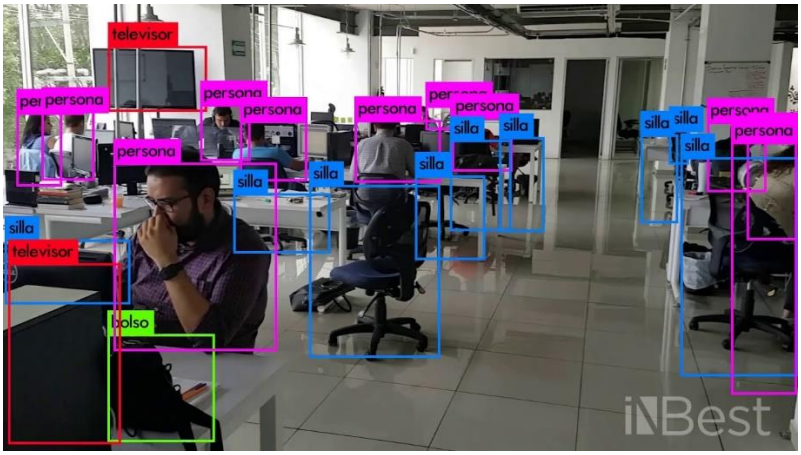
2.1.3 Aprendizaje profundo

Se trata de una clase de algoritmos de aprendizaje de máquina supervisados o no supervisados con una arquitectura en capas basados en topología de neuronas. En lugar de organizar datos para que se ejecuten a través de ecuaciones predefinidas, el aprendizaje profundo configura parámetros básicos acerca de los datos y entrena a la computadora para que aprenda por cuenta propia reconociendo patrones mediante el uso de muchas capas de procesamiento.[12]

El Aprendizaje profundo ha llamado mucho la atención por su potencial utilidad en distintos tipos de aplicaciones en el “mundo real” (pueden aplicarse con éxito a grandes volúmenes de datos para el descubrimiento y aplicación de conocimiento, así como a la realización de predicciones a partir de él), principalmente debido a que obtiene tasas de éxito elevadas con entrenamiento “no supervisado”, como visualizamos en la ilustración No 3 donde en una simple imagen el reconocimiento de imágenes del algoritmo puede encontrar más de 10 coincidencias diferentes.

[12] SAS. (21 de 10 de 2019). *Software y Soluciones De Analitica*. Obtenido de Software y Soluciones De Analitica: https://www.sas.com/es_co/insights/analytics/deep-learning.html

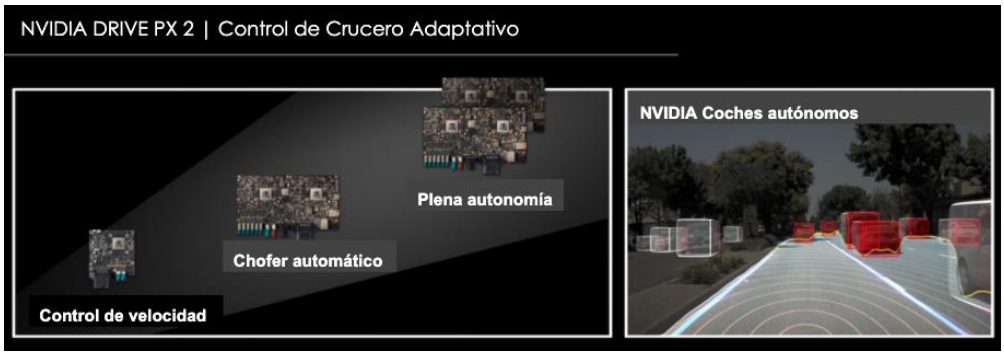
Ilustración 3 Salida del modelo de reconocimiento de imágenes en aprendizaje profundo



Fuente: Ataka. *Ataka.* [En línea] Aprendizaje de máquina y Aprendizaje profundo: cómo entender las claves del presente y futuro de la inteligenciaartificial. https://www.xataka.com/robotica-e-ia/machine- learning-y-deep-learning- como-entender-las-claves-del-presente-y-futuro-de-la-inteligencia-artificial.

Otra de las áreas donde el aprendizaje profundo tiene peso es el reconocimiento de voz. Google lleva años trabajando en este campo utilizando técnicas como Long Short-term Memory Recurrent Neural Networks (LSTM RNN) para mejorar sus servicios, incluidos los que utilizando sus teléfonos móviles Android al igual que sus asistentes virtuales para poder hacer consultas en lenguaje natural, tanto al buscador como a los asistentes presentes en los dispositivos móviles, inclusive se han realizado algoritmos donde la identificación de objetos en tiempo real es utilizado en la conducción no asistida como vemos en la ilustración No 4.

Ilustración 4 Salida del modelo de reconocimiento de objetos en aprendizaje profundo



Fuente: Ataka. *Ataka.* [En línea] Machine Learning y Aprendizaje profundo: cómo entender las claves del presente y futuro de la inteligencia artificial.

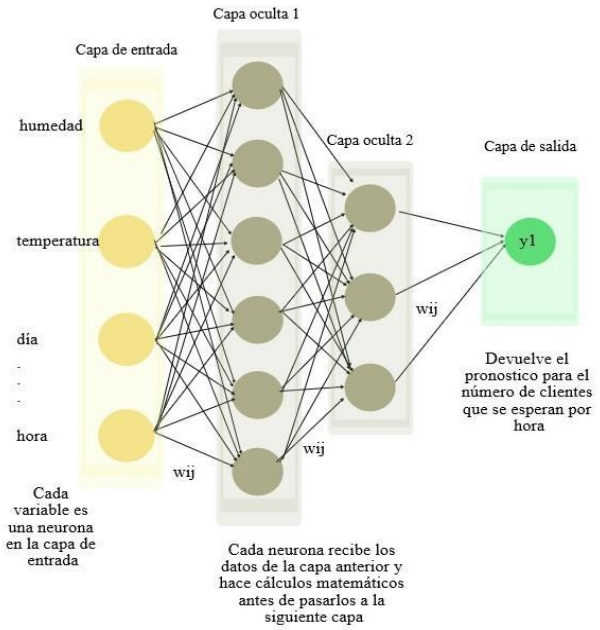
https://www.xataka.com/robotica-e-ia/machine-learning-y-deep-learning- como-entender-las-claves-del-presente-y-futuro-de-la-inteligencia-artificial.

2.1.4 Redes neuronales

Las redes neuronales intentan “emular” la forma en que las personas toman decisiones, de modo “similar” a cómo lo hacen las neuronas en el cerebro: como unidades especializadas en procesar determinada información de forma jerárquica e interconectadas entre sí. Se basan en una estructura básica, en el “perceptrón” y en el algoritmo de “Backpropagation”, que permite a la neurona aprender por sí misma y descubrir la información oculta en los datos de entrada con la que se entrena.

Cada capa (neurona) dentro de una red neuronal es una función matemática que recibe datos, los transforma y los transfiere a la siguiente capa de neuronas. En la ilustración no 4 se puede ver la arquitectura de una red neuronal que consta de una capa de entrada, dos capas ocultas, y una capa de salida, Las conexiones entre las neuronas se dan a través de los pesos que tienen asociados (Wij) y representan la influencia de la neurona de salida en la neurona de entrada en la conexión.

Ilustración 5 Arquitectura De Una Red Neuronal

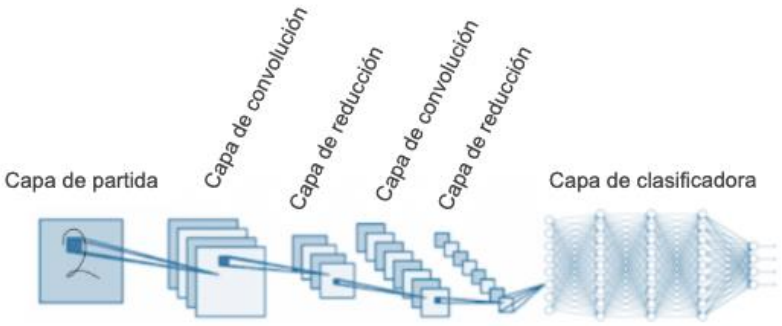


Fuente: GAGO, ENCARNACIÓN. 2019**.** Todamada de: Inteligencia Artificial, Machine Learning & Aprendizaje profundo. *Inteligencia Artificial, Machine Learning & Deep Learning.* Madrid : s.n., 2019.

2.1.5 Redes neuronales convolucionales

Es un algoritmo de aprendizaje profundo que puede tomar una imagen de entrada, asignar importancia (pesos) a varios aspectos u objetos en la imagen y poder diferenciar uno de otro, como se puede ver en la ilustración No 6 donde vemos una secuencia de una red neuronal convolucional que permite clasificar los dígitos escritos a mano. El preprocesamiento requerido en un ConvNet es mucho menor en comparación con otros algoritmos de clasificación. Mientras que en los métodos primitivos los filtros están diseñados a mano, con suficiente entrenamiento, los ConvNets tienen la capacidad de aprender estos filtros / características.

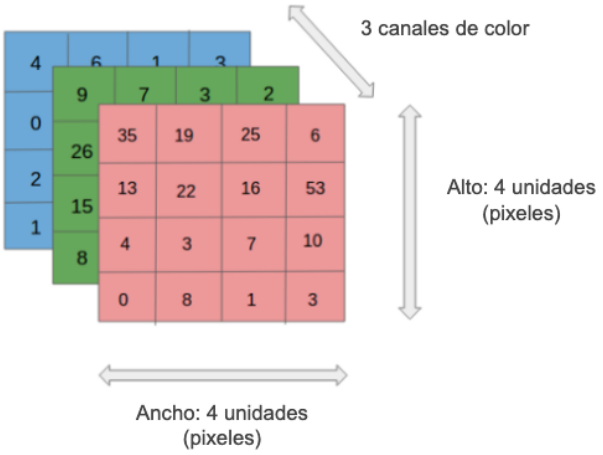
Ilustración 6 Secuencia CNN para clasificar los dígitos escritos a mano



Fuente: Saha, Sumit. 2018. Tomada de: A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way. *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way.* 2018.

Las redes neuronales convolucionales aprovechan el hecho de que la entrada consiste en imágenes y limitan la arquitectura de una forma más adaptable. En particular, a diferencia de una red neuronal normal, las capas de un ConvNet tienen neuronas dispuestas en 3 dimensiones: ancho, alto, profundidad. (Tenga en cuenta que la palabra *profundidad* aquí se refiere a la tercera dimensión de un volumen de activación, no a la profundidad de una red neuronal completa, que puede referirse al número total de capas en una red.) ver ilustración No 7 para un ejemplo más claro.

Ilustración 7 4x4x3 Imagen RGB



Fuente: Saha, Sumit. 2018. Tomada de: A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way. *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way.* 2018.

2.1.6 Residual Neural Network (ResNet)

Una red neuronal residual, es una red neuronal artificial que se basa en las construcciones de las células piramidales en la corteza cerebral del ser humano. Los modelos típicos de ResNet se implementan con saltos de doble o triple capa como se muestra en la ilustración 8 que contienen normalización por lotes en el medio.[13] ResNet proporciona un marco de aprendizaje para facilitar el entrenamiento de redes que son sustancialmente más profundas. Los experimentos mostraron que ResNet mejora enormemente la eficiencia del entrenamiento ya que los gradientes pueden propagar muchas capas a través de la conexión de atajo.

El objetivo de la técnica utilizada por este tipo de redes consiste en rediseñar las capas utilizando funciones residuales de aprendizaje con referencia a las capas de entrada. Este nuevo diseño facilita la optimización ganando precisión incrementando la capacidad de las capas. Se proporciona una serie de pruebas empíricas exhaustivas que muestran que estas redes residuales son más fáciles de optimizar, y pueden ganar precisión a partir de una profundidad considerablemente mayor. En el conjunto de datos de ImageNet

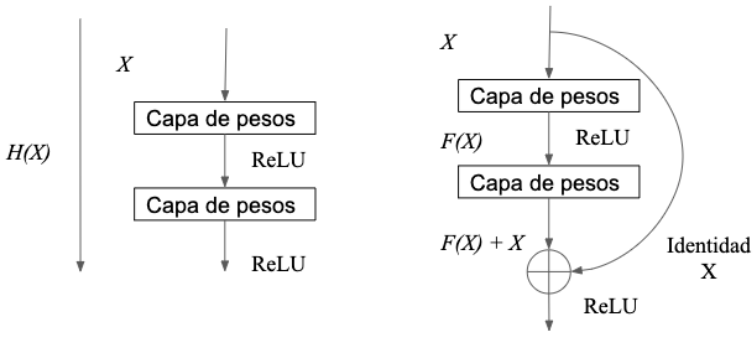
[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. 2016. *Deep Residual Learning for Image Recognition.* EEUU : s.n., 2016.

se evalúan redes residuales más profundas que las redes VGG, pero aun así con menor complejidad.

Un conjunto de estas redes residuales alcanza un error del 3,57% en el conjunto de pruebas de ImageNet. Este resultado ganó el primer lugar en la tarea de clasificación del ILSVRC 2015.

