

Reconocimiento de señas de la Lengua de Señas Mexicana mediante técnicas de Machine Learning

Recognition of signs from Mexican Sign Language using Machine Learning techniques

Gabriel A. Salgado-Martínez^a, René E. Cuevas-Valencia^b, Angelino Feliciano-Morales^c, Arnulfo Catalán-Villegas^d

Abstract:

The use of Artificial Intelligence techniques for the recognition of signs of the Mexican Sign Language through can contribute to improve communication between people who speak this language and those who do not know its meaning. In this article, the performance of Machine Learning models in the recognition of signs was evaluated, for this purpose, four different techniques were used to train and later test the recognition in new signs. Firstly, a data set of images of the dactylological alphabet of the Mexican Sign Language was collected, then this data set was used to train sign image recognition models using four different techniques and later the performance of each model was analyzed in new sign pictures.

Keywords:

Mexican Sign Language, Machine Learning, Translation, Artificial Neural Networks.

Resumen:

El empleo de técnicas de Inteligencia Artificial para el reconocimiento de señas de la Lengua de Señas Mexicana puede contribuir a mejorar la comunicación entre personas hablantes en esta lengua y quienes desconocen su significado. En este artículo se evaluó el rendimiento de modelos de Machine Learning en el reconocimiento de señas, para ello se utilizaron cuatro técnicas diferentes para entrenar y posteriormente probar el reconocimiento en nuevas señas. Primeramente, se recolectó un set de datos de imágenes del alfabeto dactilológico de la Lengua de Señas Mexicana, luego este set de datos se utilizó para entrenar modelos de reconocimiento de imágenes de señas utilizando cuatro técnicas diferentes y posteriormente se analizó el rendimiento de cada modelo en nuevas imágenes de señas.

Palabras Clave:

Lengua de Señas Mexicana, Machine Learning, Reconocimiento, Redes Neuronales Artificiales.

Introducción

En México la población de personas sordas es de aproximadamente 1,350,000 ^[1], de este grupo de

personas aproximadamente 250,000 utilizan la Lengua de Señas Mexicana (LSM) para comunicarse ^[2].

En la sociedad mexicana moderna, las personas que utilizan la LSM se encuentran con diversos desafíos que afectan su calidad de vida y bienestar en general. Uno de

^a Autor de Correspondencia, <https://orcid.org/0009-0004-8614-5469>, Universidad Autónoma de Guerrero, <https://www.uagro.mx/>, Email: 06316477@uagro.mx

^b Universidad Autónoma de Guerrero, <https://orcid.org/0000-0001-9528-7603>, <https://www.uagro.mx/>, Email: reneecuevas@uagro.mx

^c Universidad Autónoma de Guerrero, <https://orcid.org/0000-0002-7707-7319>, <https://www.uagro.mx/>, Email: afmorales@uagro.mx

^d Universidad Autónoma de Guerrero, <https://orcid.org/0009-0001-0391-7960>, <https://www.uagro.mx/>, Email: 03180@uagro.mx

estos obstáculos es el rezago educativo, debido en parte a que los programas educativos no están diseñados para satisfacer las necesidades de los estudiantes con discapacidad auditiva, esto ha llevado a que niñas, niños y adolescentes sordos se vean privados recibir educación escolar [3].

Gracias al avance de las diferentes técnicas de Inteligencia Artificial (IA), particularmente el Aprendizaje Automático o Machine Learning (ML), se han desarrollado sistemas informáticos capaces de traducir lenguas de señas. Este tipo de sistemas utilizan modelos de ML previamente entrenados con datos de muestra para reconocer patrones, permitiéndoles identificar estos patrones en nuevos datos [4]. Los sistemas traductores de señas utilizan modelos de ML, que se entrenan con imágenes y videos de personas realizando diferentes tipos de señas [5][6][7].

En este trabajo, se aborda el reconocimiento de imágenes de las señas correspondientes al alfabeto dactilológico de la LSM, que no requieren movimiento para su realización, mediante la implementación de modelos de ML de reconocimiento y clasificación de imágenes. Se recolectaron secuencias de video de personas realizando señas, posteriormente se extrajeron los fotogramas de los videos y en cada uno de ellos se extrajo la región de la mano con la configuración de cada seña, luego con estas nuevas imágenes obtenidas se recolectó un set de datos o dataset compuesto de 126,000 imágenes, que se utilizó para entrenar modelos de ML empleando cuatro técnicas diferentes, los cuales se evaluaron para determinar el modelo con mejor rendimiento en el reconocimiento de nuevas imágenes de señas.

