

Máster Data Science and Business Analytics

DETECTOR DE LENGUA DE SIGNOS

Ing. Pedro Cortés Soberanes

Ing. Daniel Marques Olavarrieta

Tutor Académico: Daniel Rubio Yagüe

Madrid, 12 de mayo de 2023

Índice de contenidos

Introducción	3
Capítulo I. Tema de estudio	4
I.1 Planteamiento del problema	4
I.2 Objetivo general de la investigación	6
I.2.1 Objetivos específicos	6
I.3 Limitaciones	7
Capítulo II. Marco teórico	8
II.1 Introducción	8
II.2 Lengua de Signos	9
II.2.1 Características de la lengua de signos	9
II.2.3 Alfabeto Dactilológico Español	10
II.3 Inteligencia Artificial	10
II.4 Python	11
II.4.1 PyCharm	12
II.4.2 NumPy	14
II.4.3 OpenCV	15
II.4.4 MediaPipe	16
II.4.5 TensorFlow	16
II.4.6 Cvzone & Handtracking	16
II.4.7 Python Text to Speech	17
II.4.8 Keras	17
II.5 Visión Computacional	18
II.6 Teachable Machine	19
Capítulo III. Marco Experimental	21
III.1 Tipo y alcance de la investigación	21

III.2 Método de desarrollo de software	22
III.2.1 Tecnologías a utilizar	22
III.2.2 Fases del proyecto	23
III.2.2.1 Fase 1 - Recolección de datos	23
III.2.2.2 Fase 2 - Creación del modelo	25
III.2.2.3 Fase 3 - Testing	27
III.2.2.4 Fase 4 - Procesamiento de predicción a voz	27
Conclusiones	28
Recomendaciones	30
Referencias bibliográficas	31

Introducción

En la sociedad actual, la comunicación efectiva es un elemento esencial para el desarrollo personal, social y profesional. Sin embargo, existen personas con discapacidades auditivas, visuales y/o dificultades en el habla que enfrentan desafíos significativos al interactuar con los demás. Estas barreras comunicativas limitan su acceso a la información y dificultan su participación plena en diversos contextos.

En este contexto, las tecnologías de asistencia juegan un papel fundamental al proporcionar soluciones innovadoras y accesibles para superar estas barreras. Una de las áreas de estudio y desarrollo en este campo es la interpretación y traducción de la lengua de signos, un medio de comunicación utilizado por las personas con discapacidad auditiva para expresar sus ideas, emociones y necesidades.

En este trabajo de fin de máster, se aborda el desafío de desarrollar un sistema de interpretación de la lengua de signos española utilizando técnicas de visión por computadora, aprendizaje automático y procesamiento de lenguaje natural. El objetivo principal es lograr una comunicación fluida y efectiva entre personas con discapacidades auditivas y personas sin estas limitaciones, eliminando las barreras comunicativas y promoviendo la inclusión.

El presente trabajo se estructura en diferentes capítulos que abarcan desde la investigación y aprendizaje de la lengua de signos española, pasando por el diseño y desarrollo de un algoritmo de detección y reconocimiento de gestos, hasta la implementación de un sistema de traducción y síntesis de voz. Se utilizarán tecnologías y herramientas como Python, OpenCV, TensorFlow y Teachable Machine para alcanzar los objetivos planteados.

Además, se presentarán resultados obtenidos a partir de pruebas y evaluaciones realizadas, así como recomendaciones para mejorar y expandir el sistema en futuras investigaciones. El trabajo tiene como propósito no solo brindar una solución

tecnológica innovadora, sino también contribuir a la inclusión y mejorar la calidad de vida de las personas con discapacidad auditiva.

En resumen, este trabajo de fin de máster busca aportar avances significativos en el campo de la comunicación inclusiva a través del desarrollo de un sistema de interpretación de la lengua de signos española. La integración de tecnologías y enfoques interdisciplinarios permitirá superar las barreras comunicativas y promover la inclusión, abriendo nuevas oportunidades para una sociedad más igualitaria y accesible.

Capítulo I. Tema de estudio

I.1 Planteamiento del problema

En la actualidad, la inclusión de todas las personas, independientemente de su género, sexo, religión, capacidades o limitaciones, es un tema de gran importancia en nuestra sociedad globalizada. La promoción de la igualdad de oportunidades y el respeto mutuo son pilares para construir un mundo más justo y equitativo.

Dentro de este contexto, las personas con discapacidad auditiva, visual y dificultades en el habla se enfrentan a desafíos significativos a la hora de comunicarse y participar plenamente en la sociedad. La comunicación es un aspecto esencial en nuestras vidas y constituye la base de nuestras interacciones diarias. Por lo tanto, resulta fundamental abordar las barreras existentes para garantizar la inclusión de todas las personas en igualdad de condiciones.

Por otro lado, de acuerdo con el Presidente de la Fundación IDDEAS, Luis Casado Aguilera, las personas con discapacidad visual se encuentran con dificultades para percibir y entender la lengua de signos; ya que dependen principalmente de la comunicación verbal y del tacto. Además, las personas con dificultades en el habla también enfrentan obstáculos significativos para expresarse y ser entendidos por los demás (Luis Casado, comunicación personal, 16 de marzo de 2023).

La lengua de signos se ha convertido en un medio de comunicación vital para las personas con discapacidad auditiva, permitiéndoles expresarse de manera efectiva a través de gestos y movimientos de manos. No obstante, la falta de conocimiento y comprensión de la lengua de signos por parte de las personas sin discapacidad auditiva se convierte en un obstáculo que dificulta una comunicación fluida e inclusiva.

En la era actual, caracterizada por el rápido avance tecnológico y la creciente digitalización, resulta imprescindible aprovechar las oportunidades que ofrece la inteligencia artificial para abordar estas necesidades sociales y fomentar la inclusión. La inteligencia artificial tiene un gran potencial para superar las barreras de comunicación al permitir la detección y traducción automática de la lengua de signos en tiempo real.

Si bien es cierto que aún enfrentamos desafíos técnicos en el desarrollo de un detector preciso y confiable de lengua de signos a través de inteligencia artificial, se han logrado avances significativos en la detección de signos y gestos. La variabilidad en los gestos y expresiones propios de la lengua de signos, así como la identificación de la intención del gesto y la adaptación a diferentes estilos de signos, son aspectos que se deben abordar para lograr una comunicación fluida e inclusiva.

En este contexto, el presente trabajo de fin de máster se enfoca en el diseño y desarrollo de un sistema de detección de lengua de signos a través de inteligencia artificial, en consonancia con el Objetivo de Desarrollo Sostenible de la ONU número 10: Reducción de las desigualdades, el cuál busca reducir las desigualdades y promover la inclusión para todas las personas.

I.2 Objetivo general de la investigación

Crear una herramienta, llamada Linguai, que permita a las personas con discapacidad auditiva, visual y dificultades en el habla comunicarse de manera natural y fluida con aquellos que no entiendan o perciban la lengua de signos.