La profundidad de las representaciones es de gran importancia para tareas de reconocimiento visual. ResNet plantea una mejora relativa del 28% en el conjunto de datos de detección de objetos COCO. Las redes residuales profundas han ganado los primeros puestos en las tareas de detección ImageNet, localización ImageNet, detección de COCO y segmentación de COCO.[14]

Ilustración 8 Aprendizaje residual: un bloque de construcción



Fuente: Burgal, Jesús Utrera. 2018**.** Tomado de: Deep Learning básico con Keras (Parte 4): ResNet. *Deep Learning básico con Keras (Parte 4): ResNet.* 2018.

2.1.1 Keras

Es un API de aprendizaje profundo de alto nivel escrita en Python que hace muy simple y rápido el funcionamiento y experimentación de las redes neuronales. Es capaz de funcionar sobre TensorFlow, Theano o el kit de herramientas cognitivas de Microsoft conocido anteriormente como CNTK. TensorFlow viene con su propia implementación de esta API, llamada tf.keras, que proporciona soporte para algunas características avanzadas de TensorFlow[15]

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. 2016. *Deep Residual Learning for Image Recognition.* EEUU : s.n., 2016.

[15] Geron, Aurelien. 2017. *Hands-On Machine Learning with Scikit-Learn and TensorFlow.* 2017.

2.1.2 Matriz de confusión

Es una herramienta que permite identificar fácilmente el rendimiento de un modelo supervisado de aprendizaje de máquina. Con esta métrica es fácil detectar dónde el sistema está confundiendo las diferentes clases o resultados de clasificación. Como se visualiza en la ilustración 9 cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real.

Ilustración 9 Ejemplo matriz de confusión

VP es la cantidad de positivos que fueron clasificados correctamente como positivos por el modelo (Verdadero positivo)

VN es la cantidad de negativos que fueron clasificados correctamente como negativos por el modelo (Verdadero negativo)

FN es la cantidad de positivos que fueron clasificados incorrectamente como negativos. Error tipo 2 (Falsos Negativos)

FP es la cantidad de negativos que fueron clasificados incorrectamente como positivos. Error tipo 1 (Falsos positivos)[16]

2.1.3 Medidas de rendimiento

Precisión (P): definida como la cantidad de registros verdaderos positivos (VP) dividida entre la suma de verdaderos positivos (VP) y falsos positivos (FP) obtenidos en la ejecución del algoritmo.

$$P = \frac{VP}{VP + FP}$$

[16] Barrios, Juan. 2019. Juan Barrios. *Juan Barrios.* [En línea] Big Data, 2019. https://www.juanbarrios.com/matriz-de-confusion-y-sus-metricas/.

Exhaustividad (R, Recall) Una medida de integridad definida como la proporción del número de registros verdadero positivos (VP) divididos por la suma de los verdaderos positivos (VP) y registros clasificados como falsos negativos (FN).

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

F1-score: Es una función de Precisión y Exhaustividad que permite buscar un equilibrio entre Precisión y Exhaustividad.

$$F1 = 2\,x\,\frac{PRE\ x\ REC}{PRE + REC}$$

Exactitud (ACC, Accuracy) Es el porcentaje de casos en los que el modelo ha acertado, es el número de predicciones correctas sobre el número total de predicciones

$$ACC = \frac{TP + TN}{FP + FN + TP + TN}$$

Macro avg Se puede calcular como:

$$B_{macro} = \frac{1}{q}\sum_{\lambda=1}^{q} B(tp_\lambda, fp_\lambda, tp_\lambda, fn_\lambda)$$

Donde:

- $B: medida\ de\ evaluación\ binaria$
- $tp: verdaderos\ negativos, tn: falsos\ positivos$
- $fp: falsos\ positivo, fn: falsos\ negativos$
- $\lambda: una\ etiqueta\ del\ conjunto\ de\ etiquetas\ L = \{\lambda_j : j = 1 \dots q\}$

Se puede considerar una medida de evaluación binaria $B\{tp_\lambda, fp_\lambda, tp_\lambda, fn_\lambda\}$ que se calcula basado en el número de verdaderos positivos (tp), verdaderos negativos (tn), falsos positivos (fp) y falsos negativos (fn). $tp_\lambda, fp_\lambda, tp_\lambda, fn_\lambda$ son el número de verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos después de la evaluación binaria por una etiqueta λ.

2.1.4 Validación cruzada

La validación cruzada es una técnica para la validación de modelos que permite evaluar la calidad del análisis estadístico y sus resultados. El objetivo es hacer que el modelo se generalice hacia un conjunto de pruebas independientes. Esta técnica es utilizada para hacer una predicción a partir de un modelo de aprendizaje automático, ayuda a estimar cómo un modelo predictivo se desempeñará con precisión en la práctica cuando se despliegue como una aplicación ML. Durante la validación cruzada, un modelo suele entrenarse con un conjunto de datos de un tipo conocido. Por el contrario, se prueba utilizando un conjunto de datos de tipo desconocido.[17]

[17] Asch, Vincent Van. 2013. *Macro- and micro-averaged evaluation.* 2013.

2.2 Marco Conceptual

2.2.1 Persona sorda

Es aquélla que, de acuerdo con valoraciones médicas, presenta una pérdida auditiva mayor de noventa (90) decibeles y cuya capacidad auditiva funcional no le permite adquirir y utilizar la lengua oral en forma adecuada, como medio eficaz de comunicación.[18]

2.2.2 Lenguaje

Se puede definir la palabra lenguaje como capacidad del ser humano para expresar sus sentimientos, pensamientos, sus ideas por medio de signos los cuales pueden ser sonoros, escritos incluso señales realizadas por el emisor. Desde un punto de vista formal se define como un conjunto de frases, que generalmente es infinito y se forma con combinaciones de elementos tomados de un conjunto (usualmente infinito) llamado alfabeto, respetando un conjunto de reglas de formación (sintácticas o gramaticales) y de sentido (semánticas).[19] Se pueden distinguir dos clases importantes del lenguaje: lenguaje natural y lenguaje formal.

2.2.3 Lenguaje natural (LN)

Es la lengua o idioma creados por un grupo de individuos con el fin de comunicarse, es aquel que ha evolucionado con el tiempo para fines de comunicación humana como el español o el alemán[20], a diferencia del lenguaje formal que están definidos por reglas y preestablecidas como las matemáticas, lenguajes de programación entre otros. Un LN se define a partir de unas reglas gramaticales, y a su vez el lenguaje se enriquece progresivamente modificando sus mismas reglas, como se puede visualizar en la ilustración 10.

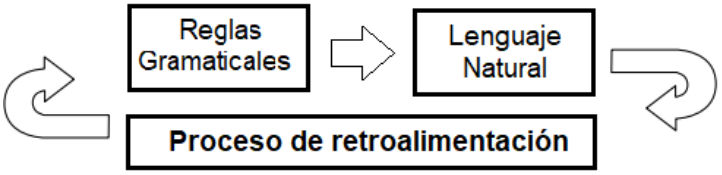
---

[18] 1997. Instituto Colombiano. *Instituto Colombiano.* [En línea] 26 de 09 de 1997. [Citado el: 08 de 02 de 2020.] https://www.icbf.gov.co/cargues/avance/docs/decreto_2369_1997.htm.

[19] *Procesamiento de lenguaje natural.* Mg. Augusto Cortez Vásquez, Mg. Hugo Vega Huerta, Lic. Jaime Pariona Quispe. 2009. 2, San Marcos : Revista de Ingeniería de Sistemas e Informática, 2009, Vol. 6. (Procesamiento del lenguaje natural, 2009)

[20] [BROOKSHEAR 1993] BROOKSHEAR J. Glean. Teoría de la computación Addisson Wesley iberoamericana Wilmington Delaware 1993

Ilustración 10 Estructura lenguaje natural



Fuente: *Procesamiento del lenguaje natural.* Vásquez Cortez, Augusto, Vega Huerta, Hugo y Paronia Quispe , Jaime. 2009. San Marcos : Revista de ingenieria de sistemas e informatica, 2009.

2.2.4 Lengua Manual Colombiano

Según decreto 2369 de 1997 articulo No 3 define a la lengua manual colombiana como: “idioma propio de la comunidad sorda del país, constituye la lengua natural de la misma, estructurada como un sistema convencional y arbitrario de señas viso gestuales, basado en el uso de las manos, los ojos, el rostro, la boca y el cuerpo.”[21]

2.2.5 Lenguaje De Señas

En 1960 William C. Stokoe en su publicación “Sign Language Structure (Estructura de la lengua de signos)” propone que las señas pueden ser analizadas como compuestos simultáneos de tres elementos sin significado: una forma de la mano, una actividad de la mano y un lugar ocupado por la mano. Eso le permitió argumentar que la lengua de señas usada por sus estudiantes era un código doblemente articulado, es decir, una lengua natural.[22] Al ser denominada una lengua natural permite su estudio en todos los niveles lingüísticos fonológico, morfológico, semántico y pragmático, es necesario aclarar que este lenguaje de señas no es internacional, es decir hay diferencias en su interpretación, entre regiones y sus países.

2.2.6 Lenguaje de señas colombianas (LSC)

Según el INSOR en su “diccionario básico de la lengua de señas colombiana”, los orígenes del lenguaje de señas colombiano se remontan a 1920 en un internado católico bogotano. En 1957 aparece la primera asociación de sordos en Bogotá y un año después otra en Cali. Parece que estos sistemas de señas recibieron influencia de la lengua de señas

[21] 1997. Instituto Colombiano. *Instituto Colombiano.* [En línea] 26 de 09 de 1997. [Citado el: 08 de 02 de 2020.] https://www.icbf.gov.co/cargues/avance/docs/decreto_2369_1997.htm.

[22] (Jr.), William C. Stokoe. 1960. *Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf.* s.l. : University of Buffalo, 1960.

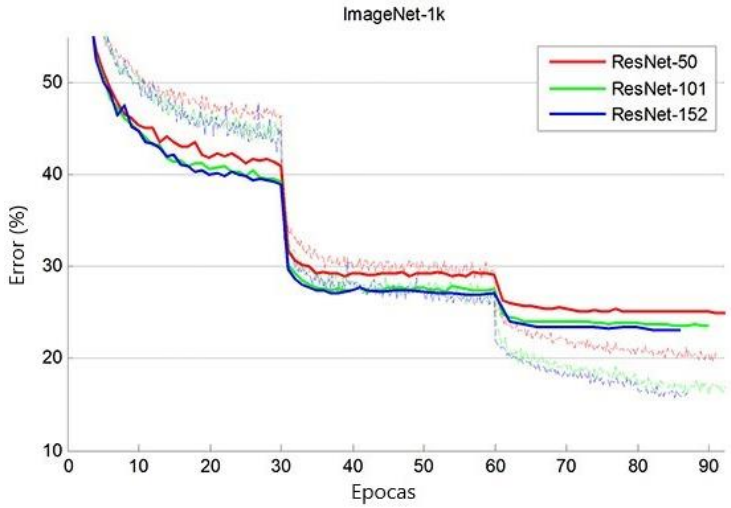
española, a través de inmigrantes o de sordos colombianos educados en España, en los años 50. El español hablado y escrito también influyó, en esa misma época, en el auge de la educación oficial oralista.[23]

El gobierno nacional decreto según la ley 324 de 1996 el reconocimiento del lenguaje de señas en Colombia en su artículo No 2 “El Estado Colombiano reconoce la Lengua Manual Colombiano, como idioma propio de la Comunidad Sorda del País.”[24]

2.2.7 ResNet-50

Es una red neuronal residual, el "50" se refiere al número de capas que tiene. Es una subclase de redes neuronales convolucionales, siendo ResNet la más utilizada para la clasificación de imágenes.[19] , los resultados obtenidos se comparan entre redes de 50, 101 y 152 capas de profundidad, como se visualiza en la ilustración 11 se demuestra que con más número de capas se obtienen mejores resultados, no obstante la complejidad añadida en la red de 152 capas no mejora sustancialmente las tasas de error que se obtiene con la red de 101 capas, por complejidad computacional y en su implementación se decidió para este proyecto utilizar el modelo de 50 capas.

Ilustración 11 Curvas en ImageNet (líneas sólidas: error de val de 1 corte; líneas discontinuas: error de entrenamiento)



Fuente: He, Kaiming. 2016. github. *github.* [En línea] github.com, 29 de 06 de 2016. [Citado el: 20 de 03 de 2020.] Tomado de: https://github.com/KaimingHe/deep-residual-networks#models.