Objetivos

Objetivo general

Evaluar el rendimiento de cuatro diferentes técnicas de Machine Learning para el reconocimiento de señas correspondientes al alfabeto dactilológico de la Lengua de Señas Mexicana.

Objetivos específicos

- Recolectar un dataset de imágenes de señas del alfabeto dactilológico de la Lengua de Señas Mexicana.
- Desarrollar modelos de Machine Learning para el reconocimiento de señas implementando cuatro técnicas diferentes.
- Evaluar el porcentaje de reconocimiento de cada modelo en un conjunto de nuevas imágenes de señas.

Metodología y procesos de desarrollo

Primeramente, se analizó la gramática de la LSM, para diferenciar el tipo de señas fijas y las que requieren movimiento en su realización. También se analizaron diferentes técnicas para la creación de modelos de ML de reconocimiento de imágenes y videos que pueden implementarse en el proceso de reconocimiento de expresiones de la LSM. Para el entrenamiento de los modelos de ML se obtuvieron videos de personas realizando diferentes señas para la elaboración del dataset. Posteriormente se realizó el entrenamiento de los modelos de ML y finalmente se realizaron pruebas en nuevas imágenes de señas. El rendimiento obtenido en cada técnica se muestra en el apartado de resultados. El esquema de procedimiento general se muestra en la Figura 1.



Figura 1. Esquema general propuesto

Lengua de Señas Mexicana

Es la lengua de la comunidad sorda en México, se considera como la lengua oficial de señas de México desde 2005, de acuerdo con el Consejo Nacional para el Desarrollo y la Inclusión de las Personas con Discapacidad, esta lengua es muy rica y compleja tanto en gramática como en vocabulario como cualquier lengua oral [8]. En la LSM el intercambio de mensajes es a través del uso de señas manuales y gestos faciales que el destinatario recibe a través de la visión.

En la LSM se pueden diferenciar dos tipos principales de señas, primeramente las señas que se realizan con una expresión manual fija, estas señas componen principalmente el alfabeto dactilológico a excepción de las letras J, K, Ñ, Q, X y Z y las letras dobles como LL y RR, (Figura 2); el segundo tipo principal de señas diferenciadas son las que implementan movimientos manuales así como expresiones faciales para su realización, con este tipo de señas se expresan la gran mayoría de palabras de la LSM [9].



Figura 2. Alfabeto dactilológico de la LSM.

Obtención y procesamiento de los datos

La obtención de datos se obtuvo mediante el trabajo con 8 alumnos del Instituto Tecnológico de Chilpancingo, Guerrero. Primeramente, se grabaron videos de las 21 señas fijas del alfabeto dactilológico de la LSM, siendo las señas correspondientes a las letras A, B, C, D, E, F, G, H, I, L, M, N, O, P, R, S, T, U, V, W e Y. Los videos tuvieron una duración de 1 minuto cada uno, por cada alumno se grabó un video para cada una de las 21 señas, en ubicaciones distintas para procurar que las imágenes resultantes fueran variadas, en total se obtuvieron 8 videos por cada seña, dando un total de 168 videos. Posteriormente se extrajeron los fotogramas de cada video, y luego, con la ayuda de la librería MediaPipe, se extrajo la región de la mano realizando la seña y posteriormente se guardó como imagen de tamaño 200 x 200 pixeles. Con este proceso se obtuvieron aproximadamente 126,000 imágenes, correspondientes a 6,000 imágenes en promedio por cada seña. En la Figura 3 se muestran ejemplos de las imágenes obtenidas.



Figura 3. Ejemplos de las imágenes obtenidas.

Una vez obtenidas las imágenes, se eligieron las más representativas, cuidando aspectos como que las imágenes no fueran tan similares, como los videos contenían en promedio 29 fotogramas por segundo, se eligieron imágenes que tuvieran suficiente variación en la posición de la mano, también se descartaron las imágenes donde la forma de la seña no fuera clara, o que tuviera desenfoco por movimiento de la mano. Después de este proceso se obtuvo un dataset final de 65,520 imágenes, correspondientes a 3,120 para cada seña.

Entrenamiento de modelos de ML

El ML se utiliza para crear patrones de reconocimiento en datos de texto, audio, imagen o video y así reconocer elementos de nuevos datos y agruparlos en los patrones creados [10].