I.2.1 Objetivos específicos

- Adquirir los conocimientos básicos necesarios sobre la lengua de signos española para comprender su estructura y vocabulario de manera efectiva.
- Desarrollar un algoritmo de detección de gestos del alfabeto español utilizando inteligencia artificial antes de mayo de 2023, que logre una precisión mínima del 80% en la identificación de las letras.
- Implementar un algoritmo de traducción de gestos detectados a texto antes de mayo de 2023, que sea capaz de convertir de manera precisa y eficiente los gestos del lenguaje de signos español en texto comprensible.
- Diseñar e integrar un algoritmo de síntesis de voz antes de mayo de 2023 que convierta el texto traducido a voz natural y fluida, con el objetivo de facilitar la comunicación oral a las personas sin conocimiento de la lengua de signos o dificultad visual.
- Realizar una evaluación exhaustiva del funcionamiento del proyecto antes de mayo de 2023, mediante pruebas y análisis rigurosos, para asegurar la efectividad y usabilidad de la herramienta en situaciones reales de comunicación.

I.3 Limitaciones

Las limitaciones encontradas en la elaboración de la presente investigación se mencionan a continuación:

Primeramente, considerando que los involucrados en el proyecto carecían de conocimiento referentes a la lengua de signos española, las primeras semanas solo fueron de aprendizaje sobre esta lengua gestual.

Es importante destacar que, hasta el momento, el enfoque del proyecto se centra en la detección de signos y gestos en la lengua de signos española, mientras que el reconocimiento del movimiento aún está siendo explorado como un área de desarrollo futuro.

Capítulo II. Marco teórico

II.1 Introducción

Este apartado, con el objetivo de sustentar la información presentada en el trabajo de investigación, contiene los principales fundamentos teóricos de las diferentes áreas de conocimiento, que son necesarios para lograr una mayor comprensión de la investigación. A continuación, se muestra en la Figura 1 un diagrama representativo que esquematiza la revisión bibliográfica.

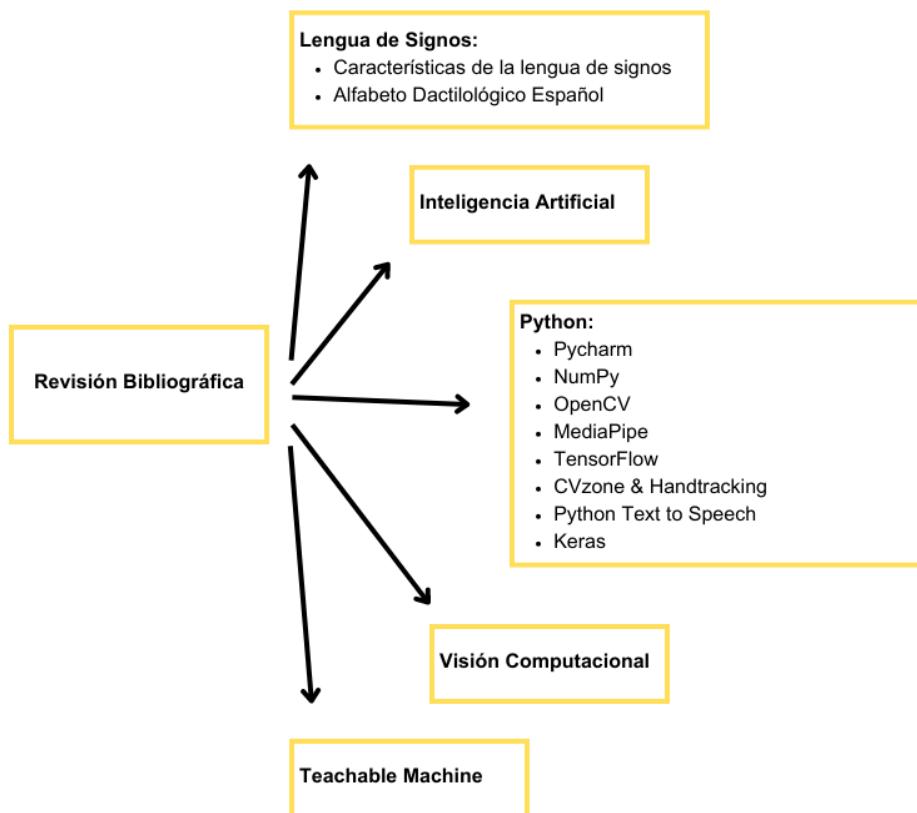


Figura 1. Esquema del Marco Teórico

II.2 Lengua de Signos

De acuerdo con la Federación de Personas Sordas de la Comunidad de Madrid, la lengua de signos, es una lengua de carácter visual y gestual con gramática propia que permite la recepción del mensaje mediante la visión y la emisión del mismo a través de las manos, el cuerpo y la expresión; permitiendo la comunicación con otras personas.

La lengua de signos no es universal, al igual que ocurre con las lenguas orales, cada país puede tener una o más lenguas de signos que se han desarrollado en sus distintas zonas geográficas. Actualmente en España existen dos, la Lengua de Signos Española (LSE) y la Lengua de Signos Catalana (LSC).

La principal forma de comunicación de las personas sordas es la lengua de signos. Gracias a ella estructuran y expresan sus sentimientos, opiniones, pensamientos, ideas y creencias. (Instituto de Enseñanza de la Lengua de Signos Española, 2018)

II.2.1 Características de la lengua de signos

Conforme a lo expuesto por la Sociedad Federada de Personas Sordas de Málaga (SFSM), la lengua de signos se caracteriza por:

- Utilizar la comunicación gestual y visual, frente a las lenguas orales que utilizan la comunicación oral y auditiva.
- Se articula con las manos, brazos, tronco y rostro.
- Utiliza el espacio y la simultaneidad con significado lingüístico, llamado espacio de signación, donde los brazos alcanzan comodidad para realizar signos y gestos con las manos.
- Es una lengua no universal, ya que cada país ha desarrollado su propia lengua de signos, incluso se consiguen diferencias entre distintas comunidades del mismo país.

II.2.3 Alfabeto Dactilológico Español

El Alfabeto Dactilológico Español o también conocido como abecedario de la lengua de signos española, no es una lengua en sí misma, sino que es un recurso lingüístico o de comunicación de la Lengua de Signos.



Figura 2. Alfabeto Dactilológico

Fuente: Fundación ONCE

II.3 Inteligencia Artificial

De acuerdo a la definición del equipo de Google, la inteligencia artificial es un campo científico que se centra en el desarrollo de sistemas informáticos y máquinas capaces de realizar tareas que requieren habilidades de razonamiento, aprendizaje y actuación similares a las de los seres humanos. Esta disciplina abarca diversos ámbitos, como la informática, el análisis de datos, la estadística, la ingeniería de hardware y software, la lingüística, la neurociencia, así como la filosofía y la psicología.

En términos operativos y empresariales, la inteligencia artificial se basa en tecnologías como el aprendizaje automático y el aprendizaje profundo, que permiten el análisis de datos, la generación de predicciones y pronósticos, la categorización de objetos, el procesamiento de lenguaje natural, las recomendaciones y la recuperación inteligente de datos, entre otros. Estas herramientas y técnicas potencian la toma de decisiones y la automatización de procesos en diversos campos de aplicación.