[23] Paulina Ramírez. Métodos de enseñanza del español a los niños sordos: actitudes de los profesionales. Santafé de Bogotá: Universidad Iberoamericana. (Tesis de maestría inédita).
[24] Colombia, El Congreso de. 1996**.** LEY 324 DE 1996 . Bogotá : Congreso, 1996

2.2.8 ImageNet

ImageNet es un conjunto de datos de imágenes organizado de acuerdo con la jerarquía de WordNet. Cada concepto significativo en WordNet, posiblemente descrito por varias palabras o frases de palabras, se denomina "conjunto de sinónimos" o "synset". Hay más de 100,000 synsets en WordNet, la mayoría de ellos son sustantivos (80,000+). En ImageNet, nuestro objetivo es proporcionar en promedio 1000 imágenes para ilustrar cada synset. Las imágenes de cada concepto son de calidad controlada y anotadas por humanos. En su finalización, esperamos que ImageNet ofrezca decenas de millones de imágenes ordenadas limpiamente para la mayoría de los conceptos en la jerarquía de WordNet.[25]

2.2.9 Fully Connected layers (FC LAYER)

Las capas completamente conectadas conectan cada entrada de una capa a cada unidad de activación de la siguiente capa. En principio, es lo mismo que la red neuronal perceptrónica multicapa (MLP) tradicional. En los modelos de aprendizaje automático más populares, las últimas capas son capas completamente conectadas que compilan los datos extraídos por las capas anteriores para formar la salida final. Las FC layer juegan un papel importante en el logro de una alta precisión en los dominios de destino mediante el ajuste fino del modelo pre entrenado[26]

2.2.10 Descenso de gradiente estocástico SGD

Es una variación del algoritmo de descenso del gradiente, consiste en que el modelo haga predicciones sobre cada conjunto de datos de entrenamiento y se calcula el error para actualizar el modelo procurando reducir el error en la predicción, su principal problemática es que su aplicación tiene un elevado costo computacional ya consume mucho más tiempo entrenar un modelo con conjuntos de datos grandes.[27]

2.2.11 Pickle

Es un protocolo que permite la serialización de objetos complejos, este protocolo fue desarrollado por Python y permite representar un objeto Python como una cadena de bytes. Al ser un protocolo de Python no puede ser usado en otros lenguajes de programación. Pickle es inseguro ya que deserializa

---

[25] Image-net.org. 2020. *Imagenet*. [online] Available at: <http://www.image-net.org/about-overview> [Accessed 30 March 2020].8

[26] ZHANG, Chen-Lin, et al. In defense of fully connected layers in visual representation transfer. En *Pacific Rim Conference on Multimedia*. Springer, Cham, 2017. p. 807-817..

[27] Roma, J. C., Rué, A. B., & Bagén, T. L. (2019a). Deep learning: principios y fundamentos. Barcelona, ESPAÑA: UOC.

datos que fueron serializados con pickle desde fuentes inseguras puede ejecutar código arbitrario, si los datos estuvieron interceptados por un atacante.[28]

2.2.1 Muestreo

Antes de definir el muestro que se utilizó en el desarrollo de este proyecto, se define la muestra como un subconjunto o parte del universo o población, hay procedimientos para obtener la cantidad de los componentes de la muestra como fórmulas y lógica, entonces que es el muestreo, Es el método utilizado para seleccionar a los componentes de la muestra del total de la población. "Consiste en un conjunto de reglas, procedimientos y criterios mediante los cuales se selecciona un conjunto de elementos de una población que representan lo que sucede en toda esa población"[29]

Se dividen en dos grupos, tipos de muestreo probabilístico y el no probabilístico, el muestreo probabilístico es el método más recomendable si se está haciendo una investigación cuantitativa porque todos los componentes de la población tienen la misma posibilidad de ser seleccionados para la muestra. "Cada uno de los elementos de la población tengan la misma probabilidad de ser seleccionados"[30], siempre y cuando cumpla estas dos condiciones, todos los elementos de la población tienen una probabilidad mayor a cero de ser seleccionados en la muestra, la probabilidad de inclusión de cada elemento en la muestra se conoce de forma precisa. El cumplimiento de ambos criterios es el que hace posible obtener resultados no sesgados cuando se estudia la muestra y determinar el grado de incertidumbre que añade el proceso de muestreo

Para el desarrollo del proyecto se escogio el muestreo aleatorio simple que consiste en seleccionar un grupo de n unidades de muestreo de forma

---

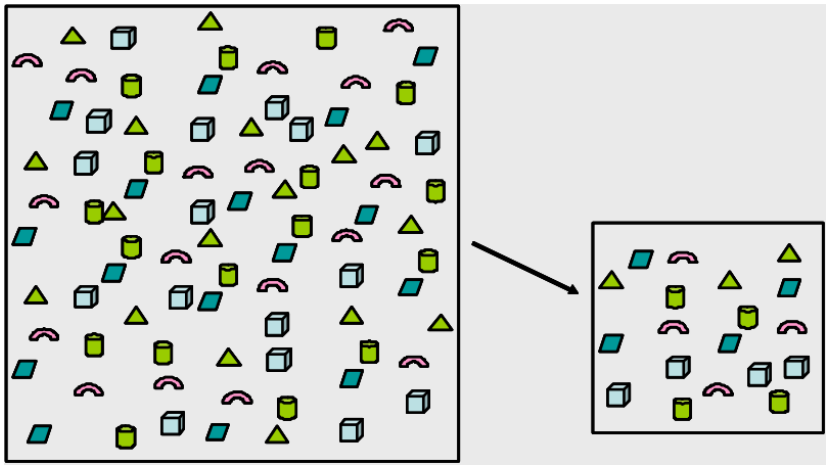
[28] pickle. Python object serialization. *Python object serialization.* [En línea] [Citado el: 13 de 04 de 03.] https://docs.python.org/3/library/pickle.html.

[29] Cristina, Mata maria. 1997. *Cómo elaborar muestras para los sondeos de audiencias. Cuadernos de investigación No 5.* Quito : MACASSI, 1997.

[30] PINEDA, Beatriz, DE ALVARADO, Beatriz y DE CANALES, Francisca. 1994. *Metodología de la investigación, manual para el desarrollo de person al de salud.* Washington : Organización Panamericana de la Salud, 1994.

que cada muestra de tamaño n tenga la misma oportunidad de ser seleccionada[31] como se evidencia en la ilustración 12.

Ilustración 12 Representación gráfica del muestreo aleatorio simple



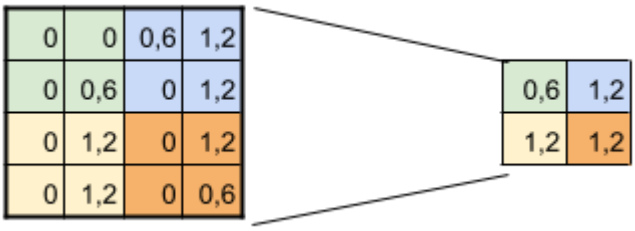
Fuente: Casal, Jordi y Mateu, Enric. 2003. *TIPOS DE MUESTREO.* Barcelona : s.n., 2003. 08193.

¿Porque es necesario realizar muestro como etapa en una red neuronal convolucional? Normalmente con una imagen en blanco y negro de 28 x 28 px tenemos una primera capa de entrada de la red neuronal de 784 neuronas y luego de la primera convolución obtenemos una capa oculta de 25.088 neuronas que realmente son nuestros 32 mapas de características de 28*28 ((28*28)*32) Si hiciéramos una nueva convolución a partir de esta capa, el número de neuronas de la próxima capa requeriría un poder computacional importante. Por ello y para reducir el tamaño de la próxima capa de neuronas hacemos un muestreo preservando las características más importantes que detectó cada filtro. Hay diversos tipos de muestreo (subsampling) , El más usado es: Max-Pooling.

Esto quiere decir que se recorrerá cada una de las imágenes de características obtenidas de n*n px de izquierda-derecha, arriba-abajo, pero en vez de tomar de a 1 pixel, se tomara de 2*2 (2 de alto por 2 de ancho = 4 pixeles) e iremos preservando el valor más alto de entre esos 4 pixeles (por eso lo de Max), como se visualiza en la ilustración 13.

[31] Richard L. Scheaffer, William Mendenhall, Lyman Ott. 2007. *Elementary survey sampling.* Magallanes : Thomson, 2007.

Ilustración 13 Subsampling aplicando max pooling de 2X2 reduciendo la salida a la mitad



Fuente: Barrios, Juan. Tomado de: Big data en salud. *Big data en salud.* [En línea] Juan Barrios. [Citado el: 21 de 03 de 2020.]

2.2.2 Hiperparametro

Un parámetro para un modelo de aprendizaje profundo se considera como una variable de configuración que interna al modelo y cuyo valor puede ser estimado a partir del estudio de los datos. En cambio los hiperparametros son variables de configuración externas al modelo en sí mismo y su valor en general no puede ser estimado a partir de los datos, y son especificados por el programador para ajustar los algoritmos de aprendizaje partiendo de su experiencia e intuición para encontrar su valor óptimo, estos valores son configurados antes de iniciar el proceso de entrenamiento y se pueden modificar para que los modelos se entrenen y den mejores resultados en las métricas y tiempo de ejecución. [32]

2.2.3 Epoch

Es un hiperparámetro que indica el número de veces que todos los conjuntos de datos de entrenamiento han pasado por la red neuronal en su entrenamiento. Una buena práctica para establecer este parámetro es incrementarlo hasta que la métrica de exactitud con los datos de prueba empiece a decrecer, incluso cuando la exactitud de los datos de entrenamiento sigue incrementando. [33]

[32] Jordi, TORRES. 2018. *DEEP LEARNING Introducción práctica con Keras.* 2018.
[33] Jordi, TORRES. 2018. *DEEP LEARNING Introducción práctica con Keras.* 2018.

2.2.4 ReLU (Rectifer Liner Unit)

Es una función de activación utilizada en las redes neuronales convolucionales que consiste en la función $f(x) = \max(0, x)$, Son utilizados después de cada convolución. Son una operación que reemplaza los valores negativos por cero y su propósito es agregar no linealidad al modelo, eliminando la relación proporcional entre la entrada y salida.

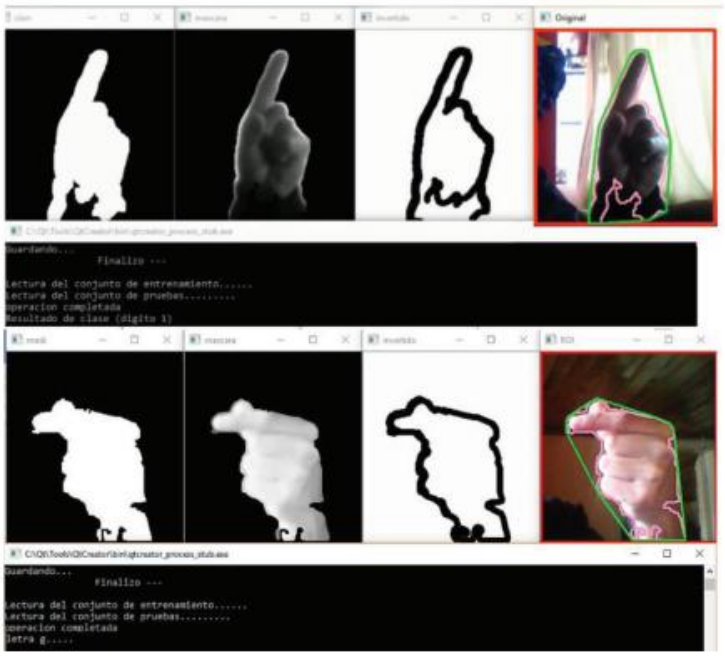
2.3 Estado del arte

2.3.1 Interpretación de lenguaje de señas ecuatoriano empleando visión por computador

Se encuentran investigaciones y aplicaciones donde se implementa aprendizaje de máquina para la identificación de algunas partes del lenguaje de señas, no solo colombiano sino que también de otros países, como por ejemplo, una trabajo desarrollado por estudiantes de la Universidad de las Fuerzas Armadas ESPE, Sangolquí – Ecuador que consiste en la aplicación de visión artificial para la interpretación del alfabeto y números en lenguaje de señas ecuatorianas a partir de imágenes en tiempo real, en el desarrollo de este trabajo se implementa también de un algoritmo de redes neuronales artificiales para detectar los datos más importantes de la imagen y con esto mediante la utilización de clasificadores reconocer cada uno de los patrones que forman las letras del abecedario y números en LSEC como se puede visualizar en la ilustración No 14 el sistema ha sido probado por diferentes usuarios, identificando satisfactoriamente 9 números y 25 letras. Son utilizadas dos redes neuronales para el aprendizaje de cada letra del abecedario y de cada número únicamente entre el uno y el nueve, y las letras del alfabeto que no necesiten de movimiento para su reconocimiento.[34]

[34]PICHUCHO, Javier P., et al. Interpretación de lenguaje de señas ecuatoriano empleando visión por computador (PICHUCHO, 2019)or. *Revista Ibérica de Sistemas e Tecnologias de Informação*, 2019, no E17, p. 960-971.

Ilustración 14 Reconocimiento del lenguaje de señas Número uno y letra g



Fuente: PICHUCHO, Javier P., et al. 2019. *Interpretación de lenguaje de señas ecuatoriano empleando visión por computador.* Quito : Revista Ibérica de Sistemas e Tecnologias de Informação, 2019.