De acuerdo con Torres et al. [11], para el entrenamiento de modelos de reconocimiento de vocabulario de señas, es importante utilizar un conjunto de datos de señas para el entrenamiento y otro usado como conjunto de validación. El conjunto de entrenamiento se utiliza para que los modelos reconozcan patrones y posteriormente ajustar los parámetros del modelo, mientras que el conjunto de validación se utiliza para evaluar el rendimiento del

modelo en datos nuevos. Cuando se tiene el modelo entrenado se prueba con nuevos datos para hacer el reconocimiento de las señas. En la Figura 4 se muestra el proceso de entrenamiento de modelos de ML para el reconocimiento de señas.

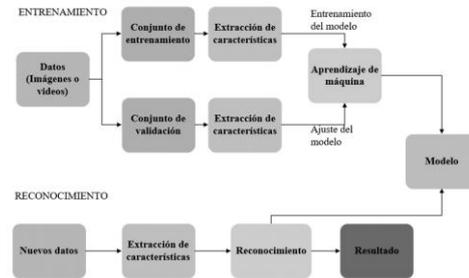


Figura 4. Entrenamiento de un modelo de ML para el reconocimiento de señas.

Con el dataset de 65,520 imágenes se entrenaron los modelos de ML implementando las siguientes técnicas: La primera técnica consistió en el entrenamiento de una Red Neuronal Convocucional (RNC) proporcionándole las imágenes de forma directa a la red, la segunda fue utilizando la técnica de Transfer Learning, que consistió en reentrenar una red neuronal previamente entrenada, la tercera técnica consistió en crear un modelo de ML utilizando la plataforma Teachable Machine (TM) [12] y la cuarta técnica consistió en entrenar nuevamente una RNC pero esta vez utilizando características específicas extraídas de las imágenes. Finalmente, en cada uno de los modelos se realizaron pruebas de inferencias o reconocimiento en 100 imágenes nuevas por cada seña. El procedimiento de cada técnica se explica a continuación:

1. Red Neuronal Convocucional

En este experimento se entrenó una RNC en el lenguaje de programación Python, utilizando las librerías Tensorflow y Keras. Para el proceso de entrenamiento se dividieron los datos en 70% para el entrenamiento correspondiente a 2,400 imágenes y 30% para validación, correspondientes a 720 imágenes. Posteriormente se guardó el modelo de ML obtenido en formato “.h5” con el cual se realizaron pruebas de inferencia en nuevos datos.

2. Transfer Learning

En esta técnica se reentrenó el algoritmo YOLO V5 [13], el cual es un algoritmo para el reconocimiento y seguimiento de objetos en tiempo real. Para este experimento se realizaron varias pruebas modificando el tamaño del set de datos, debido a que al realizar el entrenamiento en la plataforma de Google Colab, se presentaron problemas para completar el proceso. Finalmente, el entrenamiento pudo realizarse utilizando 1,000 imágenes para el entrenamiento y 300 para validación.

3. Plataforma Teachable Machine

Esta plataforma desarrollada por Google permite la creación de modelos de ML de forma rápida y fácil sin la necesidad de conocimientos en programación. En esta plataforma se realizaron varias pruebas con diferentes cantidades de imágenes para la creación del modelo, ya que al tratar de crear el modelo con 2,400 imágenes por seña la página dejaba de responder, el número máximo de imágenes con el que pudo crearse el modelo fue de 1,000 para cada una de las 21 señas.

4. Red Neuronal Convolutiva con características específicas

Primeramente, se extrajeron características de las imágenes, que consistieron en las coordenadas (X, Y) de cada uno de los 21 puntos del Landmark detectado en las manos por la librería MediaPipe [14], tal como se muestra en la Figura 5. Posteriormente, estas coordenadas se exportaron a un archivo CSV.

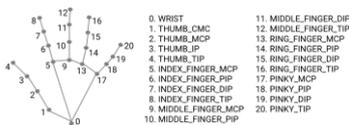


Figura 5. Puntos del Landmark de detectados por MediaPipe [14].

El archivo CSV resultante contenía 42 columnas correspondientes a las coordenadas en X y en Y de los puntos detectados y 1 columna más para la etiqueta de cada letra, en un rango del 0 al 20; así como 19,956 filas correspondientes a los datos de las imágenes en las que se pudieron obtener las coordenadas del Landmark.

