**Sistema de comunicación inclusiva**

*“Detección de manos y traducción para el lenguaje de señas”*

Trabajo final.

Brenda Antonella Avalos (MAT: 49010823)

Universidad de Morón.

Escuela Superior de Ingeniería, Informática y Ciencias Agroalimentarias.

Cursada: Trabajo de campo (COD: 3334)

Profesores titulares: Mara Alejandra Capuya, Luis Alberto Amilivia.

Control de versiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Fecha** | **Adiciones/Modificaciones** | **Modificado por** |
| 1.0 | 30/3/2023 | Presentación funcional | Brenda Avalos |
| ~~1.1~~ | ~~13/4/2023~~ | ~~Planificación.~~ | ~~Brenda Avalos~~ |
| ~~1.2~~ | ~~20/4/2023~~ | ~~Diagrama de componentes~~ | ~~Brenda Avalos~~ |
| 1.3 | 27/4/2023 | Detalle de configuración previa | Brenda Avalos |
| 1.4 | 22/6/2023 | Desarrollo del sistema | Brenda Avalos |
| 1.5 | 27/6/2023 | Conclusión | Brenda Avalos |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Tabla de contenidos:**

[Abstract 4](#_Toc143777652)

[Introducción 5](#_Toc143777653)

[Lenguaje de señas 6](#_Toc143777654)

[Reconocimiento facial 7](#_Toc143777655)

[Planteamiento del problema 8](#_Toc143777656)

[Propuesta 9](#_Toc143777657)

[Configuración previa 10](#_Toc143777658)

[Estructura del sistema 16](#_Toc143777659)

[Conclusión 24](#_Toc143777660)

Abstract

En este trabajo de investigación se indagará en el desarrollo de un sistema de reconocimiento facial para el lenguaje de señas, con el objetivo de mejorar la comunicación de las personas con discapacidad auditiva.   
El sistema utilizará técnicas de procesamiento de imágenes y aprendizaje automático para identificar y traducir los gestos del lenguaje de señas en tiempo real. Se analizarán diversas tecnologías y técnicas de aprendizaje automático para determinar las más adecuadas para la detección y reconocimiento de gestos. Además, se explorarán diferentes métodos para la síntesis de voz o la traducción automática para convertir los gestos en palabras o frases comprensibles para la persona con la que se está comunicando.   
El resultado final será un sistema capaz de reconocer y traducir el lenguaje de señas en tiempo real, mejorando la inclusión social y la comunicación de las personas con discapacidad auditiva en diferentes ámbitos de la vida.

***Palabras claves:*** *reconocimiento facial, lenguaje de señas, aprendizaje automático.*

Introducción

Proyecto de reconocimiento facial para lenguaje de señas tiene como objetivo desarrollar un sistema capaz de detectar los movimientos de las manos y traducir el lenguaje de señas a lenguaje oral o escrito. Este sistema utiliza técnicas de procesamiento de imágenes y aprendizaje automático para identificar los patrones de los gestos realizados por una persona sorda o con discapacidad auditiva y convertirlos en palabras o frases en tiempo real.

Para desarrollar un proyecto de reconocimiento facial para lenguaje de señas, es necesario contar con una base de datos de imágenes que contengan ejemplos de los distintos gestos del lenguaje de señas, lo que permitirá entrenar un modelo de aprendizaje automático capaz de identificar y reconocer estos gestos. También es necesario contar con una cámara o un dispositivo de captura de video para capturar los gestos realizados por la persona sorda o con discapacidad auditiva en tiempo real.

Una vez entrenado el modelo de reconocimiento facial, el sistema puede ser utilizado para interpretar los gestos y convertirlos en texto o voz. Para ello, se pueden utilizar tecnologías de síntesis de voz o de traducción automática para convertir los gestos en palabras o frases comprensibles para la persona con la que se está comunicando.

Un proyecto de reconocimiento facial para lenguaje de señas puede tener una gran aplicación en la vida cotidiana de las personas sordas o con discapacidad auditiva, permitiéndoles comunicarse más fácilmente con personas que no conocen el lenguaje de señas y mejorar su inclusión social.

Lenguaje de señas

El lenguaje de señas, también conocido como lengua de señas, es una forma de comunicación visual y gestual utilizada por personas sordas o con discapacidad auditiva. Es un lenguaje completamente distinto al habla verbal y tiene su propia gramática y vocabulario.

En el lenguaje de señas, los signos se realizan con las manos, los dedos, los brazos y la cara. Cada signo tiene un significado propio y puede representar una palabra, una frase o una idea completa. La velocidad y la fluidez de los signos son importantes para la comprensión del mensaje.