2.3.2 Traducción automática del lenguaje dactilológico de sordos y sordomudos mediante sistemas adaptativos

Consiste en el desarrollo de un sistema que integra hardware y software, para el reconocimiento automático del lenguaje dactilológico de señas utilizado por personas con discapacidad auditiva. El hardware está compuesto por un sistema inalámbrico adherido a un guante, el cual posee un conjunto de sensores que capturan una serie de señales generadas por los movimientos gestuales de la mano, y el software es un modelo por adaptación basado en los principios de la computación neuronal, el cual permite su reconocimiento en términos de un lenguaje dactilológico y lograron un 90% de asertividad en el reconocimiento dado que un movimiento o gesto no se realiza siempre de la misma forma entre una persona y otra, proponen algoritmos de optimización que mejoren el proceso de

aprendizaje.[35] Este desarrollo fue elaborado por estudiantes de la escuela de Ingeniería de Antioquia (EIA)

2.3.3 Fundación HETAH

La fundación Hetah, Entidad sin ánimo de lucro que busca aportar generosamente conocimientos e iniciativas, para identificar, resolver y superar los problemas de la humanidad mediante investigación desarrollo y aplicación de tecnologías que trascienden las diferencias. Esta fundación tiene actualmente implementado un traductor en internet de español a lenguaje de señas colombiano. El traductor de LSC pretende facilitar la comunicación entre una persona oyente y un sordo, este sistema hace un análisis gramatical y utiliza unos componentes de inteligencia artificial para encontrar una secuencia de imágenes en el lenguaje de señas que corresponda a las palabras que previamente han sido digitadas.[36]

2.3.4 Clasificación automática de las vocales en el lenguaje de señas colombiano

Investigación desarrollada por estudiantes del Instituto Tecnológico Metropolitano – ITM, Medellín Colombia., donde se experimentó con aprendizaje de máquina para el reconocimiento de las vocales del lenguaje colombiano de señas, a partir de imágenes. en este proyecto adquirieron 151 imágenes por cada vocal o clase. A partir de cada imagen se separó el objeto de interés del resto de la escena usando información de color; luego, se extrajeron características para describir el gesto correspondiente a cada vocal. Posteriormente, se seleccionan cuatro conjuntos de características, se prueban múltiples clasificadores para cada conjunto por medio de validación cruzada, obteniendo un desempeño superior al 90%, concluyendo que la tesis propuesta permite una adecuada separación de las clases o vocales.[37]

---

[35] BETANCUR, D.B., GÓMEZ, M.V. and PALACIO, A.P. 2013. *Traducción Automática Del Lenguaje Dactilológico De Sordos y Sordomudos Mediante Sistemas adaptativos.* s.l. : ProQuest Central, 2013. ISSN 19099762.

[36] Ingeniero Colombiano Crea Traductor De Lenguaje De Señas Para Sordos. 2013. Madrid : ProQuest Central, 2013.

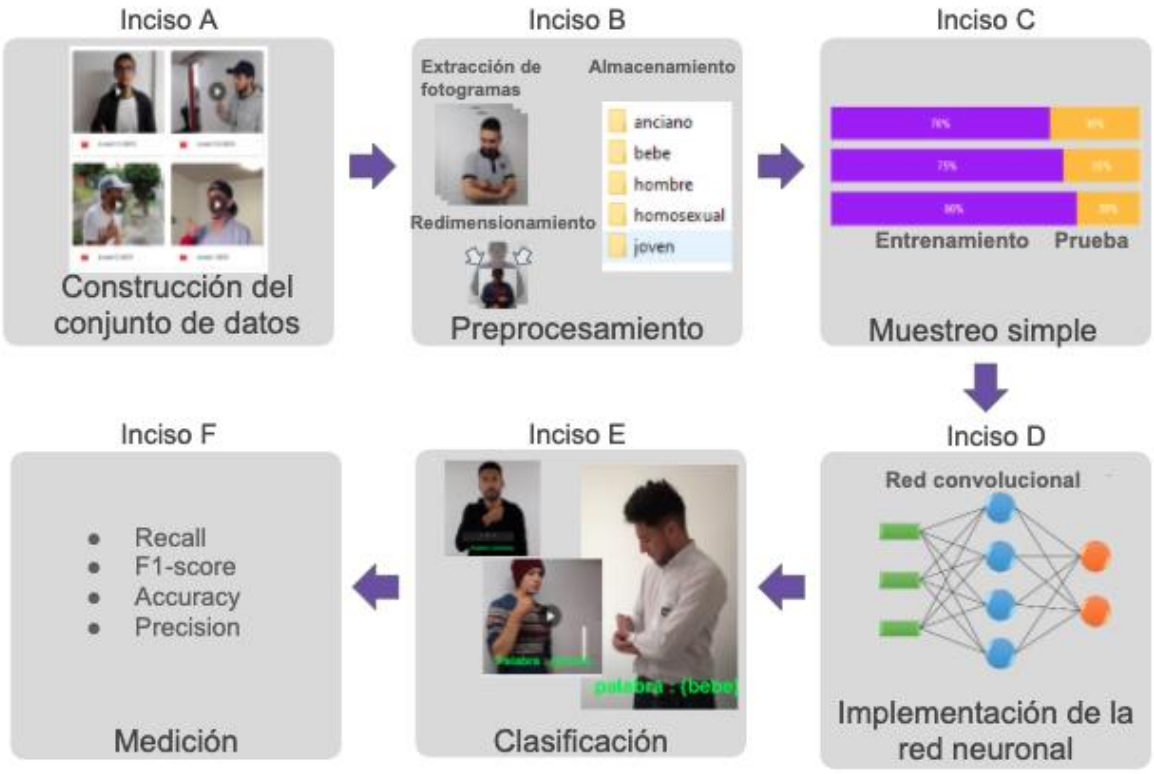
[37] BOTINA-MONSALVE, D.J., DOMÍNGUEZ-VÁSQUEZ, M.A. MADRIGAL-GONZÁLEZ y CASTRO-OSPINA. 2018. *Clasificación automática de las vocales en el lenguaje de señas.* 2018.

3. METODOLOGÍA

3.1 Fases del proyecto

Para la clasificación de las palabras en lenguaje de señas colombiano en video se utilizó la siguiente metodología estructurada en las siguientes fases como se evidencia en la ilustración No 15.

Ilustración 15 Metodología Aprendizaje profundo



Fuente: Los autores

Construcción del conjunto de datos.

En esta fase del proyecto se realizó la creación de un conjunto de videos grabados en formato .MOV, en los cuales distintas personas realizaron las palabras abuelo, bebe, homosexual, joven y hombre del lenguaje de señas colombiano como se muestra en el inciso A de la ilustración 15, la grabación de estos vídeos se realizó en diferentes posiciones, ambientes naturales, y ángulos de enfoque con el fin de simular el experimento lo más real posible.

Pre procesamiento.

En esta fase se realiza el ajuste de los videos grabados en la fase anterior como se muestra en el inciso B de la ilustración 15, convirtiendo cada uno de los videos en una secuencia de fotogramas en formato .jpg. Cada uno de los fotogramas se convierten del sistema de colores BGR a RGB y se redimensionan a un tamaño de 224 x 224px. Por último, las imágenes son almacenadas en una carpeta por cada palabra.

Muestreo

En esta fase del proyecto se utilizó el método de muestreo aleatorio simple como se muestra en el inciso C de la ilustración No 15, Se trabaja con tres tipos de muestreo en particiones de 70% entrenamiento y 30% pruebas, 75% entrenamiento y 25% pruebas y 80% entrenamiento y 20% pruebas.

Implementación de la red neuronal

En esta fase se implementó la red neuronal convolucional como se muestra en el inciso D de la ilustración No 15, la cual generó un modelo que permite la clasificación de las cinco palabras en lenguaje de señas colombiano en video.

Clasificación

En esta fase se analizó un video de prueba como se muestra en el inciso E de la ilustración No 15, en el cual se muestra a una persona gesticulando una de las cinco palabras en lenguaje de señas; al video se le aplica el modelo generado en la fase de extracción de características y como salida el algoritmo genera un nuevo video con una etiqueta la cual clasifica la seña traducida al español.

Medición

Como se muestra en el inciso F de la ilustración No 15, en esta fase se calculan una serie de métricas como resultado de la implementación de la red neuronal convolucional generando porcentajes para las medidas de Precisión (P), Exhaustividad (R, *Recall*), F1-score y exactitud (ACC, *Accuracy*).

4. DISEÑO METODOLÓGICO

Construcción del conjunto de datos

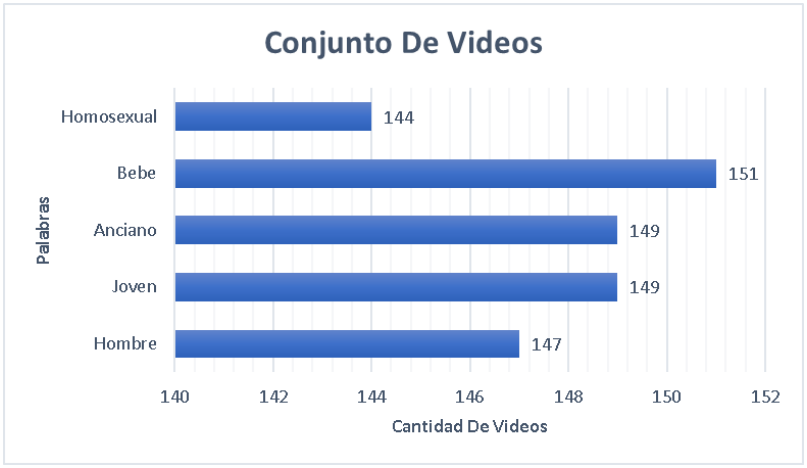
En esta etapa se graban a 51 estudiantes de la Universidad Católica de Colombia realizando la seña cada una de las cinco palabras que se escogieron para el desarrollo del experimento. Si contrastamos las primeras once palabras del diccionario básico de la lengua de señas colombiana de la categoría hombre las cuales son (adulto, algunos, anciano, bebe, cada-uno, el-ella , ellas-ellos, ellos-dos/ustedes-dos, hombre, homosexual, joven), se tuvo en cuenta que las palabras "adulto", "algunos" y "cada-uno" requerían una complejidad computacional adicional por simular el gesto con más de dos movimientos en diferentes manos, las palabras "el-ella", "ellas-ellos", "ellos-dos/ustedes-dos", sus movimientos y posición de la mano son muy similares y al momento de entrenar el algoritmo se presentaría sobre posición de las clases afectando la clasificación de la red neuronal, por estos motivos se escogieron las palabras anciano, bebe, joven, homosexual y hombre de la misma categoría ya que estas palabras en su gesticulación son diferentes realizando el gesto con una sola mano.

Cada palabra es grabada desde tres ángulos: de frente, desde la derecha y desde la izquierda; cada video tiene una duración en promedio de 3 segundos y fueron grabados con una cámara Canon EOS Rebel T6 Sensor CMOS (APS-C) de 18 Mp y procesador Digic 4+.[38]

Este conjunto de datos está conformado por 740 videos distribuidos por cada palabra como se muestra en la ilustración 16.

[38] CANON. EOS Rebel T6 Comparte lo impresionante con todo el mundo. [En línea]. En: CANON [Citado el 3 de mayo, 2020]. disponible en: https://www.canon.com.mx/productos/fotografia/camaras-eos-reflex/p/1139

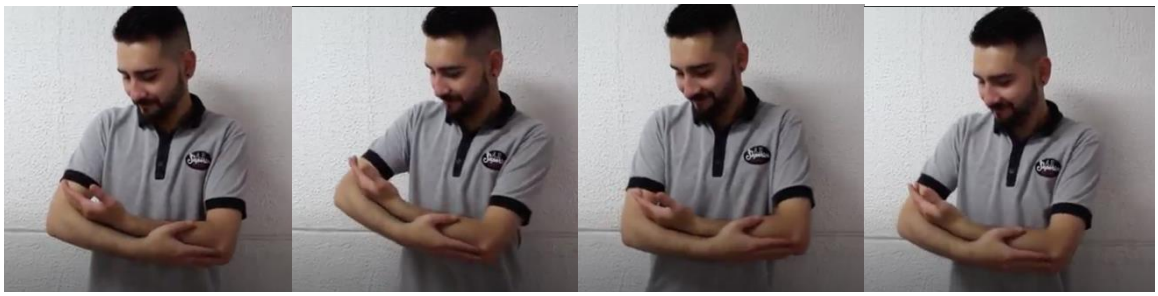
Ilustración 16 Conjunto de videos



Preprocesamiento

Se implementa un algoritmo para extraer los fotogramas de cada uno de los videos anteriormente mencionados como se muestra en la ilustración 17, estos fotogramas fueron organizados según su clase y almacenados de manera local, es importante que el nombre de carpeta donde se guarden los fotogramas coincida con el nombre de las clases correspondientes a las palabras: “bebe”, “anciano”, “joven”, “homosexual” y “hombre”.

Ilustración 17 Fotogramas de la seña bebe en LSC.



Se generaron tres conjuntos de datos con el fin de comprobar cuál de estos permite obtener mejores resultados al medir el modelo, el primero se creó con fotogramas extraídos por cada cambio de movimiento que realiza la persona, el segundo conformado con los fotogramas extraídos cada 0.01 segundos y el ultimo con extracción de fotogramas cada 0.04 segundos, como se puede evidenciar en la tabla 1.