II.4 Python

Según el equipo de Amazon Web Service, Python es un popular lenguaje de programación ampliamente empleado en diversos campos, como el desarrollo de aplicaciones web, el software, la ciencia de datos y el aprendizaje automático (machine learning). Los programadores eligen Python debido a su eficiencia y su curva de aprendizaje accesible, lo que facilita su adopción por parte de desarrolladores de diferentes niveles de experiencia. Además, Python es un lenguaje multiplataforma, lo que significa que puede ejecutarse en distintos sistemas operativos.

El software Python está disponible de forma gratuita y se destaca por su gran versatilidad y flexibilidad. Gracias a su amplia comunidad de desarrolladores, Python cuenta con una amplia variedad de bibliotecas y frameworks que facilitan el desarrollo de aplicaciones de manera rápida y eficiente. Estas herramientas adicionales se integran de manera fluida con Python, lo que acelera el proceso de desarrollo y permite aprovechar al máximo las funcionalidades del lenguaje.



Figura 3. Python Logo

Fuente: (1000 marcas, 2022)

La naturaleza de código abierto de Python, junto con su compatibilidad con una amplia gama de sistemas y su capacidad para interactuar con otros lenguajes de programación, lo convierten en una elección popular tanto para proyectos pequeños como para aplicaciones empresariales de gran escala. Su popularidad en el campo de la ciencia de datos y el machine learning se debe a la gran cantidad de bibliotecas especializadas disponibles, como NumPy, OpenCV, cvZone, Math, Mediapipe y TensorFlow, que facilitan la manipulación y análisis de datos, así como la implementación de algoritmos de aprendizaje automático.

En resumen, Python es un lenguaje de programación poderoso y versátil que ha ganado reconocimiento en numerosos campos debido a su eficiencia, facilidad de uso y su amplia comunidad de desarrolladores. Su capacidad para trabajar en diversos sistemas operativos y su integración con otras herramientas y bibliotecas hacen de Python una opción popular para el desarrollo de aplicaciones y proyectos de datos y aprendizaje automático.

II.4.1 PyCharm

Acorde con la documentación de Pycharm de JetBrains, Pycharm es un entorno de desarrollo o IDE, *integrated development environment* por sus siglas en inglés, el cual permite hacer análisis del código, debugo, pruebas de unidad y control de versiones.

Algunas de las funcionalidades destacadas:

- Ejecución de comandos instantáneos: PyCharm permite ejecutar comandos en tiempo real tanto en un Jupyter notebook como en la consola de Python. Además, proporciona una visualización clara de las variables creadas, lo que facilita el análisis y la exploración en profundidad de los DataFrames y las NumPy Series.

- Visualización de gráficos: El IDE muestra los gráficos directamente dentro de la interfaz, lo que permite un análisis visual más cómodo. Además, PyCharm realiza un seguimiento de los gráficos más recientes, lo que facilita el cambio entre diferentes gráficos y mejora la productividad en el análisis de datos.
- Compatibilidad con bibliotecas científicas: PyCharm ofrece una compatibilidad integrada con bibliotecas científicas populares como Pandas, NumPy, Matplotlib y otras más. Además de brindar inteligencia de código avanzada, también proporciona herramientas adicionales como visores de vectores para facilitar el análisis y la manipulación de datos científicos.
- Gestión de entornos Conda: PyCharm facilita la creación y selección de entornos Conda aislados por proyecto. Esto permite mantener las dependencias separadas, evitando conflictos y asegurando un entorno de desarrollo limpio y coherente para cada proyecto.
- Complemento de R: Además de su soporte completo para Python, PyCharm cuenta con un complemento para trabajar con el lenguaje R. Este complemento permite realizar cálculos estadísticos utilizando R, aprovechando las características del IDE como la asistencia a la codificación, la depuración visual, la ejecución inteligente y las herramientas de vista previa.

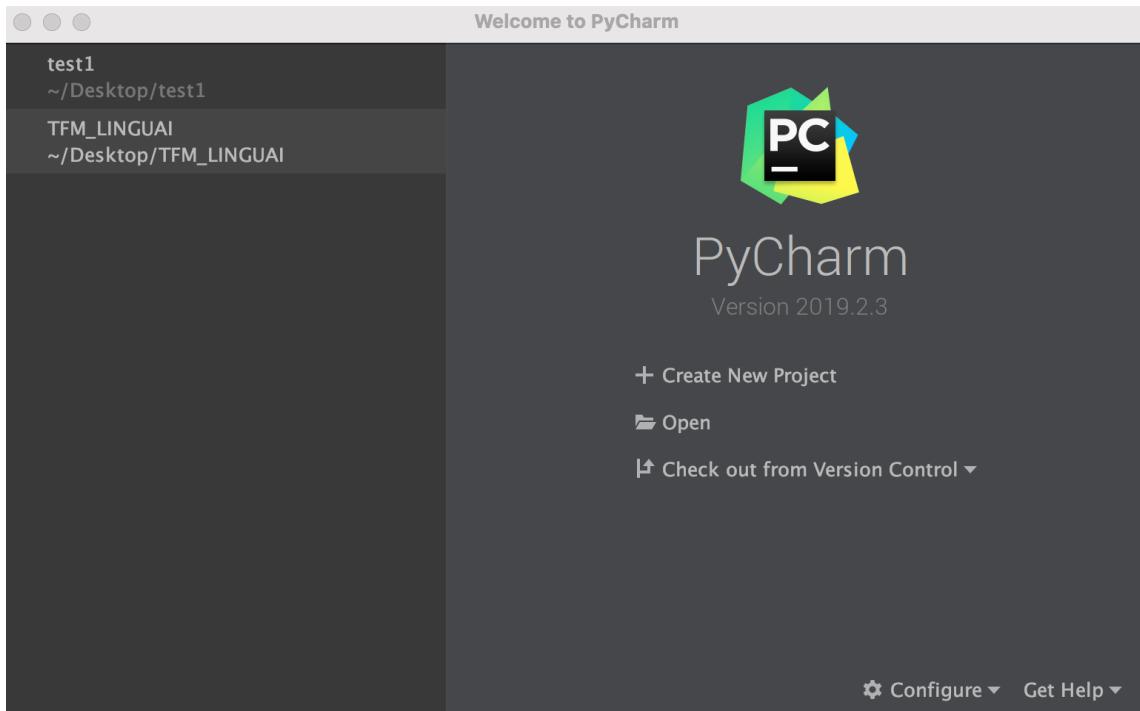


Figura 4. Pycharm

II.4.2 NumPy

NumPy (Numeric Python), según la Escuela de Ingenierías Industriales de la Universidad de Valladolid, es un paquete fundamental en Python diseñado para el manejo eficiente de matrices y el cálculo numérico. Proporciona una amplia gama de herramientas y funciones especializadas para trabajar con matrices de una manera altamente eficiente.