Este archivo CSV se utilizó para entrenar una red neuronal convolutiva en el lenguaje de programación Python, utilizando la librería Keras. Posteriormente se realizaron pruebas de entrenamiento, dividiendo los datos en 70% y 20%, y 80% y 20% para entrenamiento y validación respectivamente, finalmente se optó por la segunda configuración de división de los datos, ya que esta alcanzó mayores valores de precisión en el entrenamiento y la validación del modelo.

Una vez entrenada la red neuronal se exportó el modelo de ML a un archivo de extensión "h5" con el cual se realizaron nuevamente pruebas de inferencias con nuevos datos de imágenes.

Resultados

1. Red Neuronal Convolutiva

En este caso se entrenó una RNC con tres capas de convolución, de 32 filtros de tamaño (3, 3), 64 filtros de tamaño (3, 3), 128 filtros de tamaño (3, 3) con función de activación "relu", respectivamente, y tres capas de Max

Pooling de tamaño (2, 2) respectivamente, luego una capa Flatten del tipo densa de 256 neuronas, con función de activación "relu" y un dropout de 0.5, y finalmente una capa densa de 21 neuronas y función de activación "softmax".

La red se compiló con los siguientes parámetros: optimizer: "Adam", loss: "categorical_crossentropy", y metrics: "accuracy". La red se entrenó a 20 épocas, obteniendo los siguientes resultados en la última época: loss: 0.0187, accuracy: 0.9956, val_loss: 0.0351 y val_accuracy: 0.9949. La gráfica de precisión del modelo se muestra en la Figura 6.

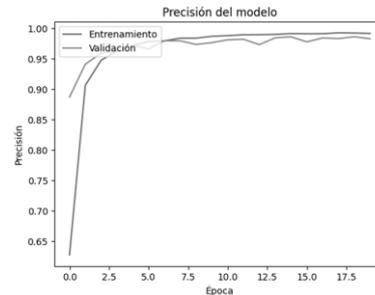


Figura 6. Gráfica de precisión de la RNC.

En las Figura 6 se aprecia que la precisión en los datos de validación comienza a asemejarse al entrenamiento después del 95% a partir de las primeras épocas.

Posteriormente se exportó el modelo al formato ".h5", con este modelo se realizaron las pruebas de inferencia en el conjunto de 100 nuevas imágenes por cada seña, los resultados obtenidos se muestran en la Tabla 1.

Tabla 1. Resultados de la inferencia del modelo de la técnica 1.

Seña	Inferencias correctas	Inferencias incorrectas	Seña	Inferencias correctas	Inferencias incorrectas
A	40	60	N	35	65
B	0	100	O	54	46
C	76	24	P	97	3
D	0	100	R	85	15
E	60	40	S	61	39
F	84	16	T	100	0
G	85	15	U	100	0
H	74	26	V	100	0
I	100	0	W	98	2
L	100	0	Y	90	10
M	18	82			

En la Tabla 2 se aprecia que las señas correspondientes a las letras M, N, S, y W tuvieron un número muy bajo de reconocimientos correctos, esto puede ser debido a que las señas M y N, S y T, y V y W son muy parecidas en su configuración manual.

2. Transfer Learning

El entrenamiento del algoritmo YOLO V5 se realizó en la plataforma Colab, utilizando el notebook oficial de Ultralytics [15] con los parámetros establecidos por defecto para el entrenamiento, tales como el tamaño de imagen de 640 x 640 pixeles, tamaño de batch de 16, número de épocas 20 y el archivo de pesos utilizado fue "yolov5s.pt".

Los resultados obtenidos al realizar pruebas con las nuevas imágenes se muestran en la Tabla 2.

Tabla 2. Resultados de la inferencia del modelo de la técnica 2.

Seña	Inferencias correctas	Inferencias incorrectas	Seña	Inferencias correctas	Inferencias incorrectas
A	98	2	N	100	0
B	99	1	O	90	10
C	97	3	P	99	1
D	100	0	R	100	0
E	100	0	S	100	0
F	100	0	T	100	0
G	18	82	U	98	2
H	100	0	V	100	0
I	100	0	W	100	0
L	100	0	Y	100	0

En la Tabla 2 se puede apreciar que las señas que tuvieron mayores problemas al reconocerse fueron la G y M, con un número de inferencias correctas menor a 90.