Tabla 1 Conjunto de datos fotogramas

Intervalo	Palabra	Cantidad de fotogramas
0.01s	Joven	13.867
	Bebe	14.443
	Hombre	18.157
	Homosexual	13.114
	Anciano	14.393
0.04s	Joven	3.403
	Bebe	3.636
	Hombre	4.558
	Homosexual	3.203
	Anciano	3.499
Por cada fotograma	Joven	13.121
	Bebe	13.729
	Hombre	12.279
	Homosexual	12.870
	Anciano	15.094

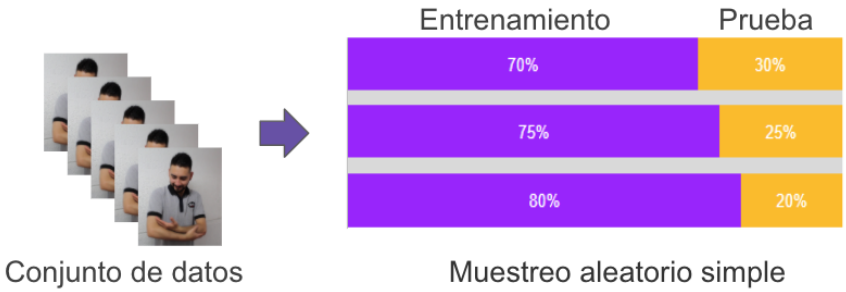
A cada fotograma del conjunto de datos se le aplico el intercambio de canales de color BGR a RGB como se muestra en la ilustración 18, lo cual permite la compatibilidad del uso de las librerías OpenCV y Keras también se realiza una redimensión del fotograma original a 224 x 224 px, ya este tamaño es el estándar para poder aplicar la red neuronal convolucional.

Ilustración 18 Intercambio de canales de color para OpenCV

Muestreo

En esta fase se utiliza el método de muestreo simple aleatorio para dividir el conjunto de datos en dos partes, la primera utilizada para el entrenamiento y la segunda para pruebas, con el fin de lograr identificar qué distribución del conjunto de datos es la más adecuada para obtener los mejores resultados en la ejecución del modelo se realizaron 3 muestreos, uno de 70% y 30% otro de 75% y 25% y el último de 80% y 20% del conjunto de datos para entrenamiento y pruebas respectivamente como se muestra en la ilustración 19.

Ilustración 19 Muestreo del conjunto de datos.



La cantidad de fotogramas resultado de la división de los conjuntos de datos realizada para el muestreo se ilustra en la tabla 2.

Tabla 2 Cantidad de fotogramas del muestreo

Conjunto de datos	Tipo	Fotogramas					
		Porcentaje	Cantidad	Porcentaje	Cantidad	Porcentaje	Cantidad
0,01 segundos	Entrenamiento	70%	51.782	75%	55.481	80%	59.179
	Prueba	30%	22.192	25%	18.494	20%	14.795
0,04 segundos	Entrenamiento	70%	12.809	75%	13.724	80%	14.639
	Prueba	30%	5.490	25%	4.575	20%	3.660

Implementación de la red neuronal

En esta fase se implementa una red neuronal convolucional ResNet 50, para realizar el entrenamiento de la red se utilizó el modelo pre entrenado de ImageNet para Keras el cual contiene los pesos y sesgos que representan las características de un conjunto de datos disponibles en línea los cuales fueron transferidos a nuestro modelo, permitiéndonos ahorrar tiempo y recursos informáticos.[39]

Posterior a este paso se continuo con el entrenamiento del modelo con los conjuntos de datos construidos, haciendo uso de la extracción de características anteriormente mencionada, para esto se realizaron las siguientes configuraciones en el algoritmo, se creó un arreglo con las etiquetas de las palabras a clasificar como se muestra en la ilustración 20, como se mencionó anteriormente, estas etiquetas deben coincidir con el nombre de las carpetas donde se almacenaron los conjuntos de datos que contienen cada uno de los fotogramas de los diferentes videos, también se realizó la configuración del hiperparámetro epoch con un valor de 100 con el fin de garantizar mejores resultados.

Ilustración 20 Configuración de etiquetas.

```
LABELS = set(["bebe",
              "anciano",
               "hombre",
               "homosexual",
               "joven"])
```

El algoritmo de entrenamiento genera los siguientes archivos: activity.model que es el modelo clasificador basado en ResNet50 para reconocer las cinco palabras en lenguaje de señas colombiano y el archivo lb.pickle que es un binarizador de etiquetas en serie que contiene las etiquetas de las clases. Estos archivos son insumos para la fase de clasificación.

[39] Kaggle. *Kaggle.* [En línea] Kaggle. [Citado el: 16 de 02 de 2020.] https://www.kaggle.com/keras/resnet50#imagenet_class_index.json.

Para esta fase fue necesaria la instalación de las siguientes librerías

- imutils
- Tensorflow 2.1
- Tensorflow-estimator 2.1
- anaconda navigator
- keras
- cv2
- sklearn
- matplotlib
- pickle
- Tensorflow-gpu 2.1

Con el fin de optimizar el tiempo y la infraestructura de la máquina en la cual se ejecuta el experimento, el procesamiento del algoritmo se realiza por medio de una GPU de referencia NVIDIA GeForce RTX 2080 con una capacidad de cómputo 7.5GB[40]. En el proceso de configuración se deben instalar las siguientes librerías para que el algoritmo se ejecute utilizando la GPU.

- Tensorflow
- tensorflow-tensorboard
- tensorflow-gpu
- anaconda cudatoolkit=10.1
- anaconda cudnn

Clasificación

En esta fase se implementa un nuevo algoritmo el cual toma como parámetro de entrada un video de prueba de una persona simulando un palabra en el lenguaje de señas colombiano, el algoritmo lo segmenta en fotogramas detectando los movimientos realizados por la persona, a estos fotogramas se les aplica un redicionamiento de 224 x 224 px y se convierten del sistema de colores BGR a RGB, tal cual como se realiza en la etapa de preprocesamiento, el siguiente parámetro de entrada para el algoritmo son los archivos activity.model, lb.pickle generados en la fase de implementación de la red neuronal, haciendo uso del promedio móvil

[40] NVIDIA. *Recommended GPU for Developers.* [En línea] NVIDIA. [Citado el: 05 de 03 de 2020.] https://developer.nvidia.com/cuda-gpus.

disponible en la librería Python collections el cual permite hacer la clasificación de la palabra en lenguaje de señas colombiano simulada en el video la cual se escribe en los fotogramas, el script carga los paquetes y módulos necesarios que se nombran a continuación.

- keras
- load_model
- deque
- numpy
- argparse
- pickle
- cv2

Posteriormente se escribe la palabra encontrada en la imagen en forma de subtítulo y se genera un nuevo video como se muestra en la figura 21. el cuál será la salida del algoritmo.

Ilustración 21 Clasificación del LSC en video

Medición

En esta fase se mide el desempeño del algoritmo de aprendizaje profundo implementado. Para esto se utiliza las métricas precisión, exhaustividad, f1-score para cada palabra y la exactitud que se calcula para cada modelo generado, los resultados obtenidos después de entrenar cada modelo, están reflejados en las siguientes tablas.

Tabla 3 Mediciones muestreo 70-30 por cada fotograma.

Conjunto de datos	Palabra	Precisión	Exhaustividad	f1-score	Muestreo
Por cada fotograma	*anciano*	0,45	0,86	0,59	70% - 30%
	bebe	0,93	0,62	0,74	
	hombre	0,65	0,27	0,38	
	homosexual	0,63	0,5	0,55	
	joven	0,55	0,63	0,59	
	Exactitud	-	-	0,48	
	macro avg.	0,64	0,58	0,57	

Tabla 4 Mediciones muestreo 70-30 fotogramas cada 0,04s

Conjunto de datos	Palabra	Precisión	Exhaustividad	f1-score	Muestreo
Fotograma cada 0,04s	Anciano	0,02	0,84	0,42	70% - 30%
	bebe	0,78	0,75	0,76	
	hombre	0,89	0,18	0,29	
	homosexual	0,98	0,06	0,11	
	joven	0,51	0,5	0,51	
	Exactitud			0,46	
	macro avg	0,69	0,47	0,42	

Tabla 5 Mediciones muestreo 70-30 fotogramas cada 0,01s

Conjunto de datos	Palabra	Precisión	Exhaustividad	f1-score	Muestreo
Fotograma cada 0,01s	Anciano	0,61	0,82	0,7	70% - 30%
	bebe	0,89	0,81	0,85	
	hombre	0,91	0,92	0,91	
	homosexual	0,92	0,76	0,83	
	joven	0,72	0,95	0,82	
	Exactitud			0,83	
	macro avg	0,81	0,85	0,82	

Tabla 6 Mediciones muestreo 75-25 por cada fotograma.

Conjunto de datos	Palabra	Precisión	Exhaustividad	f1-score	Muestreo
Por cada fotograma	Anciano	0,44	0,86	0,58	75% - 25%
	bebe	0,75	0,64	0,69	
	hombre	0,52	0,29	0,37	
	Homosexual	0,6	0,51	0,55	
	joven	0,59	0,55	0,57	
	Exactitud			0,47	
	macro avg	0,58	0,57	0,55	

Tabla 7 Mediciones muestreo 75-25 fotogramas cada 0,04s

Conjunto de datos	Palabra	Precisión	Exhaustividad	f1-score	Muestreo
Fotograma cada 0,04s	Anciano	0,27	0,9	0,41	75% - 25%
	bebe	0,82	0,72	0,77	
	hombre	0,94	0,09	0,16	
	homosexual	0,82	0,02	0,03	
	joven	0,49	0,4	0,44	
	Exactitud			0,42	
	macro avg	0,67	0,43	0,36	

Tabla 8 Mediciones muestreo 75-25 fotogramas cada 0,01s

Conjunto de datos	Palabra	Precisión	Exhaustividad	f1-score	Muestreo
Fotograma cada 0,01s	Anciano	0,39	0,95	0,55	75% - 25%
	bebe	0,93	0,9	0,91	
	hombre	0,91	0,79	0,85	
	homosexual	0,95	0,6	0,74	
	joven	0,73	0,69	0,71	
	Exactitud			0,76	
	macro avg	0,78	0,79	0,75	

Tabla 9 Mediciones muestreo 80-20 por cada fotograma.

Conjunto de datos	Palabra	Precisión	Exhaustividad	f1-score	Muestreo
Por cada fotograma	Anciano	0,48	0,65	0,55	80% - 20%
	bebe	0,88	0,59	0,71	
	hombre	0,64	0,26	0,37	
	homosexual	0,72	0,37	0,49	
	joven	0,52	0,68	0,59	
	Exactitud			0,49	
	macro avg	0,65	0,51	0,54	

Tabla 10 Mediciones muestreo 80-20 fotogramas cada 0,04s

Conjunto de datos	Palabra	Precisión	Exhaustividad	f1-score	Muestreo
Fotograma cada 0,04s	Anciano	0,3	0,77	0,43	80% - 20%
	bebe	0,71	0,8	0,76	
	hombre	0,94	0,1	0,19	
	homosexual	0,89	0,16	0,27	
	joven	0,47	0,57	0,52	
	Exactitud			0,47	
	macro avg	0,66	0,48	0,43	

Tabla 11 Mediciones muestreo 80-20 fotogramas cada 0,01s

Conjunto de datos	Palabra	Precisión	Exhaustividad	f1-score	Muestreo
Fotograma cada 0,01s	Anciano	0,57	0,82	0,67	80% - 20%
	bebe	0,87	0,84	0,85	
	hombre	0,88	0,86	0,87	
	homosexual	0,83	0,84	0,83	
	joven	0,69	0,69	0,69	
	Exactitud			0,8	
	macro avg	0,77	0,81	0,78	

4.1 Instalaciones y equipo requerido

Las instalaciones y equipo requerido para el desarrollo del proyecto se describen a continuación:

Instalaciones

- Sede el claustro de la universidad católica de Colombia
- Salas de informática de la universidad católica de Colombia
- Residencia personal de cada integrante del proyecto

Equipo de hardware

- 1 computador portátil con: Procesador Intel Core i7 2.00GHz 8 GB de memoria RAM y 400 GB de espacio disponible
- 1 computador portátil con: procesador i7 8va Generación, 8 GB de RAM y 500GB de espacio disponible.
- 1 computador con: Procesador Intel Core i7 de 3.4GHz, 32GB de memoria RAM, GPU NVIDIA GeForce RTX 2080 y 500 GB de espacio libre.

Software

- Anaconda con Python 3.7
- IDE Júpiter notebook
- Sistema operativo Windows
- Librerías
- Google Drive 100GB de almacenamiento.

5. RESULTADOS

5.1 Construcción conjunto de datos

Como resultado de la creación del conjunto de datos el cual se utilizó para la ejecución del experimento y se puede visualizar en la tabla número 12 se grabaron en total 740 videos en formato .MOV.