El lenguaje de señas es utilizado en todo el mundo, aunque hay diferentes sistemas de signos para cada país o región. Por ejemplo, la lengua de señas española es distinta de la lengua de señas estadounidense o de la lengua de señas japonesa.

Además de las señas, el lenguaje de señas también utiliza expresiones faciales y corporales para transmitir emociones y entonaciones. Estos gestos pueden modificar el significado de los signos y añadir matices al mensaje.

El lenguaje de señas es una forma de comunicación importante para las personas sordas o con discapacidad auditiva, ya que les permite comunicarse entre sí y con personas que no conocen el lenguaje de señas. Además, muchas personas que pueden oír también aprenden el lenguaje de señas para poder comunicarse con personas sordas o con discapacidad auditiva y fomentar la inclusión y la diversidad.

Reconocimiento facial

El reconocimiento facial es una tecnología de identificación que utiliza algoritmos para analizar y comparar rasgos faciales únicos con el fin de identificar y autenticar a una persona.   
Es una herramienta que suele ser utilizada en la seguridad y en la tecnología de la información, así como en aplicaciones más cotidianas como la identificación de caras en fotos o el desbloqueo de dispositivos móviles.

El proceso de reconocimiento facial comienza con la captura de una imagen o un video del rostro de una persona. La imagen se somete a un análisis para identificar rasgos faciales únicos como la distancia entre los ojos, la forma de la nariz, la posición de las orejas y la boca, entre otros. Estos rasgos se convierten en datos digitales que se comparan con una base de datos de imágenes de rostros previamente almacenados.

El sistema de reconocimiento facial utiliza algoritmos avanzados de aprendizaje automático para comparar la imagen capturada con la base de datos de rostros previamente almacenados. El algoritmo compara los rasgos faciales de la imagen capturada con los rasgos faciales de las imágenes almacenadas previamente en la base de datos y devuelve una puntuación de similitud o una lista de posibles coincidencias.

Planteamiento del problema

El lenguaje de señas es utilizado por personas sordas o con discapacidad auditiva como medio de comunicación y lenguaje. Sin embargo, a pesar de su importancia, el acceso a la educación y a los servicios públicos para estas personas sigue siendo limitado en muchas partes del mundo.

Uno de los principales problemas asociados con el lenguaje de señas es la falta de reconocimiento y aceptación de este como un lenguaje completo y autónomo. Esto ha llevado a la falta de recursos y herramientas para la enseñanza y aprendizaje del lenguaje de señas, lo que a su vez limita el acceso a la información y la comunicación para las personas sordas o con discapacidad auditiva.

Además, la falta de estandarización del lenguaje de señas entre los distintos países y regiones del mundo hace que la comunicación entre personas sordas de diferentes lugares pueda ser difícil o incluso imposible. Esto puede tener consecuencias graves en situaciones de emergencia o en contextos en los que la comunicación es crucial.

Por lo tanto, el problema del lenguaje de señas se centra en la necesidad de promover su reconocimiento como un lenguaje autónomo y completo, así como en la necesidad de desarrollar recursos y herramientas que permitan su enseñanza y aprendizaje de manera efectiva y accesible para las personas sordas o con discapacidad auditiva.

.

Propuesta

La propuesta para abordar el problema del lenguaje de señas es el desarrollo de una tecnología de reconocimiento facial que permita la interpretación en tiempo real de los gestos y expresiones utilizados en el lenguaje de señas.

Esta tecnología podría utilizarse para crear una aplicación que permita la comunicación entre personas sordas y oyentes, en la que las personas sordas pudieran expresarse en lenguaje de señas y la tecnología la tradujera automáticamente al lenguaje oral para que los oyentes pudieran entenderla. Del mismo modo, los oyentes podrían hablar en su idioma oral y la tecnología los traduciría a lenguaje de señas para que las personas sordas pudieran entenderlos.

Además, esta tecnología podría ser útil para la enseñanza y aprendizaje del lenguaje de señas, ya que podría proporcionar una retroalimentación en tiempo real sobre la corrección de los gestos y expresiones realizados por los usuarios.

Para desarrollar esta tecnología sería necesario un amplio conjunto de datos de gestos y expresiones utilizados en el lenguaje de señas, así como algoritmos de aprendizaje automático para el reconocimiento facial y la traducción del lenguaje de señas a lenguaje oral y viceversa.

Si bien es cierto que esta propuesta no aborda directamente el problema de la falta de reconocimiento del lenguaje de señas como un lenguaje completo y autónomo, podría ser una herramienta útil para mejorar la comunicación y el acceso a la información y la educación para las personas sordas o con discapacidad auditiva.