3. Plataforma Teachable Machine

La plataforma de TM permite guardar los modelos creados en varios formatos, en este caso se descargó el formato de Tensorflow, específicamente con extensión “.h5” y posteriormente se realizaron las pruebas de inferencias en las nuevas imágenes. En la Tabla 3 se muestran los resultados obtenidos.

Tabla 3. Resultados de la inferencia del modelo de la técnica 3.

Seña	Inferencias correctas	Inferencias incorrectas	Seña	Inferencias correctas	Inferencias incorrectas
A	68	32	N	98	2
B	0	100	O	54	46
C	70	30	P	11	89
D	45	55	R	18	82
E	76	24	S	0	100
F	18	82	T	11	89
G	58	42	U	99	1
H	64	36	V	83	17
I	91	9	W	66	34
L	90	10	Y	4	96
M	1	99			

Como se puede apreciar en la Tabla 3 los resultados obtenidos fueron malos en términos generales, con casos donde el reconocimiento correcto fue de 0 como en las señas B y S, además solo las señas I, L, N, y U tuvieron reconocimientos arriba de 90.

4. Red Neuronal Convolutiva con características específicas

En este caso se entrenó una red neuronal del tipo perceptrón multicapa o *Feedforward*, el modelo tiene una primera capa de entrada de 256 neuronas con función de activación “*relu*”, luego se asignó una capa de *dropout* con una tasa de 0.5, que indica que se utilizarán solamente el 50% de las neuronas de forma aleatoria para evitar el sobreajuste durante el entrenamiento. Después, se configuró una capa oculta de 128 neuronas con función de activación “*relu*”, con una capa de *dropout* de 0.3. Posteriormente, se configuraron dos capas ocultas más, una de 64 neuronas y la otra de 32 neuronas, con funciones de activación “*relu*”, cada una con capas de *dropout* de 0.2 y 0.1 respectivamente. Finalmente, la capa

de salida tiene 21 neuronas, esto para clasificar las 21 señas fijas del abecedario dactilológico de la LSM, esta última capa tiene una función de activación “*softmax*”. La red se compiló con los siguientes parámetros: *optimizer*: “*Adam*”, *loss*: “*categorical_crossentropy*”, y *metrics*: “*accuracy*”.

Para el entrenamiento se realizaron varios experimentos modificando el número de épocas entre 50, 100, 150, 200 y 300, al final se optó por monitorear la métrica “*val_accuracy*” con una paciencia de 10, para detener el entrenamiento si mejoraba después de transcurridas 10 épocas, con este ajuste el entrenamiento se detuvo en 120 épocas, en la que se obtuvieron los siguientes valores; *loss*: 0.1167, *accuracy*: 0.9635, *val_loss*: 0.0593 y *val_accuracy*: 0.9855.

A continuación, se muestran las gráficas de precisión, pérdida y la matriz de confusión del modelo:

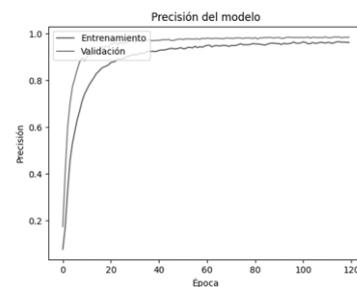


Figura 7. Gráfica de precisión del modelo.

En la Figura 7, se muestra la precisión alcanzada del modelo durante las épocas de entrenamiento, a partir de aproximadamente la época 40 la precisión de los datos de entrenamiento y validación comenzó a converger, siendo una precisión final en el entrenamiento del 96% y en los datos de validación del 98%.

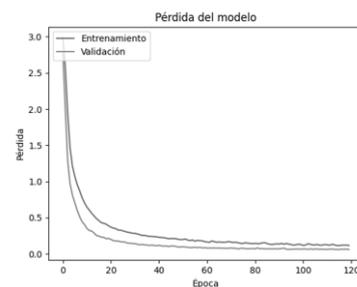


Figura 8. Gráfica de pérdida del modelo.

En la Figura 8, se muestra la tasa de pérdida de los datos de entrenamiento y prueba, a partir de la época 40 los datos comienzan a converger a 0 y se alcanza una pérdida en el entrenamiento de 0.1315... y en la validación de 0.0669...

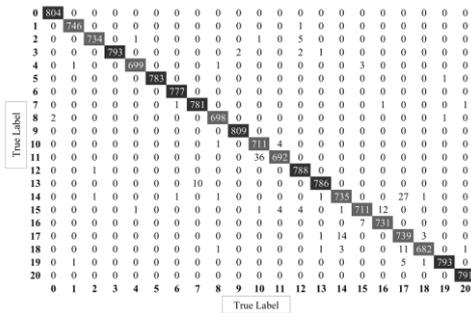


Figura 9. Matriz de confusión del modelo.