Una de las características distintivas de NumPy es su capacidad para manejar matrices homogéneas, donde todos los datos son del mismo tipo, lo que permite un almacenamiento interno eficiente y un acceso rápido a los elementos. Esto contrasta con las listas de Python, que ofrecen flexibilidad pero penalizan el rendimiento al manejar matrices de gran tamaño.

NumPy incorpora una nueva clase de objetos llamados "arrays", que permiten representar colecciones de datos en múltiples dimensiones. Estos arrays proporcionan

operaciones y funciones optimizadas para el procesamiento rápido de datos numéricos y científicos.

Una ventaja significativa de NumPy radica en su rendimiento superior en comparación con las listas predefinidas de Python. El procesamiento de arrays en NumPy es considerablemente más rápido, hasta 50 veces más rápido, lo cual lo convierte en la elección ideal para el manejo de vectores y matrices de grandes dimensiones.

II.4.3 OpenCV

De acuerdo con el Ing. Rafael Marín, de INESEM Business School, OpenCV (Open Source Computer Vision) es una biblioteca de visión por computadora ampliamente utilizada en el procesamiento de imágenes y vídeo. Proporciona una amplia gama de funciones y algoritmos para el análisis y tratamiento de imágenes mediante técnicas de inteligencia artificial.

Con el módulo cv2 de OpenCV, los desarrolladores tienen acceso a una interfaz fácil de usar que les permite trabajar con funciones de procesamiento de imágenes y vídeo. Esta biblioteca es especialmente útil en campos como la fotografía, el marketing y la seguridad, donde se utilizan técnicas de detección de rostros y objetos.

OpenCV es un proyecto de código abierto que comenzó como una iniciativa de investigación en Intel y se ha convertido en la biblioteca de visión por computadora más grande en términos de funcionalidad, con más de 2500 algoritmos implementados.

En pocas palabras, OpenCV es una potente biblioteca de visión por computadora que ofrece una amplia variedad de funciones y algoritmos para el procesamiento de imágenes y vídeo, incluyendo la detección de rostros y objetos. Es ampliamente utilizada en diferentes aplicaciones que requieren análisis y tratamiento de imágenes mediante técnicas de inteligencia artificial.

II.4.4 MediaPipe

Acorde con la documentación de Google, MediaPipe es un framework que permite crear canalizaciones de aprendizaje automático para datos de series temporales, como video y audio. Ofrece soluciones de ML de autoservicio con abstracciones fáciles de usar, permitiendo la personalización, evaluación, prototipado e implementación mediante una API de bajo código o estudio sin código. Con la experiencia de Google ML, MediaPipe proporciona soluciones avanzadas de ML optimizadas *end-to-end*, incluyendo aceleración de hardware, siendo lo suficientemente liviano para funcionar eficientemente en cualquier dispositivo. Algunas de sus funciones incluyen el rastreo y etiquetado de objetos en imágenes, identificación de contenido en imágenes, y reconocimiento de gestos de las manos. En resumen, MediaPipe facilita el desarrollo de soluciones de aprendizaje automático en dispositivos para diversas tareas y aplicaciones.

II.4.5 TensorFlow

TensorFlow, de acuerdo a la página oficial de TensorFlow.org, es una librería de código libre especializada en Machine Learning (ML) que fue desarrollada por Google. Permite construir y entrenar redes neuronales para detectar patrones y razonamientos utilizados por los humanos. TensorFlow procesa y carga datos de entrada para crear modelos de aprendizaje automático previamente entrenados o modelos personalizados, brindando así una plataforma completa para el desarrollo y aplicación de algoritmos de ML.

II.4.6 Cvzone & Handtracking

HandTracking de cvzone es un paquete de visión por computadora que utiliza las bibliotecas OpenCV y Mediapipe para facilitar el procesamiento de imágenes y funciones de IA. Es capaz de detectar y realizar un seguimiento de una mano en una imagen de *input*, centrándose en su movimiento y orientación. A su vez, resuelve el

problema de tener que escribir código repetitivo en diferentes proyectos para realizar el seguimiento manual de la mano. Mediante la creación de un módulo de seguimiento manual, es posible importarlo en cualquier proyecto de Python y realizar el seguimiento de la mano de forma rápida, sencilla y eficiente. (Kiruri, 2022)

II.4.7 Python Text to Speech

Pyttsx3, también conocido como Python Text to Speech, es una potente librería que ofrece capacidades de síntesis de voz computarizada. Con esta herramienta, es posible llevar a cabo la transformación de texto escrito en lenguaje humano a una salida de audio que simula de manera realista el habla humana. Mediante algoritmos avanzados, Pyttsx3 permite generar una experiencia auditiva inmersiva, proporcionando una voz natural y comprensible para la comunicación entre máquinas y seres humanos. Su integración sencilla y su capacidad para personalizar las voces y los ajustes de pronunciación la convierten en una opción valiosa para aplicaciones de texto a voz en el entorno de programación Python. (Rockikz, 2020)

II.4.8 Keras

Según la guía digital de IONOS, Keras es una biblioteca de código abierto escrita en Python, que se centra en acelerar la creación de redes neuronales. Diseñada como una interfaz intuitiva (API), Keras permite acceder y desarrollar varios frameworks de aprendizaje automático, como TensorFlow, Theano y Microsoft Cognitive Toolkit. Esta biblioteca, desarrollada por François Chollet de Google, proporciona una forma eficiente y flexible de construir arquitecturas de redes neuronales, incluyendo redes convolucionales y recurrentes, para entrenar modelos de deep learning. Desde su lanzamiento en 2015 como parte del proyecto ONEIROS, Keras se ha convertido en una herramienta poderosa y versátil en el campo del aprendizaje automático.

II.5 Visión Computacional

Acorde con el equipo de IBM, La Visión Computacional es un campo de la inteligencia artificial que permite a las computadoras analizar imágenes y videos para obtener información relevante y tomar decisiones basadas en ella. A través de algoritmos y el procesamiento de datos visuales, la visión por computadora capacita a las máquinas para ver, observar y comprender su entorno de manera similar a la visión humana, aunque en un tiempo mucho más corto y utilizando cámaras y datos en lugar de órganos sensoriales biológicos.

Al replicar las capacidades visuales humanas, el *Computer Vision* permite a las computadoras identificar objetos, reconocer personas y entender el contenido visual en imágenes y videos. Su aplicabilidad abarca diversas áreas, y se utiliza en numerosas innovaciones y soluciones modernas. La tecnología de visión por computadora puede implementarse tanto en entornos de nube como en sistemas locales, brindando flexibilidad en su uso y aplicaciones. Con su capacidad para analizar grandes cantidades de datos visuales en tiempo real, la visión computacional supera las capacidades humanas en términos de velocidad y precisión en tareas como la detección de defectos o problemas en la producción industrial y otras aplicaciones prácticas.