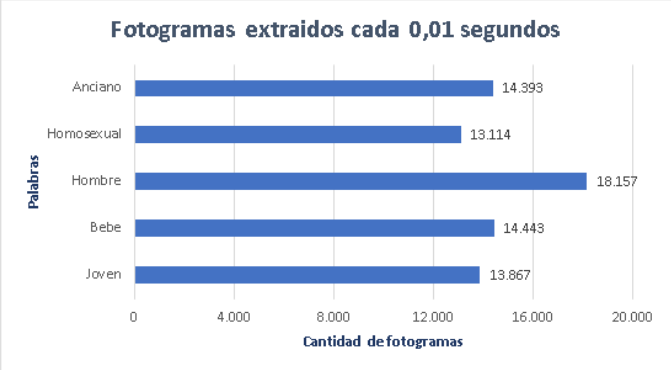
Tabla 12 Consolidado de videos grabados en formato .MOV

Palabra	Total videos
Hombre	147
Joven	149
Anciano	149
Bebe	151
Homosexual	144
Total General	740

Posterior a la grabación de los videos como se indica en la metodología que se desarrolló, se realizó una extracción de fotogramas por cada video en diferentes tiempos quedando así 3 conjuntos de datos como se visualiza en la tabla 1 del capítulo diseño metodológico. A continuación se muestran los resultados de la extracción de fotogramas cada 0,01 segundos en la ilustración 22 con un total de 73.974 fotogramas extraídos correspondiente al experimento con los resultados más óptimos de las 3 ejecuciones que se realizaron.

Ilustración 22 Fotogramas extraídos cada 0,01 segundo

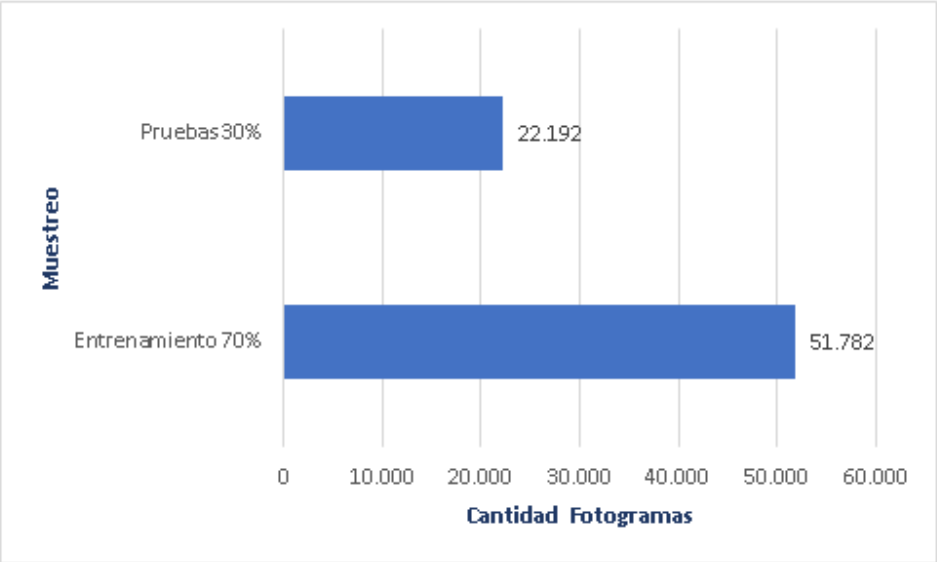
Intervalo	Palabra	Cantidad de fotogramas
0.01s	Joven	13.867
	Bebe	14.443
	Hombre	18.157
	Homosexual	13.114
	Anciano	14.393
	Total	73.974



En la fase correspondiente al muestreo se utilizó un porcentaje de 70% de los fotogramas para entrenamiento y 30% para pruebas del total de fotogramas obtenidos en la ilustración 23 para un consolidado de 51.782 fotogramas que se utilizaron para entrenamiento y 22.192 de pruebas como se evidencia en la ilustración 23.

Ilustración 23 Total de fotogramas utilizados en muestreo 70/30.

Intervalo	Muestreo	Cantidad de fotogramas
0.01s	Entrenamiento 70%	51.782
	Pruebas 30%	22.192



En el capítulo de anexos se indican las url de los repositorios de libre acceso donde se encuentra esta información.

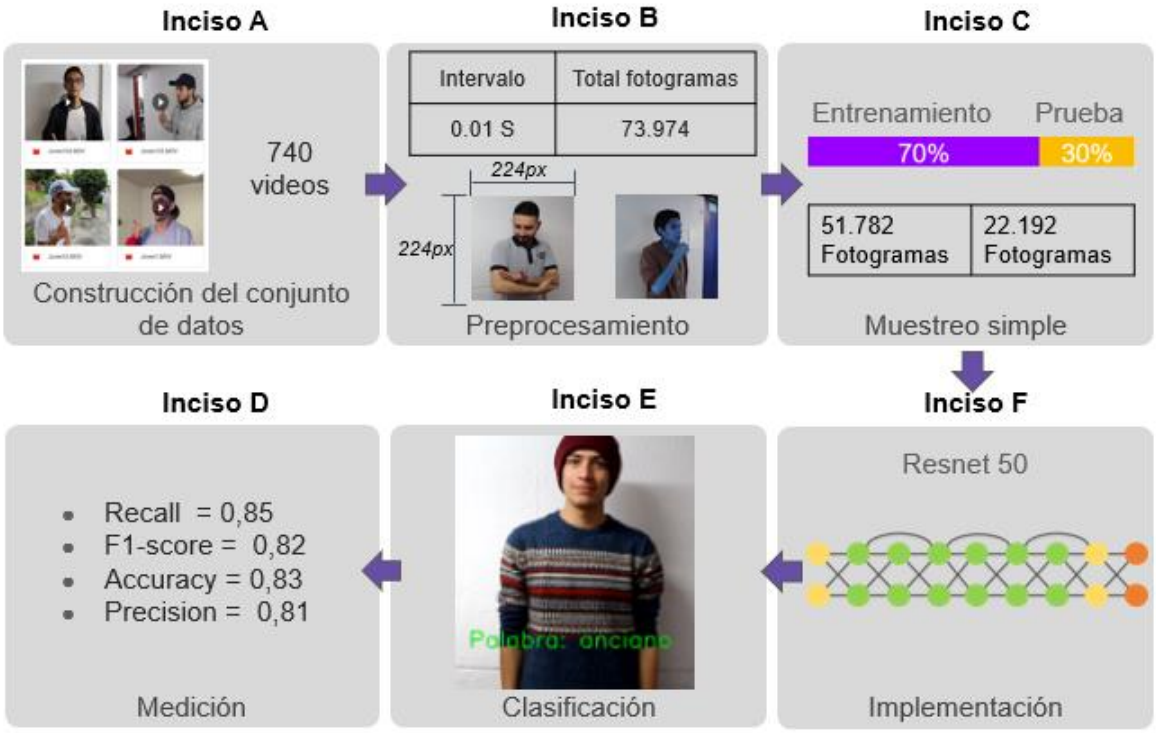
5.2 Diseño de la estrategia metodológica

Como resultado de la investigación del estado del arte se diseñó una estrategia metodológica conformada por seis fases, la cual fue aplicada para la clasificación de palabras del lenguaje de señas colombiano donde se utilizaron los 740 videos grabados como se muestra en el inciso A de la ilustración 24, la segunda fase de la metodología que comprende el preprocesamiento del conjunto de datos construido, se utilizan los fotogramas extraídos con el intervalo de 0,01 segundos con un total de 73.974 los cuales fueron redimensionados a 224px x 224px y convertidos del sistema de colores BGR a RGB como se muestra en el inciso B de la ilustración 24; en la fase de muestreo se utiliza la técnica de muestreo aleatorio simple para dividir el conjunto de datos en dos partes que corresponden a entrenamiento y pruebas con una proporción del 70% de entrenamiento que corresponden a 51.782 fotogramas y 30% para pruebas que corresponden a 22.192 fotogramas como se muestra en el inciso C de la ilustración 24.

Para la fase de implementación de la metodología se ejecuta el algoritmo que se describe en el apartado Diseño del algoritmo, donde se utiliza una red convolucional ResNet 50 y un modelo pre entrenado de ImageNet de Keras, para esta ejecución se utiliza el muestreo resultado de la fase anterior y se obtiene como salida un

modelo para la clasificación de las cinco palabras del lenguaje de señas, en el inciso E de la ilustración 24 se muestra el resultado de la clasificación de las palabras en lenguaje de señas colombiano la cual se obtiene de la aplicación del modelo generado en la fase de implementación a un video de prueba, lo que genera el mismo video con un subtítulo de la palabra que se ejecuta en el video, en el inciso D de la ilustración 24 se evidencian los resultados obtenidos de la aplicación de las métricas precisión , exhaustividad, f1-score y la exactitud calculada para el modelo generado obteniendo unos valores satisfactorios en la aplicación del diseño metodológico diseñado.

Ilustración 24 Resultado de la metodología

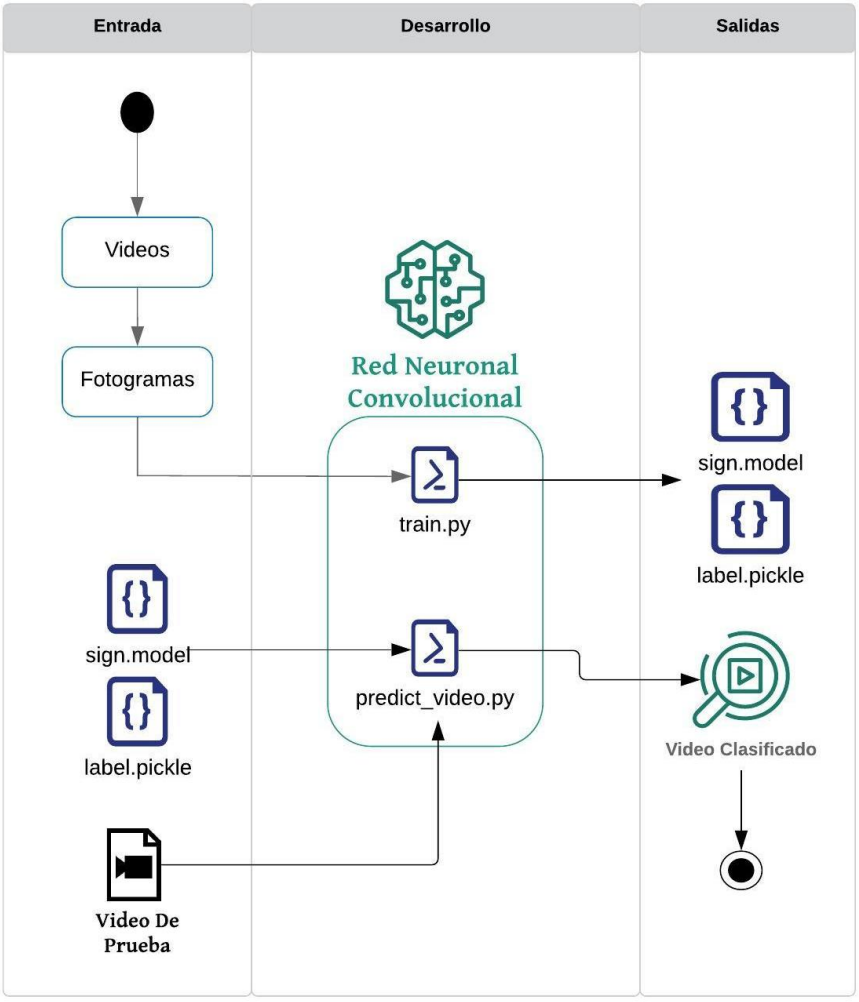


Fuente: Los autores

5.3 Desarrollo del algoritmo

En la ilustración 25 se puede visualizar el funcionamiento de los algoritmos implementados en la red neuronal convolucional, como primera entrada se tiene el conjunto de datos en video y el pre procesamiento de los fotogramas, este último ingresa como parámetro de entrada a la red neuronal convolucional exactamente al algoritmo “train.py” el cual procesa los fotogramas y genera dos archivos de salida “sign.model” y “label.pickle”, los cuales son parámetros de entrada para el algoritmo “predict_video.py” junto al video de prueba el cual será la salida final del experimento etiquetando el video con la clasificación correspondiente a las palabras en LSC hombre, bebe, anciano, homosexual y joven.

Ilustración 25 Funcionamiento algoritmo



El primer algoritmo “train.py” es el que se encarga de las etapas de:

- Lectura y almacenamiento de fotogramas.
- Implementación de la red neuronal convolucional.
- Almacenamiento y serialización del archivo .model y .pickle.

A continuación de describe el funcionamiento del algoritmo train.py

Como todo algoritmo requiere de la importación de librerías con el objetivo de que sus invocaciones funcionen correctamente, este algoritmo adicional a las librerías de *sklearn* y *keras* requiere las siguientes precondiciones que se visualizan en la ilustración 26, consiste en la definición de los argumentos de línea de comandos en la cual se establecen las rutas para el conjunto de datos de entrada, el camino hacia el archivo de modelo de Keras de salida, la ruta del archivo pickle de binarizador de etiqueta de salida y la cantidad épocas que va entrenar la red, por defecto se establece el valor en 25 pero al realizar el entrenamiento de la red neuronal convolucional el valor se aumenta a 100.