Configuración previa

**Lenguaje de programación:** Python 3.8.

**Entorno virtual** : Anaconda.

Al utilizar el entorno Anaconda, se pueden administrar y crear fácilmente entornos virtuales que contienen diferentes versiones de Python y las bibliotecas necesarias para proyectos específicos. Esto permite tener entornos aislados y personalizados para diferentes proyectos sin conflictos entre paquetes y dependencias.

**Editor de texto:** Jupyter notebook.

**Lista de paquetes configurados + versiones:**

|  |
| --- |
| Package Version |
| absl-py 1.4.0 |
| anyio 3.5.0 |
| apache-beam 2.46.0 |
| argon2-cffi 21.3.0 |
| argon2-cffi-bindings 21.2.0 |
| array-record 0.2.0 |
| asttokens 2.0.5 |
| astunparse 1.6.3 |
| attrs 22.1.0 |
| avro-python3 1.10.2 |
| backcall 0.2.0 |
| beautifulsoup4 4.12.2 |
| bidict 0.22.1 |
| bleach 4.1.0 |
| blinker 1.6.2 |
| cachetools 5.3.1 |
| camera 1.3.0 |
| certifi 2023.5.7 |
| cffi 1.15.1 |
| charset-normalizer 3.1.0 |
| clang 5.0 |
| click 8.1.3 |
| cloudpickle 2.2.1 |
| colorama 0.4.6 |
| comm 0.1.2 |
| contextlib2 21.6.0 |
| contourpy 1.0.7 |
| crcmod 1.7 |
| cycler 0.11.0 |
| Cython 0.29.35 |
| debugpy 1.5.1 |
| decorator 5.1.1 |
| defusedxml 0.7.1 |
| dill 0.3.1.1 |
| dm-tree 0.1.8 |
| docopt 0.6.2 |
| entrypoints 0.4 |
| etils 1.3.0 |
| executing 0.8.3 |
| fastavro 1.7.4 |
| fasteners 0.18 |
| fastjsonschema 2.16.2 |
| Flask 2.3.2 |
| Flask-SocketIO 5.3.4 |
| flatbuffers 1.12 |
| fonttools 4.39.4 |
| gast 0.4.0 |
| gevent 22.10.2 |
| gevent-websocket 0.10.1 |
| gin-config 0.5.0 |
| google-api-core 2.11.0 |
| google-api-python-client 2.88.0 |
| google-auth 2.19.1 |
| google-auth-httplib2 0.1.0 |
| google-auth-oauthlib 0.4.6 |
| google-cloud-speech 2.20.0 |
| google-pasta 0.2.0 |
| googleapis-common-protos 1.59.0 |
| greenlet 2.0.2 |
| grpcio 1.56.0 |
| grpcio-status 1.56.0 |
| h5py 3.1.0 |
| hdfs 2.7.0 |
| httplib2 0.21.0 |
| idna 3.4 |
| immutabledict 2.2.4 |
| importlib-metadata 6.6.0 |
| importlib-resources 5.2.0 |
| ipykernel 6.19.2 |
| ipython 8.12.0 |
| ipython-genutils 0.2.0 |
| itsdangerous 2.1.2 |
| jax 0.4.11 |
| jedi 0.18.1 |
| Jinja2 3.1.2 |
| joblib 1.2.0 |
| jsonschema 4.17.3 |
| jupyter\_client 8.1.0 |
| jupyter\_core 5.3.0 |
| jupyter-server 1.23.4 |
| jupyterlab-pygments 0.1.2 |
| kaggle 1.5.13 |
| keras 2.9.0 |
| keras-nightly 2.5.0.dev2021032900 |
| Keras-Preprocessing 1.1.2 |
| kiwisolver 1.4.4 |
| libclang 16.0.0 |
| lvis 0.5.3 |
| lxml 4.9.2 |
| Markdown 3.4.3 |
| MarkupSafe 2.1.3 |
| matplotlib 3.7.1 |
| matplotlib-inline 0.1.6 |
| mistune 0.8.4 |
| mkl-fft 1.3.6 |
| mkl-random 1.2.2 |
| mkl-service 2.4.0 |
| ml-dtypes 0.1.0 |
| nbclassic 0.5.5 |
| nbclient 0.5.13 |
| nbconvert 6.5.4 |
| nbformat 5.7.0 |
| nest-asyncio 1.5.6 |
| notebook 6.5.4 |
| notebook\_shim 0.2.2 |
| numpy 1.21.0 |