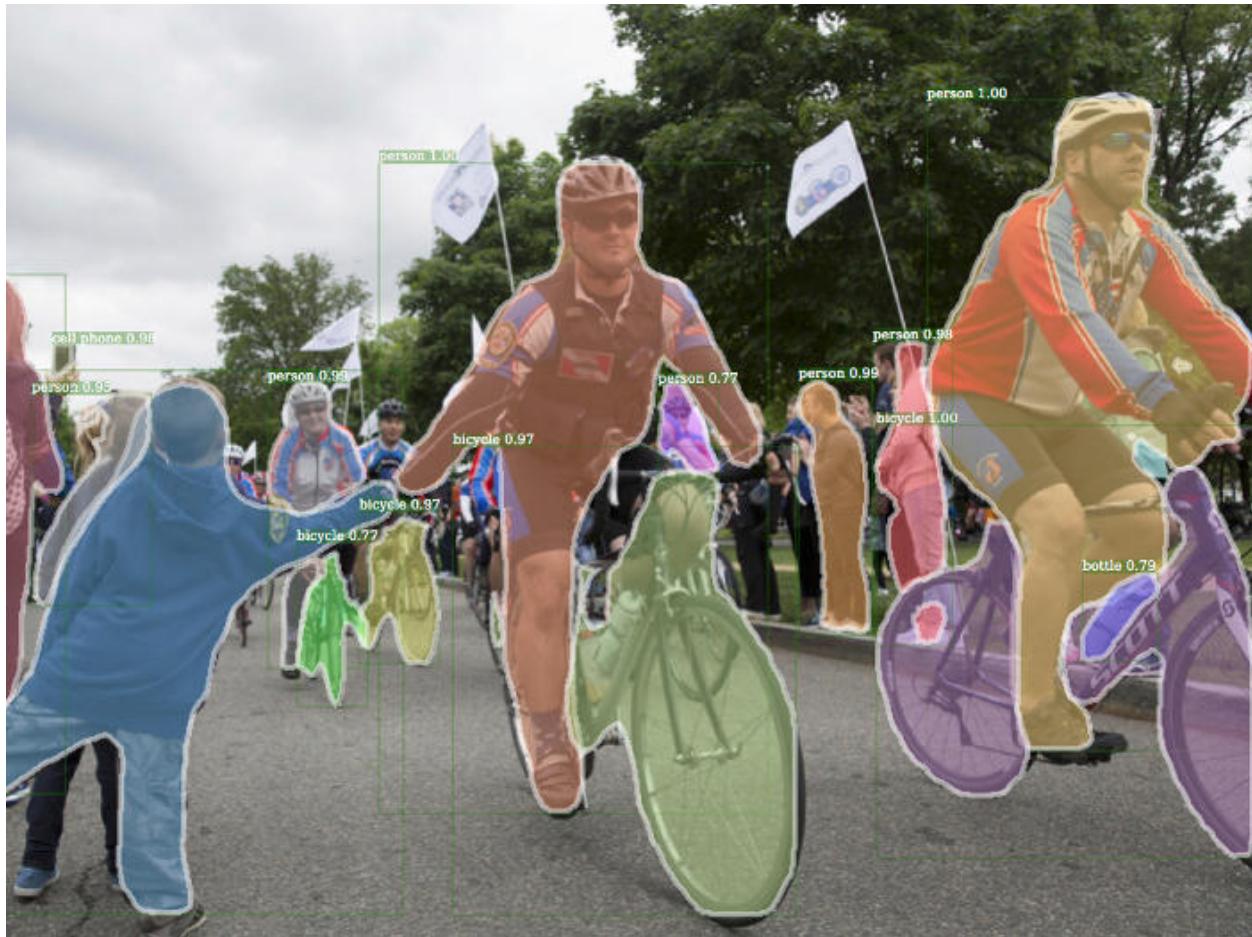


Figura 5. Análisis de imágenes con Computer Vision.

Fuente: (Veloso, 2020)

II.6 Teachable Machine

Conforme a lo expuesto por el equipo Google en la página oficial de teachablemachine.withgoogle.com, Teachable Machine es una herramienta basada en la web desarrollada por Google que facilita y acelera la creación de modelos de aprendizaje automático, permitiendo a cualquier persona experimentar con Inteligencia Artificial sin requerir conocimientos técnicos o de programación. Esta plataforma web accesible solo necesita un navegador, cámara y/o micrófono para su funcionamiento.

Con Teachable Machine, los usuarios pueden crear y entrenar modelos de manera rápida y sencilla, agrupando ejemplos en categorías y realizando pruebas instantáneas para comprobar la capacidad de clasificación. Además, los modelos pueden ser exportados y utilizados en diferentes proyectos, como sitios web y aplicaciones.

La creación de un modelo en esta herramienta, está basada en los siguientes pasos:

1. Reunir y agrupar ejemplos en clases o categorías.
2. Entrenar el modelo y probarlo directamente en la plataforma para ver si puede clasificar correctamente las categorías antes definidas.
3. Exportar el modelo, ya sea descargándolo o alojando en línea.

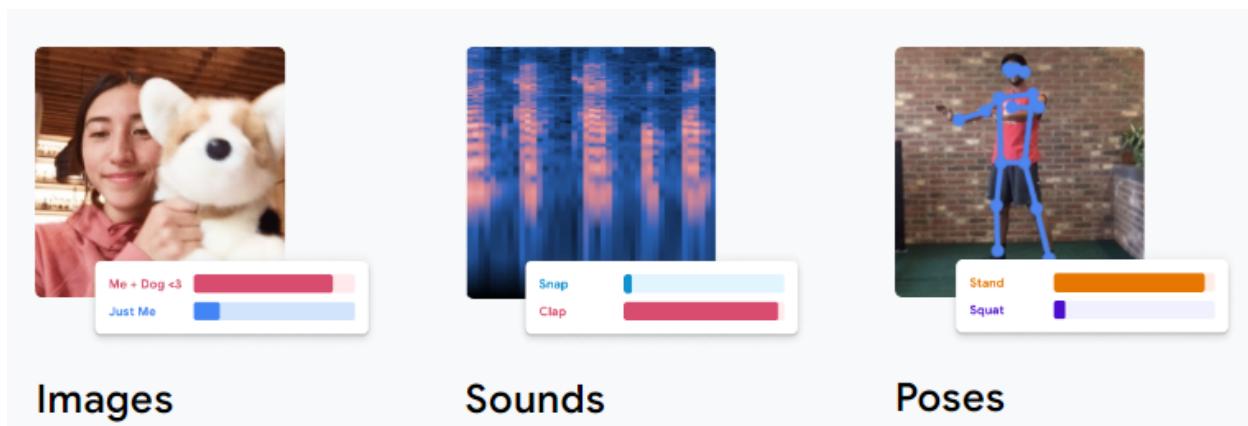


Figura 6. Tipo de modelos para proyecto en Teachable Machine.

Fuente: (teachablemachine.withgoogle.com, s/f)

Capítulo III. Marco Experimental

III.1 Tipo y alcance de la investigación

La investigación es un proceso sistemático que mediante la aplicación del método científico gestiona información relevante y lógica para entender, verificar, decidir, corregir o aplicar el conocimiento a partir de una serie de acciones lógicas. Esta genera procedimientos, presenta resultados y debe llegar a conclusiones. (Landau, 2001).