Ilustración 26 Precondiciones de ejecución

```
#PRECONDICIONES DEL ALGORITMO
#generación de argumentos de entrada (dataset, model, label_bin)
    ap = argparse.ArgumentParser()
#se define el epoch=100 //Hiperparametro red neuronal
    ap.add_argument("-e", "--epochs", type=int, default=25,
        help="# of epochs to train our network for")
```

Posterior a la definición de las librerías y precondiciones se procede a la lectura y almacenamiento de los fotogramas como se puede visualizar en la ilustración 27 definiendo los arreglos de la clase con las palabras en LSC, la data de imágenes y etiquetas. Se recorre la estructura de carpetas donde se encuentran el conjunto de datos (fotogramas) establecidos en los argumentos de línea de comandos del paso anterior y por último se define el muestreo asignándole al conjunto de datos un 70% para entrenamiento y 30% para pruebas.

Ilustración 27 Lectura y almacenamiento de fotogramas

```
8                #LECTURA Y ALMACENAMIENTO DE FOTOGRAMAS
9   #Definición de arreglo con los nombres de las clases
10      LABELS = set(["bebe", "anciano", "hombre", "homosexual", "joven"])
11  #Definición de arreglo para las imagenes[] y labels[]
12      imagePaths = list(paths.list_images(args["dataset"]))
13      data = []
14      labels = []
15  #Lectura de imágenes y labels según el conjunto de datos de entrada
16  #almacenándolos en los arreglos imagenes[] y labels[]
17      for imagePath in imagePaths:
18          label = imagePath.split(os.path.sep)[-2]
19          if label not in LABELS:
20      continue
21
22      image = cv2.imread(imagePath)
23      data.append(image)
24      labels.append(label)
25  #Definición del muestreo 0.30 y 0.70
26      (trainX, testX, trainY, testY) = train_test_split(data, labels,
27                  test_size=0.30, stratify=labels, random_state=42)
```

Como se visualiza en la ilustración 28 se procede a la implementación de la red neuronal convolucional basándose en los pesos que el modelo Resnet50 previamente ha aprendido de ImageNet, de esta manera nos permite reutilizar la red Resnet50 sin volver a entrenarla optimizando el consumo de recursos. Posterior a este paso se procede compilar nuestro modelo por medio de la optimización del descenso de gradiente estocástico (SGD) consumiendo la API de Keras para la clase model.compile(), model.fit_generator(), para el entrenamiento y model.predict(), para la evaluación de la red.

Ilustración 28 Implementación red neuronal convolucional

```
28
29              #IMPLEMENTACIÓN RED NEURONAL CONVOLUCIONAL
30
31  #Carga la red neuronal resnet 50 del modelo pre entrenado de ImageNet
32      baseModel = ResNet50(weights="imagenet", include_top=False,
33                           input_tensor=Input(shape=(224, 224, 3)))
34  #Compilación CNN
35      opt = SGD(lr="", momentum="", decay="" / args["epochs"])
36      model.compile(loss="", optimizer=opt,metrics="")
37  #Entrenamiento
38      H = model.fit_generator(trainAug.flow(trainX, trainY, batch_size=""),
39                              epochs=args["epochs"])
40  #Evalua la red
41      predictions = model.predict(testX, batch_size="")
```

Por último se guarda el modelo con la función de la API de Keras model.save() y se serializa el binarizador de etiquetas con la función de pickle generando así los dos archivos de entrada (sign.model y label.pickle) para el algoritmo predict_video.py como se muestra en la ilustración 29.

Ilustración 29 Almacenamiento archivos .model y .pickle

```
43      #ALMACENAMIENTO MODELO sign.model / label.pickle
44  #Serializa el modelo y el label y lo guarda en disco
45  model.save(args["model"])
46  f = open(args["label_bin"], "wb")
47  f.write(pickle.dumps(lb))
48  f.close()
49
```

El segundo algoritmo “predict_video.py” es el que se encarga de las etapas de:

- Lectura del video de prueba, modelo y etiquetas.
- Clasificación del video de prueba.
- Generación del video de prueba con la clasificación de las palabras en LSC.

A continuación de describe el funcionamiento del algoritmo predict_video.py

Para su ejecución se requieren las siguientes precondiciones que se visualizan en la ilustración 30, consiste en la definición de los argumentos de línea de comandos en la cual se establecen las rutas para el conjunto de datos de entrada y de salida para los archivos .model, .pickle y el video de entrada de prueba, adicional a esats configuraciones se realiza la lectura de los archivos .model, .pickle y el video de prueba.

Ilustración 30 Lectura del video de prueba, modelo y etiquetas

```
1                #PRECONDICIONES DEL ALGORITMO
2  #Generación de argumentos de entrada (--model, --label-bin, input_video, output)
3  ap = argparse.ArgumentParser()
4  #Lectura de los archivos de entrada .model / .pickle / video de prueba
5  model = load_model(args["model"])
6  lb = pickle.loads(open(args["label_bin"], "rb").read())
7  vs = cv2.VideoCapture(args["input"])
```

En un ciclo while se procede a realizar la lectura de cada uno de los fotogramas que componen el video de prueba aplicandole una transformación de color BGR a RGB, y un redimensionamiento de 224X224 pixeles, a cada fotograma se le aplica el modelo pre entrenado lo cual permite clasificar el video con las etiquetas obtenidas de la ejecución del algoritmo “train.py” como se evidencia en la ilustración 31, posteriormente se procede con la escritura del video con un label en la parte inferior donde se indica la palabra en lenguaje de señas de colombia en español. Para finalizar se almacena el video resultante en formato .avi.

Ilustración 31 Clasificación video de prueba

```
                    #CLASIFICACIÓN DEL VIDEO DE PRUEBA
while True:

        frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        frame = cv2.resize(frame, (224, 224)).astype("float32")

        preds = model.predict(np.expand_dims(frame, axis=0))[0]
        Q.append(preds)

        results = np.array(Q).mean(axis=0)
        i = np.argmax(results)
        label = lb.classes_[i]

        text = "Palabra: {}".format(label)
        cv2.putText(output, text, (500, 900), cv2.FONT_HERSHEY_SIMPLEX,
                3.25, (0, 255, 0), 5)

        writer = cv2.VideoWriter(args["output"], fourcc, 30,
                        (W, H), True)
        writer.write(output)

        cv2.imshow("Output", output)
        key = cv2.waitKey(1) & 0xFF

        if key == ord("q"):
                break

```

Finalmente en la ilustración 32 se puede visualizar el pseudocodigo completo del algoritmo implementado en el proyecto.

Ilustración 32 Pseudocódigo del algoritmo

```
ALGORITHM train_py
    datasetPath ← "/dataset"  // The path to the input dataset.
    epochs ← 100 // Epochs to train network
    words ← ["bebe", "anciano", "hombre","homosexual","joven"] // Set of class labels
    imagePaths[] ← createPaths(datasetPath) // Grab the list of images in dataset directory

    FOR EACH imagePath IN imagePaths:
        label ← imagePath // Extract the class label

        IF label NOT IN words:
            continue  //Ignore any label not in the labels set.
        END IF

        image ← readImage(imagePath) // Load and preprocess an image
        image ←  swappingColorChannelsBGR2RGB(image) // Swapping color channels
        image = resize(image, (224, 224)) // Resizing to 224×224px
        data[] ← add(image)  //The image is added to the data list
        labels[] ← add(label) //The label is added to the data labels list
    END FOR

    // Partition the data into training and testing splits using 75%
      of the data for training and the remaining 25% for testing
    train[] ← getVideosTraining(data[], 0.70)
    test[] ← getVideosTest(data[], 0.30)

    baseModel ← getModelResNet50Imagenet() //Load ResNet50 pre-trained with ImageNet weights

    FOR EACH layer IN baseModel.layers
        layer.trainable <- False //  Freeze the baseModel so that it will not be trained via BP
    END FOR

    setUp[0] ← featureExtractorActivation <- ReLu // Activation function  in convolutional layers
    setUp[1] ← classificationActivation ← softmax // Activation function for multiclass classification

    // Compile model with the Stochastic Gradient Descent (SGD) optimizer
    opt ← createSGD(epochs)
    mymodel ← compileModel(opt)
    trainModel(mymodel, train[], test[])  // Trains the network
    save(mymodel)  // Store Keras model

    //Serialize and store label binarizer in pickle format.
    serializeLabels(labels)
    save(labels)
END ALGORITHM
```

5.4 Medición del algoritmo desarrollado

Como resultado de la medición se hace énfasis en la tabla número 13 replicada de del subcapítulo medición correspondiente al muestreo del 70/30 con los resultados del experimento.

Tabla 13 Copia de la tabla número 5 de la sección medición.

Conjunto de datos	Palabra	Precisión	Exhaustividad	f1-score	Muestreo
Fotograma cada 0,01s	Anciano	0,61	0,82	0,7	70% - 30%
	bebe	0,89	0,81	0,85	
	hombre	0,91	0,92	0,91	
	homosexual	0,92	0,76	0,83	
	joven	0,72	0,95	0,82	
	Exactitud			0,83	
	macro avg	0,81	0,85	0,82	

6. DISCUSIÓN DE RESULTADOS

En la tabla 14 se evidencia los resultados obtenidos para las medidas de clasificación exactitud y f1 score para los experimentos realizados sobre las cinco clases de las palabras del lenguaje de señas. Se puede evidenciar que la medida de exactitud para el muestreo de 80/20 sobre el conjunto de datos conformado por cada fotograma corresponde al 0,49 es decir que de 100 casos que valido el modelo en 49 acertó correctamente la clasificación de las cinco clases entrenadas, para la selección de fotogramas de 0,04s en la misma tasa de muestreo el resultado fue de 0,47 es decir que de 100 casos que valido el modelo en 47 acertó de manera correcta, de igual forma para la medida de clasificación de F1 Score los resultados más altos corresponden al conjunto de datos por cada fotograma con un muestreo del 70/30 y para los fotogramas extraídos cada 0,04s equivalente al muestreo 80/20 es decir que el resultado de f1 score el cual concierne a la media ponderada de la precisión y la exhaustividad corresponden al 0,57 y 0,43 respectivamente.

Tabla 14 Consolidación de resultados por experimento exactitud y f1 score.

Exactitud			
Conjunto de datos (Extracción fotogramas)	**Muestreo**		
	70 - 30	**75 - 25**	**80 - 20**
Por cada fotograma	0,48	0,47	0,49
Fotograma cada 0,04s	0,46	0,42	0,47

F1 Score			
Conjunto de datos (Extracción fotogramas)	**Muestreo**		
	70 - 30	**75 - 25**	**80 - 20**
Por cada fotograma	0,57	0,55	0,54
Fotograma cada 0,04s	0,42	0,36	0,43

Con el análisis de los resultados anteriormente mencionados se identificó que los experimentos realizados con los conjuntos de datos de extracción de fotogramas por cada 0,04s y por cada fotograma no fueron aceptables, por esta razón se realizó un tercer experimento con resultados favorables para su interpretación, como se puede evidenciar en las tablas 5, 8, 11 donde se obtiene un incremento en los valores obtenidos para el conjunto de datos resultado del proceso de extracción de fotogramas cada 0,01s en los muestreos de 70/30, 75/25 y 80/20, siendo la ejecución del muestreo correspondiente al 70/30 la de mejor desempeño obtenido como se puede observar en la tabla número 5 con valores por encima de 0,7 para las medidas de f1 score, exactitud y macro avg, de igual forma las medidas individuales por cada clase exactamente para la medida de precisión se obtuvieron resultados de 0,92 y 0,91 para las clases de homosexual y hombre y para las medidas de exhaustividad los resultados mostraron índices de 0,95 y 0,92 de las clases joven y hombre obteniendo así los resultados esperados para la ejecución del proyecto.

El modelo bajo la arquitectura de una red neuronal convolucional sustentada en una ResNet 50 presenta un efectivo método de clasificación de las cinco palabras del lenguaje de señas colombiano una vez el conjunto de datos haya sido previamente redimensionado para la ejecución de la red permitiéndole al modelo optimizar el

entrenamiento mejorando sustancialmente los tiempos de ejecución como se puede visualizar en los resultados obtenidos del rendimiento del experimento.

Se ha comprobado que el uso de la red entrenada con los diferentes muestreos de datos específicamente los experimentos realizados con los porcentajes 75/25 y 80/20 con extracción de fotogramas cada 0.01s también se obtienen resultados aceptables por lo tanto en la etapa de pre procesamiento es determinante realizar esta segmentación de frames para el éxito de la ejecución de la red neuronal convolucional resnet 50.

Con estos resultados podemos concluir que es posible tomar el código fuente del experimento como punto de partida para la generación de nuevos proyectos donde se utilice un conjunto de datos más extenso el cual contenga una variedad diferente de palabras en el lenguaje de señas colombiano gesticulando más de una palabra a la vez, inclusive se podría optar por la clasificación de palabras compuestas y así lograr la traducción de una oración completa, ya que el data set utilizado para la ejecución del proyecto fue grabado con poca y alta luminosidad simulando un uso real en condiciones naturales logrando así resultados satisfactorios.