| oauth2client 4.1.3 |
| oauthlib 3.2.2 |
| object-detection 0.1 |
| objsize 0.6.1 |
| opencv-python 4.7.0.72 |
| opencv-python-headless 4.7.0.72 |
| opt-einsum 3.3.0 |
| orjson 3.9.0 |
| packaging 23.0 |
| pandas 2.0.2 |
| pandocfilters 1.5.0 |
| parso 0.8.3 |
| pickleshare 0.7.5 |
| Pillow 9.5.0 |
| pip 23.1.2 |
| pkgutil\_resolve\_name 1.3.10 |
| platformdirs 2.5.2 |
| portalocker 2.7.0 |
| prometheus-client 0.14.1 |
| promise 2.3 |
| prompt-toolkit 3.0.36 |
| proto-plus 1.22.2 |
| protobuf 3.19.4 |
| psutil 5.9.0 |
| pure-eval 0.2.2 |
| py-cpuinfo 9.0.0 |
| pyarrow 9.0.0 |
| pyasn1 0.5.0 |
| pyasn1-modules 0.3.0 |
| PyAudio 0.2.13 |
| pycocotools 2.0 |
| pycparser 2.21 |
| pydot 1.4.2 |
| Pygments 2.15.1 |
| pymongo 3.13.0 |
| pyparsing 2.4.7 |
| PyQt5 5.15.9 |
| PyQt5-Qt5 5.15.2 |
| PyQt5-sip 12.12.1 |
| pyrsistent 0.18.0 |
| python-dateutil 2.8.2 |
| python-engineio 4.4.1 |
| python-slugify 8.0.1 |
| python-socketio 5.8.0 |
| pytz 2023.3 |
| pywin32 305.1 |
| pywinpty 2.0.10 |
| PyYAML 5.4.1 |
| pyzmq 25.0.2 |
| regex 2023.6.3 |
| requests 2.31.0 |
| requests-oauthlib 1.3.1 |
| rsa 4.9 |
| sacrebleu 2.2.0 |
| scikit-learn 1.2.2 |
| scipy 1.10.1 |
| Send2Trash 1.8.0 |
| sentencepiece 0.1.99 |
| seqeval 1.2.2 |
| setuptools 67.8.0 |
| six 1.15.0 |
| slim 0.1 c:\users\bavalos\desktop\proyecto\tfodcourse\tensorflow\models\research\slim |
| sniffio 1.2.0 |
| soupsieve 2.4 |
| SpeechRecognition 3.10.0 |
| stack-data 0.2.0 |
| tabulate 0.9.0 |
| tensorboard 2.9.1 |
| tensorboard-data-server 0.7.0 |
| tensorboard-plugin-wit 1.8.1 |
| tensorflow 2.9.0 |
| tensorflow-addons 0.20.0 |
| tensorflow-datasets 4.9.0 |
| tensorflow-estimator 2.9.0 |
| tensorflow-hub 0.13.0 |
| tensorflow-intel 2.12.0 |
| tensorflow-io 0.31.0 |
| tensorflow-io-gcs-filesystem 0.31.0 |
| tensorflow-metadata 1.13.0 |
| tensorflow-model-optimization 0.7.5 |
| tensorflow-text 2.10.0 |
| termcolor 1.1.0 |
| terminado 0.17.1 |
| text-unidecode 1.3 |
| tf-models-official 2.10.1 |
| tf-slim 1.1.0 |
| threadpoolctl 3.1.0 |
| tinycss2 1.2.1 |
| toml 0.10.2 |
| tornado 6.2 |
| tqdm 4.65.0 |
| traitlets 5.7.1 |
| typeguard 2.13.3 |
| typing-extensions 3.7.4.3 |
| tzdata 2023.3 |
| uritemplate 4.1.1 |
| urllib3 1.26.16 |
| waitress 2.1.2 |
| wcwidth 0.2.5 |
| webencodings 0.5.1 |
| websocket-client 0.58.0 |
| Werkzeug 2.3.4 |
| wget 3.2 |
| wheel 0.40.0 |
| wrapt 1.12.1 |
| zipp 3.15.0 |
| zope.event 5.0 |
| zope.interface 6.0 |
| zstandard 0.21.0 |

Estructura del sistema

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente con confianza media

**Desarrollo del sistema**

* 1. **Collección de imágenes - Data set.ipynb**
  2. Definición flexible de gestos. Definición de la cantidad de fotos a capturar.
  3. Se crean automáticamente carpetas para cada uno de los gestos predefinidos.