La finalidad del proyecto consiste en una investigación eminentemente aplicada, ya que busca dar solución a un problema práctico. Esta investigación tiene un alcance o carácter aplicativo que busca desarrollar y aplicar los conocimientos referentes a la traducción de los signos y gestos de la LSE a voz, enfocándose principalmente en el Alfabeto Dactilológico Español. Al mismo tiempo este es un proyecto del ámbito de la gestión del conocimiento, con el mérito de contribuir a convertir conocimiento habitualmente tácito en explícito.

A su vez el proyecto es una investigación con un alcance o carácter descriptivo. La investigación descriptiva consiste en la caracterización de un hecho, fenómeno, individuo o grupo, con el fin de establecer su estructura o comportamiento. (Fidias, 2012).

III.2 Método de desarrollo de software

III.2.1 Tecnologías a utilizar

En el desarrollo de este proyecto se emplearon diversas tecnologías, tanto en el entorno de desarrollo como en el funcionamiento del proyecto en sí.

Como principal lenguaje de programación, se utilizó Python, reconocido por su amplio uso en el desarrollo de aplicaciones web, software, ciencia de datos y aprendizaje automático (machine learning).

Para el desarrollo en Python, se empleó PyCharm, un entorno de desarrollo integrado (IDE) que brinda funcionalidades como análisis de código, depuración, pruebas unitarias y control de versiones.

Asimismo, se requirieron varias bibliotecas de Python, entre las cuales se incluyen OpenCV, cvZone, Numpy, Math, Mediapipe y TensorFlow. De estas bibliotecas, OpenCV y cvZone desempeñan un papel fundamental en el proyecto, ya que permiten el desarrollo de la visión computacional. En particular, se utilizaron los módulos HandTrackingModule y ClassificationModule para realizar el seguimiento de las manos en el video y clasificar las letras del alfabeto en lengua de señas española.

En cuanto al entrenamiento y creación del modelo, se usó Teachable Machine, una herramienta proporcionada por Google.

Por último, se utilizará el paquete de python Text-to-Speech para el procesamiento del lenguaje natural (NLP) y la conversión de texto a voz.

III.2.2 Fases del proyecto

El proyecto se estructura en cuatro etapas, las cuales se introducirán en este apartado y luego se explicarán en detalle.

En la primera etapa, se lleva a cabo la recopilación de muestras y la obtención de datos, específicamente imágenes. Estas imágenes son esenciales para el funcionamiento del modelo, por lo tanto, se ha desarrollado un código que permite la captura de imágenes de nuestras manos, a fin de realizar un muestreo adecuado y poder entrenar el modelo de manera efectiva.

En la segunda etapa, se procede a la creación de un modelo utilizando la biblioteca Keras, con el propósito de clasificar cada una de las letras del alfabeto dactilológico español. Este modelo será fundamental para el proceso de reconocimiento y traducción de los signos de la lengua de señas.

A continuación, se desarrolla un módulo de pruebas, el cual integra el modelo creado anteriormente y nos brinda la capacidad de realizar predicciones de los signos del Alfabeto Dactilológico Español. Esta etapa es crucial para evaluar el desempeño y la precisión del modelo implementado.

En la etapa final del proyecto, se realiza el procesamiento y la conversión de la traducción obtenida en el paso anterior a voz. Para ello, se emplean técnicas y herramientas de procesamiento del lenguaje natural, permitiendo así que el sistema pueda emitir la traducción en forma de audio, enriqueciendo la experiencia de comunicación con los usuarios.

III.2.2.1 Fase 1 - Recolección de datos

Para llevar a cabo el proceso de recolección de datos, hemos desarrollado un script en Python denominado "data_collector", el cual se apoya en las bibliotecas OpenCV y

cvZone. Estas herramientas nos permiten realizar la transmisión de video en tiempo real y rastrear la posición de nuestras manos mediante el módulo "HandTracking" incorporado en dichas bibliotecas.

El rastreo de manos nos proporciona una representación visual de las manos dentro de la imagen, marcando 21 puntos en la palma de cada mano. Esta información resulta fundamental para realizar predicciones precisas posteriormente.

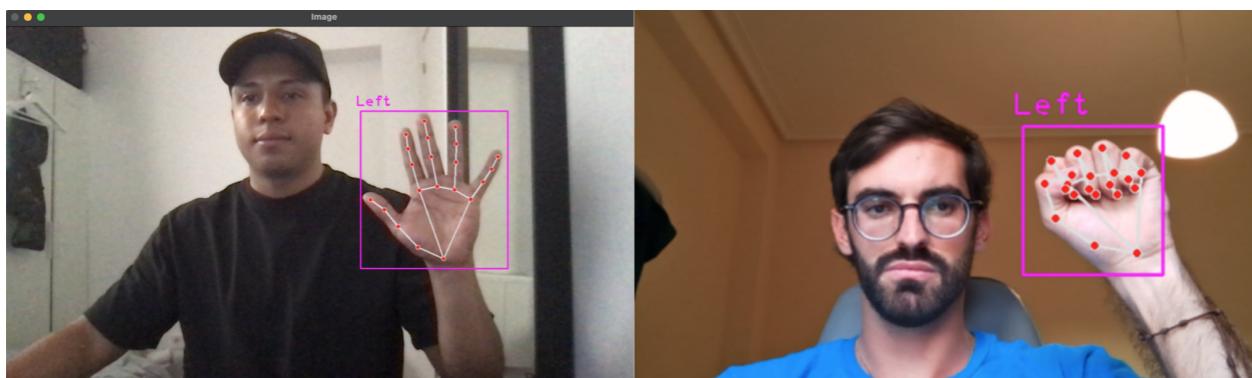


Figura 7. HandTracking en tiempo real.

Una vez que el script de recolección de datos está listo, procedemos a capturar las imágenes necesarias. Sin embargo, nos enfrentamos a un desafío: garantizar que todas las imágenes se obtengan de manera consistente, incluso al realizar diferentes tipos de señas que pueden afectar la posición relativa de la mano en el cuadro principal.

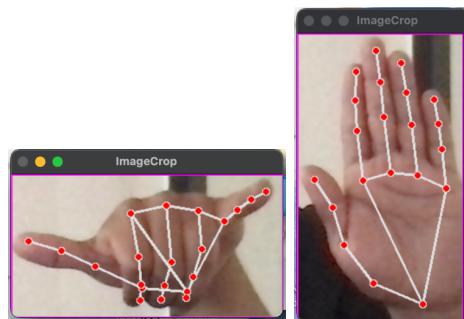


Figura 8. Imagen de la mano con dimensiones distintas.

Para abordar esta problemática, hemos implementado una técnica de redimensionamiento instantáneo de las imágenes, donde se agrega un cuadro blanco de fondo para estandarizar el tamaño de píxeles en todas nuestras capturas. De esta manera, aseguramos un muestreo homogéneo y coherente en nuestros datos.