7. CONCLUSIONES

- De los conjuntos de datos que se construyeron el que obtuvo mejores resultados fue el de los fotogramas extraídos cada 0,01 segundos con un muestreo de 70% de entrenamiento y 30% para pruebas, ya que al realizar la extracción de fotogramas en un intervalo de tiempo menor la red neuronal profunda puede procesar una mayor cantidad de fotogramas detectando los cambios de movimiento que realiza la persona grabada en video.

- El planteamiento de la estrategia metodológica aporto una serie de etapas consecutivas la cual apoyo la implementación de la red neuronal ResNet 50 logrando implementar el algoritmo de aprendizaje profundo de manera satisfactoria en la clasificación de las cinco palabras en lenguaje de señas colombiano.

- El desarrollo del algoritmo puede ser tomado como base para la clasificación de diferentes palabras en lenguaje de señas colombiano detectando movimientos más complejos lo cual implicaría realizar la construcción de un nuevo conjunto de datos que cumpla con las características necesarias al momento de grabar los videos.

- Los resultados que se obtuvieron de las medidas precisión, exactitud y exhaustividad fueron aceptables en el proceso de clasificación de las cinco palabras en el lenguaje de señas colombiano ya que estuvieron por encima del 70%.

8. RECOMENDACIONES

- Para el desarrollo y ejecución de un algoritmo basado en aprendizaje profundo en el cual se requiere el tratamiento de un conjunto de datos en imágenes se recomienda el uso dx|e la herramienta Google Colab ya que permite programar y ejecutar algoritmos en Python 2 o Python 3 haciendo uso de Jupyter Notebooks y procesamiento por GPU de manera remota. Aunque cabe aclarar que la herramienta tiene ciertas restricciones ya que a nivel de infraestructura en la nube ofrece solo 12 GB de RAM y 50 GB de almacenamiento disponibles para el uso.

- Para el entrenamiento de una red neuronal que requiera una gran cantidad de imágenes se recomienda que el conjunto de datos se encuentre previamente preprocesados, es decir se realicen las transformaciones y redimensiones necesarias a estas imágenes con el fin de optimizar la ejecución del script de entrenamiento ya que en la ejecución si las imágenes no son previamente ajustadas se pueden presentar saturaciones en la infraestructura del equipo.

- En el proceso de construcción del conjunto de datos, es recomendable que al momento de grabar los videos se realice a partir del inicio de la ejecución de la seña para evitar que en la etapa de preprocesamiento se realice la eliminación de fotogramas en los cuales no se esté ejecutando ninguna seña.

- Se recomienda continuar con la construcción del conjunto de datos de todas las palabras de la lengua de señas colombiana ya que es importante contar con este insumo para futuras investigaciones.

- Se recomienda implementar una aplicación que permita traducir a español las señas gesticuladas por una persona con discapacidad auditiva en tiempo real permitiendo una comunicación unidireccional con una persona oyente.

- Se recomienda a tiempo futuro realizar la implementación del algoritmo en tiempo real en una interfaz gráfica de usuario ya sea en una aplicación web o móvil y utilizarlo en las fundaciones de sordomudos existentes en la ciudad de Bogotá.

9. Bibliografía

Asch, Vincent Van. 2013. *Macro- and micro-averaged evaluation.* 2013.

Ataka. *Ataka.* [En línea] Machine Learning y Deep Learning: cómo entender las claves del presente y futuro de la inteligencia artificial. https://www.xataka.com/robotica-e-ia/machine-learning-y-deep-learning-como-entender-las-claves-del-presente-y-futuro-de-la-inteligencia-artificial.

Barrios, Juan. Big data en salud. *Big data en salud.* [En línea] Juan Barrios. [Citado el: 21 de 03 de 2020.]

—. 2019. Juan Barrios. *Juan Barrios.* [En línea] Big Data, 2019. https://www.juanbarrios.com/matriz-de-confusion-y-sus-metricas/.

Barros, Laura. 2014. El Universal. *El Universal.* [En línea] El Universal, 28 de Mayo de 2014. [Citado el: 20 de 04 de 2019.] https://www.eluniversal.com.co/tecnologia/software/software-para-sordos-finalista-en-mundial-de-emprendimiento-160945-DXEU253808.

Bécares, Bárbara. 2015. siliconweek. *siliconweek.* [En línea] 20 de 05 de 2015. [Citado el: 15 de 01 de 2020.] https://www.siliconweek.com/cloud/siel-software-colombiano-para-comprender-el-lenguaje-de-signos-59195.

BETANCUR, D.B., GÓMEZ, M.V. and PALACIO, A.P. 2013. *Traducción Automática Del Lenguaje Dactilológico De Sordos y Sordomudos Mediante Sistemas adaptativos.* s.l. : ProQuest Central, 2013. ISSN 19099762..

BOTINA-MONSALVE, D.J., DOMÍNGUEZ-VÁSQUEZ, M.A. MADRIGAL-GONZÁLEZ y CASTRO-OSPINA. 2018. *Clasificación automática de las vocales en el lenguaje de señas.* 2018.

BROOKSHEAR , J. Glean. 1993. *Teoría de la computación.* s.l. : Addisson Wesley iberoamericana Wilmington Delaware, 1993.

BROOKSHEAR, J. Glean. 1993. *Teoría de la computación.* s.l. : Addisson Wesley iberoamericana Wilmington Delaware, 1993.

Burgal, Jesús Utrera. 2018. Deep Learning básico con Keras (Parte 4): ResNet. *Deep Learning básico con Keras (Parte 4): ResNet.* 2018.

Casal, Jordi y Mateu, Enric. 2003. *TIPOS DE MUESTREO.* Barcelona : s.n., 2003. 08193.

Colombia, El Congreso de. 1996. LEY 324 DE 1996 . *LEY 324 DE 1996, por el cual se crean algunas normas a favor de la población sorda.* Bogotá : s.n., 1996.

Cristina, Mata maria. 1997. *Cómo elaborar muestras para los sondeos de audiencias. Cuadernos de investigación No 5.* Quito : MACASSI, 1997.

DANE. 2010. DANE Información para todos. *DANE Información para todos.* [En línea] DANE, Marzo de 2010. [Citado el: 25 de Abril de 2019.] https://www.dane.gov.co/index.php/estadisticas-por-tema/demografia-y-poblacion/discapacidad.

GAGO, ENCARNACIÓN. 2019. Inteligencia Artificial, Machine Learning & Deep Learning. *Inteligencia Artificial, Machine Learning & Deep Learning.* Madrid : s.n., 2019.

Geron, Aurelien. 2017. *Hands-On Machine Learning with Scikit-Learn and TensorFlow.* 2017.

He, Kaiming. 2016. github. *github.* [En línea] github.com, 29 de 06 de 2016. [Citado el: 20 de 03 de 2020.] https://github.com/KaimingHe/deep-residual-networks#models.

Image-net.org. 2020. . Imagenet. *Imagenet.* [En línea] Image-net.org, 30 de 03 de 2020. . http://www.image-net.org/about-overview.

Ingeniero Colombiano Crea Traductor De Lenguaje De Señas Para Sordos. **2013.** Madrid : ProQuest Central, 2013.

INSOR. 2017. INSOR. *INSOR.* [En línea] 02 de 2017. [Citado el: 21 de 10 de 2019.] http://www.insor.gov.co/bides/wp-content/uploads/archivos/caracterizacion_acceso_perm_grad_estudiantes_sordos_ies.pdf.

—. 2017. INSOR instituto nacional para sordos. *INSOR instituto nacional para sordos.* [En línea] 2017. [Citado el: 21 de 09 de 2019.] http://www.insor.gov.co/bides/info-general/.

—. 2012. INSOR Instituto nacional para sordos. *INSOR Instituto nacional para sordos.* [En línea] INSOR, 01 de 2012. [Citado el: 21 de 10 de 2019.] http://www.insor.gov.co/bides/experiencia-universidad-ecci/.

—. 2017. INSOR Instituto Nacional Para Sordos. *INSOR Instituto Nacional Para Sordos.* [En línea] Instituto Caro y Cuervo, Instituto Nacional para sordos (Insor), 2017. [Citado el: 21 de 10 de 2019.] http://www.insor.gov.co/descargar/diccionario_basico_completo.pdf.

1997. Instituto Colombiano. *Instituto Colombiano.* [En línea] 26 de 09 de 1997. [Citado el: 08 de 02 de 2020.] https://www.icbf.gov.co/cargues/avance/docs/decreto_2369_1997.htm.

Jordi, TORRES. 2018. *DEEP LEARNING Introducción práctica con Keras.* 2018.

Kaggle. *Kaggle.* [En línea] Kaggle. [Citado el: 16 de 02 de 2020.] https://www.kaggle.com/keras/resnet50#imagenet_class_index.json.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. 2016. *Deep Residual Learning for Image Recognition.* EEUU : s.n., 2016.

—. 2016. *Deep Residual Learning for Image Recognition.* EEUU : s.n., 2016.

NVIDIA. NVIDIA. *Recommended GPU for Developers.* [En línea] NVIDIA. [Citado el: 05 de 03 de 2020.] https://developer.nvidia.com/cuda-gpus.

PICHUCHO, Javier P., et al. 2019. *Interpretación de lenguaje de señas ecuatoriano empleando visión por computador.* Quito : Revista Ibérica de Sistemas e Tecnologias de Informação, 2019.

pickle. Python object serialization. *Python object serialization.* [En línea] [Citado el: 13 de 04 de 03.] https://docs.python.org/3/library/pickle.html.

PINEDA, Beatriz, DE ALVARADO, Beatriz y DE CANALES, Francisca. 1994. *Metodología de la investigación, manual para el desarrollo de person al de salud.* Washington : Organización Panamericana de la Salud, 1994.

Procesamiento del lenguaje natural. **Vásquez Cortez, Augusto, Vega Huerta, Hugo y Paronia Quispe , Jaime. 2009.** San Marcos : Revista de ingenieria de sistemas e informatica, 2009.

Ramírez. , Paulina . *Métodos de enseñanza del español a los niños sordos: actitudes de los profesionales.* Santafé de Bogotá : Universidad Iberoamericana.

republica, Congreso de la. 1996. *LEY 324 DE 1996.* Colombia, : s.n., 1996.

Richard L. Scheaffer, William Mendenhall, Lyman Ott. 2007. *Elementary survey sampling.* Magallanes : Thomson, 2007.

Roma, J. C., Rué, A. B., & Bagén, T. L. 2019. *eep learning: principios y fundamentos.* Barcelona, España : s.n., 2019.

Saha, Sumit. 2018. A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way. *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way.* 2018.

Salesforce. Salesforce Blog. *Salesforce Blog.* [En línea] [Citado el: 05 de 08 de 2019.] https://www.salesforce.com/mx/blog/2017/6/Que-es-la-inteligencia-artificial.html..

SAS. 2019. Inteligencia Artificial. *Inteligencia artificial Qué es y por qué es importante.* Bogotá : s.n., 2019.

—. 2019. Software y Soluciones De Analitica. *Software y Soluciones De Analitica.* [En línea] 21 de 10 de 2019. https://www.sas.com/es_co/insights/analytics/deep-learning.html.

SENA, Servicio Nacional de Aprendizaje. 2017. Perdiodico SENA. *Perdiodico SENA.* [En línea] Servicio Nacional de Aprendizaje SENA , 10 de 2017. [Citado el:

11 de 10 de 2019.] http://periodico.sena.edu.co/inclusion-social/noticia.php?t=comunidad_sorda&i=86.

Spot. 2019. La diferencia entre Inteligencia Artificial, Machine Learning y Deep Learning. *La diferencia entre Inteligencia Artificial, Machine Learning y Deep Learning.* 2019.

ZHANG, Chen-Lin, et al. 2017. *In defense of fully connected layers in visual representation transfer.* Springer, Cham : En Pacific Rim Conference on Multimedia, 2017.

10. ANEXOS

Anexo A: Conjuntos de datos

Se publica en el repositorio del conjunto de datos utilizados en la implementación de este proyecto en la herramienta KAGGLE.

- Conjunto de datos en video de las palabras bebe, anciano, joven, homosexual y hombre en la lengua de señas colombiana.

  URL:https://www.kaggle.com/luisfelipemoreno/conjunto-de-datos

- Conjunto de datos de los fotogramas generados por cada cambio de movimiento en los videos.

  URL:https://www.kaggle.com/luisfelipemoreno/conjuntodedatos-fotogramas-lsc

- Conjunto de datos de los fotogramas generados cada 0,04 segundos.

  URL:https://www.kaggle.com/luisfelipemoreno/conjuntodedatos-fotogramas004s-lsc

- Conjunto de datos de los fotogramas generados cada 0,01 segundos.

  URL:https://www.kaggle.com/luisfelipemoreno/conjuntodedatosfotogramas001slsc

Anexo B: Código del proyecto

Se publica el código fuente utilizado en el proyecto para el pre procesamiento, entrenamiento y pruebas.

- URL: https://github.com/ypmunoz36/kerasDeepLSC