TFODCourse\Tensorflow\workspace\images\collectedimages

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* 1. Se realiza la captura de las imágenes y se coloca en su respectiva carpeta.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* 1. Interfaz de usuario gráfica

     Descripción generada automáticamenteSe realiza el etiquetado de imágenes bajo el repositorio que el sistema de etiquetado. [*https://github.com/tzutalin/labelImg*](https://github.com/tzutalin/labelImg)
  2. Interfaz de usuario gráfica, Aplicación

     Descripción generada automáticamenteSe seleccionan imágenes para el entrenamiento y prueba de este.  
     C:\Users\BAVALOS\Desktop\Proyecto\TFODCourse\Tensorflow\workspace\images
  3. **Detección y entrenamiento.ipynb**

1. Descarga de Modelos entrenados de TF desde "Tensorflow Model Zoo" e intalación especifica de TFOD (detector de objetos)  
   [*http://download.tensorflow.org/models/object\_detection/tf2/20200711/ssd\_mobilenet\_v2\_fpnlite\_320x320\_coco17\_tpu-8.tar.gz*](http://download.tensorflow.org/models/object_detection/tf2/20200711/ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8.tar.gz)*'*
2. Generación de mapa de etiquetado. (Identificador)  
   C:\Users\BAVALOS\Desktop\Proyecto\TFODCourse\Tensorflow\workspace\annotations  
   Interfaz de usuario gráfica, Texto, Aplicación

   Descripción generada automáticamente
3. Se convierte las anotaciones de las imágenes en un formato de archivo que se pueda utilizar por TF.

*https://github.com/nicknochnack/GenerateTFRecord*

#Successfully created the TFRecord file: Tensorflow\workspace\annotations\test.record

#Successfully created the TFRecord file: Tensorflow\workspace\annotations\train.record

1. Se ejecuta el entrenamiento del modelo. Se valida el entrenamiento exitoso.

C:\Users\BAVALOS\Desktop\Proyecto\TFODCourse\Tensorflow\workspace\models\my\_ssd\_mobnet

Tabla

Descripción generada automáticamente

1. Se evalúa el modelo entrenado.

Descarga de tensorBoard para el análisis de métricas y precisión del modelo.

*TensorBoard es una herramienta de visualización y seguimiento de TensorFlow. Permite visualizar y analizar de manera interactiva los modelos de TensorFlow, incluyendo gráficos computacionales, métricas de rendimiento, distribución de datos, y más. TensorBoard facilita la comprensión y el análisis de los modelos de TensorFlow, lo que ayuda en el proceso de desarrollo, ajuste y optimización de estos. Además, proporciona una interfaz intuitiva para explorar y compartir los resultados del entrenamiento de los modelos de TensorFlow*.

*C:\Users\BAVALOS\Desktop\Proyecto\TFODCourse\Tensorflow\workspace\models\my\_ssd\_mobnet\train>tensorboard --logdir=.*

[**http://localhost:6006/**](http://localhost:6006/)

Interfaz de usuario gráfica, Gráfico, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

1. Se realizan pruebas manuales.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente Pantalla de celular con imagen de hombre

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* 1. **Deteccion de voz.ipynb**

Se implemento la capacidad de detectar la voz a través de la cámara web. Sin embargo, debido a la exigencia, en términos de recursos, de la computadora, se configuro la detección de voz para que funcione a través del micrófono utilizando una página web. Esto permite la reducción de la carga de procesamiento, ya que la detección de voz se realiza en un entorno remoto, en lugar de en la propia máquina.

* 1. **Exportación.ipynb**

Se valida el almacenamiento y la exportación del modelo.

*C:\Users\BAVALOS\Desktop\Proyecto\TFODCourse\Tensorflow\workspace\models\my\_ssd\_mobnet*

* 1. **Presentación.ipynb**

Se levanta una página web para la detección de voz y la muestra de detección de objetos.

Imagen que contiene Texto

Descripción generada automáticamente

Exportación de modelo:

C:\Users\BAVALOS\Desktop\Proyecto\TFODCourse\Tensorflow\workspace\models\my\_ssd\_mobnet\export

Una captura de pantalla de una red social

Descripción generada automáticamente

Conclusión

Si bien es cierto que esta propuesta no aborda directamente el problema de la falta de reconocimiento del lenguaje de señas como un lenguaje completo y autónomo, ofrece una solución prometedora para mejorar la comunicación y el acceso a la información y la educación de las personas sordas o con discapacidad auditiva.

Estoy convencida de que esta aplicación puede marcar una diferencia significativa en la vida de las personas con discapacidad auditiva al proporcionar una forma más inclusiva y efectiva de comunicación en tiempo real. Su implementación podría abrir nuevas oportunidades y superar barreras, impulsando un entorno más inclusivo y equitativo para todos.