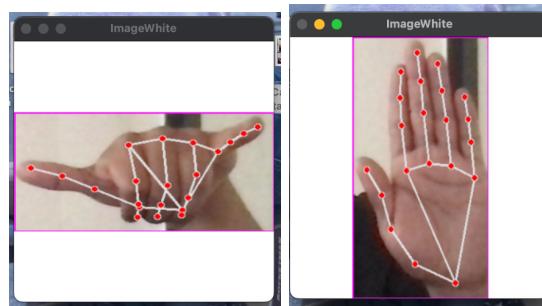


Figura 9. Imagen de la mano redimensionada.

Una vez resuelto este aspecto, se ha llevado a cabo la captura de alrededor de 6000 imágenes, involucrando a 6 personas diferentes en el proceso de recolección de datos. Este enfoque nos ha permitido obtener una muestra amplia y diversa para su posterior análisis y entrenamiento de nuestro modelo.

III.2.2.2 Fase 2 - Creación del modelo

Durante la etapa de creación del modelo en este proyecto, hemos empleado una herramienta altamente reconocida y eficaz conocida como Teachable Machine, desarrollada por Google. Esta herramienta, basada en la web, nos ha permitido aprovechar las capacidades de aprendizaje automático y la inteligencia artificial de una manera accesible y efectiva.

Teachable Machine nos ha brindado una interfaz intuitiva y amigable que facilita la tarea de entrenar nuestro modelo. Con esta herramienta, hemos podido reunir y

clasificar nuestras muestras de datos en diferentes categorías o clases, lo que resulta fundamental para que el modelo pueda aprender y reconocer patrones significativos.

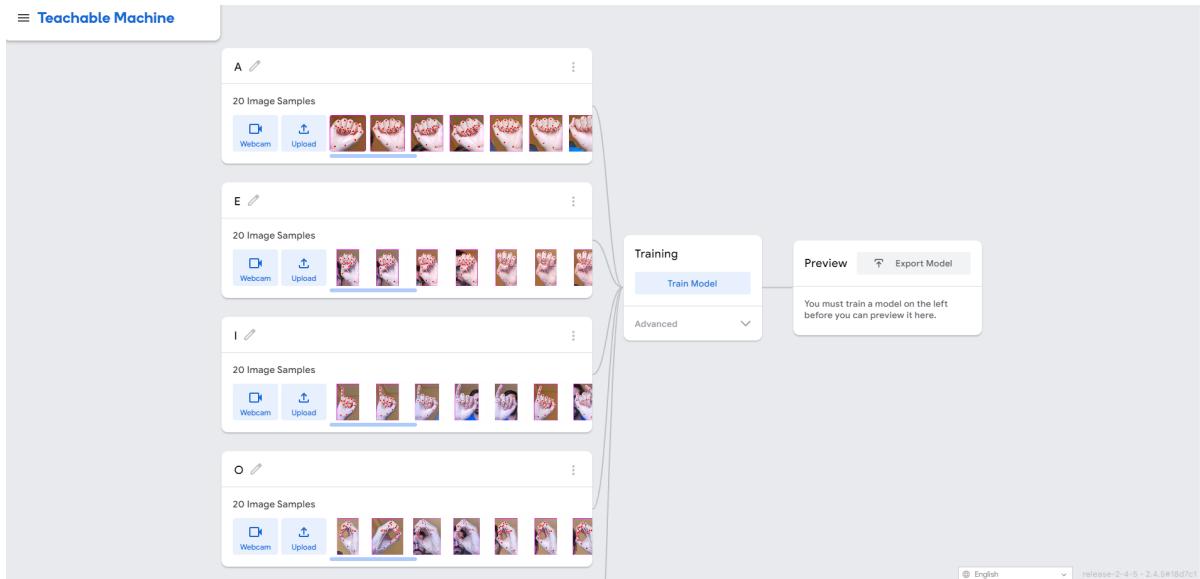


Figura 10. Visual de creación del primer modelo en Teachable Machine.

Para entrenar el modelo, lo ejecutamos con la configuración que nos recomendaba Teachable Machine por default: 50 epochs, un batch size de 16 y un learning rate de 0.001.

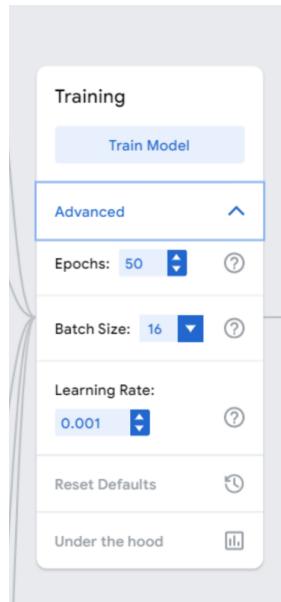


Figura 11. Configuración del modelo.

Una vez que hemos completado el proceso de entrenamiento, Teachable Machine nos ha permitido realizar pruebas instantáneas para evaluar la capacidad de nuestro modelo para clasificar correctamente. Esta capacidad de prueba nos ha permitido ajustar y optimizar nuestro modelo, asegurando su precisión y rendimiento.

Además, Teachable Machine nos ofrece opciones flexibles para exportar el modelo resultante, lo que nos ha permitido integrar fácilmente en nuestros proyectos.

III.2.2.3 Fase 3 - Testing

Durante esta fase, llevamos a cabo la prueba y usabilidad del modelo. Se realizó un script de python donde corremos el modelo mientras capturamos imagen en tiempo real para predecir cada una de las letras dentro del alfabeto dactilológico español.

Este script obtiene los índices que nos devuelve el modelo realizado, con cada uno de estos índices, encuentra la letra correspondiente y se va guardando en memoria para así poder concatenar las letras y escribir palabras.

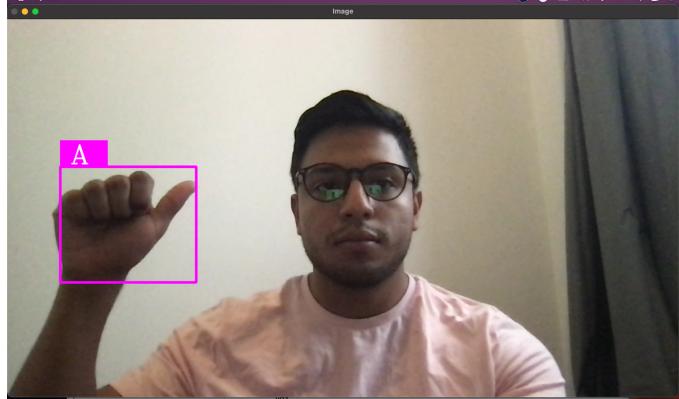


Figura 12. Predicción de letras

Uno de los retos más grandes dentro de esta fase fue al concatenar las letras. Nuestro modelo, al realizar predicciones en tiempo real nos regresa los índices de predicción de manera rápida y consistente y al ser video en streaming no podemos parar la predicción ya que el video se ralentiza, por lo cual tuvimos que guardar cada una de las letras obtenidas dentro de una lista en memoria y poner un temporizador para obtener la letra a predecir cada determinado tiempo, con esto, obtenemos la letra deseada y damos tiempo al usuario de cambiar de posición la mano para obtener la letra siguiente.

Figura 13. Concatenación de letras para formar palabras

III.2.2.4 Fase 4 - Procesamiento de predicción a voz

En esta última etapa del proyecto, se llevó a cabo la implementación de un proceso de conversión de texto a voz utilizando una técnica conocida como Procesamiento del Lenguaje Natural (NLP, por sus siglas en inglés). Para lograr esto, se integró la librería Python Text to Speech (pyttsx3). En esta fase, el texto que fue previamente generado y concatenado durante la etapa anterior, se utiliza como *input* para la función encargada de realizar la conversión del texto a una representación audible mediante síntesis de voz.

Conclusiones

La realización de esta investigación comenzó con el aprendizaje de la estructura y vocabulario de la lengua de signos española, haciendo énfasis en el alfabeto dactilológico español.

Posteriormente, se avanzó hacia el desarrollo del algoritmo de detección de los gestos antes mencionados. Cada fase elaborada contribuyó a ofrecer elementos claves para un mejor desempeño en la fase siguiente, a continuación detallamos cada una de ellas.

La primera fase de recolección de datos ha sido crucial para obtener una muestra diversa y representativa que nos permitió entrenar nuestro modelo con un conjunto completo de imágenes de las manos en diferentes posiciones y gestos. La implementación del script "data_collector" junto con las bibliotecas OpenCV y cvZone nos brindó las herramientas necesarias para capturar y estandarizar las imágenes de manera eficiente, asegurando la consistencia en nuestro conjunto de datos.

En la fase de creación del modelo, la utilización de la herramienta Teachable Machine fue fundamental para agilizar y simplificar el proceso de entrenamiento. La interfaz intuitiva y las capacidades de aprendizaje automático incorporadas nos permitió clasificar y entrenar nuestro modelo de manera efectiva. Gracias a su facilidad de uso, potencia y versatilidad, hemos logrado desarrollar un modelo robusto y preciso que respalda los objetivos y resultados esperados en este proyecto de investigación.

La fase de testing fue parte fundamental para nuestros resultados y objetivos esperados, ya que al realizar la predicción y concatenar la letras de forma correcta para formar palabras nos da un producto finalizado y utilizable. Los retos dentro de esta etapa nos permitieron agilizar el modelo y a su vez hacer un script automático para la creación de palabras.

En la última fase del proyecto, el procesamiento de predicción a voz, la integración de la librería pytsxs3 ha sido clave para convertir el texto predicho y concatenado en la etapa anterior en una salida de audio de voz humana. Esta capacidad de síntesis de voz ha agregado una dimensión auditiva y natural a nuestro proyecto, brindando una experiencia más inmersiva y accesible para los usuarios.

En conclusión, el período de aprendizaje y el conjunto de las cuatro fases del proyecto han permitido la construcción de un sistema completo y funcional que utiliza técnicas de visión por computadora, aprendizaje automático y procesamiento de lenguaje natural para interpretar y comunicar la lengua de signos a través de la voz. Estas etapas han demostrado ser fundamentales para lograr un modelo preciso y eficiente, proporcionando un paso importante hacia la creación de soluciones tecnológicas inclusivas y accesibles en el campo de la comunicación con y entre personas con discapacidades auditivas, visuales y/o dificultades en el habla.

Recomendaciones

El presente trabajo ha logrado desarrollar un sistema de detección y clasificación de signos y gestos en la lengua de signos española mediante el uso de técnicas de visión por computadora y aprendizaje automático. Aunque hemos obtenido resultados prometedores en la detección y clasificación precisa de los signos, es importante destacar que el reconocimiento del movimiento aún se encuentra en una fase inicial de exploración y desarrollo.

Por último, este trabajo representa un paso importante hacia la creación de soluciones tecnológicas que faciliten la comunicación y promuevan la inclusión de las personas con discapacidades auditivas, visuales y/o dificultades en el habla. A medida que continuamos investigando y desarrollando en el campo de la inteligencia artificial, esperamos abordar las limitaciones identificadas y seguir avanzando hacia un sistema más completo y preciso que pueda reconocer tanto los signos como los movimientos en la lengua de signos, mejorando así la calidad de vida de las personas.

Referencias bibliográficas

Objetivo 10: Reducir la desigualdad en y entre los países. (2020, December 10).

Organización de las Naciones Unidas.

<https://www.un.org/sustainabledevelopment/es/inequality/>

Definición de LSE. (2019, February 18). Federación de Personas Sordas de la Comunidad de MadridFeSorCam. <https://www.fesorciam.org/definicion-de-lse/>

¿Qué es la lengua de signos española?. (2018, June 19). IELSE.

<https://ielse.es/lengua-de-signos-espanola/>

Lengua de Signos (no date). Sociedad Federada de Personas Sordas de Málaga SFSM. Available at: <https://sfsm.es/lengua-de-signos/>

Banco de imágenes y signos, Dactilológico. (no date). Fundación ONCE. Available at: <https://www.fundacioncnse.org/educa/bancolse/dactilogico.php#gsc.tab=0>

¿Qué es la inteligencia artificial o ia? | google cloud (no date) Google. Available at: <https://cloud.google.com/learn/what-is-artificial-intelligence?hl=es-419>

Lahtela, M. and Kaplan, P. (Provenance) (1966) ES, Amazon. Available at: <https://aws.amazon.com/es/what-is/python/>

PyCharm: The python IDE for professional developers by jetbrains (no date) JetBrains. Available at: <https://www.jetbrains.com/es-es/lp/pycharm-anaconda/> (Accessed: 17 May 2023).

El paquete Numpy - Fundamentos de Programación en Python. (no date). Escuela de Ingenierías Industriales de la Universidad de Valladolid. Available at: https://www2.eii.uva.es/fund_inf/python/notebooks/Bibliotecas/03_Numpy/Numpy.html

Marín, R. (2020, February 12) *OpenCV, instalación en python y Ejemplos Básicos*, *Canal Informática y TICS*. INESEM Business School. Available at: <https://www.inesem.es/revistadigital/informatica-y-tics/opencv/>

Kiruri, S. and Dawe, D. (2022) *Creating a hand tracking module using python, opencv, and MediaPipe*, *Section*. Available at: <https://www.section.io/engineering-education/creating-a-hand-tracking-module/>

Tensorflow (2023) *TensorFlow*. Available at: <https://www.tensorflow.org/>

What is Computer Vision? (2021) IBM. Available at: <https://www.ibm.com/topics/computer-vision>

Veloso, F. (2020) *Computer Vision, que es y un ejemplo, Feeding The Machine*. Available at: <https://www.feedingthemachine.ai/computer-vision/>

Teachable Machine. (No date). Google. Available at: <https://teachablemachine.withgoogle.com/>

Ríos, P. (2017). Metodología de la investigación: un enfoque pedagógico. Caracas: Cognitus.

Keras: Biblioteca de Código Abierto Para Crear redes neuronales (2020) IONOS Digital Guide. Available at: <https://www.ionos.es/digitalguide/online-marketing/marketing-para-motores-de-busqueda/que-es-keras/>

Rockikz, A. (2020) *How to convert text to speech in python, Python Code*. Available at: <https://www.thepythontutorials.com/article/convert-text-to-speech-in-python>