En la Figura 9, la matriz de confusión muestra en los ejes X y Y las etiquetas correspondientes a las 21 letras del entrenamiento, en la diagonal principal se muestran los valores reconocidos correctamente y en las demás celdas se muestran los falsos positivos. Como se puede apreciar la mayoría de los datos fueron reconocidos correctamente.

Posteriormente se realizaron las pruebas de inferencias en las nuevas imágenes, los resultados obtenidos se muestran en la Tabla 4.

Tabla 4. Resultados de la inferencia del modelo de la técnica 4.

Seña	Inferencias correctas	Inferencias incorrectas	Seña	Inferencias correctas	Inferencias incorrectas
A	100	0	N	100	0
B	100	0	N	100	0
C	100	0	O	100	0
D	100	0	P	100	0
E	100	0	R	100	0
F	100	0	S	100	0
G	100	0	T	100	0
H	100	0	U	100	0
I	100	0	V	100	0
L	100	0	W	89	11
M	93	7	Y	100	0

En la tabla 4 se aprecia que únicamente en las señas correspondientes a las letras M y W se realizaron inferencias incorrectas, siendo 7 y 11 respectivamente, y por en las señas restantes el porcentaje de reconocimiento correcto fue del 100%.

Conclusiones

En este estudio se analizaron y compararon cuatro técnicas diferentes de desarrollo de modelos de ML para el reconocimiento de señas de la LSM. El objetivo fue evaluar el rendimiento de estas técnicas en el entrenamiento y reconocimiento de nuevas imágenes de señas del alfabeto dactilológico de la LSM.

De acuerdo al objetivo general y al objetivo específico de desarrollar modelos de ML y en concordancia con los resultados obtenidos podemos concluir que el modelo que mejor rendimiento demostró fue el obtenido con la técnica de Red Neuronal Convolutiva con características específicas, puesto que demostró una alta capacidad para reconocer características de nuevas imágenes, lo que

condujo a una buena precisión de la clasificación de nuevas imágenes de señas.

Esto indica que al realizar el entrenamiento con extracción de características específicas de las imágenes de señas el rendimiento es mejor que al entrenar el modelo con los datos de las imágenes completas, como fue en el caso de la primera técnica, donde a pesar de que se obtuvo un valor de *accuracy* del 99% a lo hora de realizar inferencias en nuevas imágenes el rendimiento no fue el esperado.

En el caso de los modelos desarrollados con las técnicas de la Plataforma Teachable Machine y Transfer Learning, los resultados sugieren que es necesario continuar realizando pruebas de entrenamiento con mayor número de imágenes, para mejorar el rendimiento en el reconocimiento de nuevas señas.

Aunque en este estudio únicamente se analizaron señas fijas, los resultados sugieren que en futuros trabajos se podría explorar el desarrollo de modelos para el reconocimiento de señas con movimiento, al implementar MediaPipe para extraer características de puntos específicos de las manos y otras partes del cuerpo, a partir de secuencias de video, que servirían para entrenar modelos de reconocimiento de secuencias de datos.

Para poder desarrollar los modelos fue necesario recolectar los datos de imágenes a través de la grabación de videos de las señas realizadas por un equipo 8 personas, ya que en la actualidad no se cuenta con un base de datos señas de la LSM. El dataset resultante mostró ser una buena fuente de datos para entrenar modelos de reconocimiento de señas, por lo que este dataset obtenido puede servir como base para futuros trabajos enfocados al reconocimiento de la LSM mediante técnicas de ML.

Agradecimientos

Agradecemos a la coordinación de la Maestría en Ingeniería para la Innovación y Desarrollo Tecnológico por prestar sus ubicaciones para poder llevar a cabo la realización de este proyecto. Agradecemos de igual forma a los alumnos del Instituto Tecnológico de Chilpancingo, que generosamente brindaron su tiempo en la realización de las actividades solicitadas, sin su apoyo este trabajo no hubiera podido llevarse a cabo. Agradecemos también al CONACYT por el apoyo económico brindado durante la realización de este proyecto.

Referencias

[1] Instituto Nacional de Estadística y Geografía. (2020). Población con discapacidad o limitación en la actividad cotidiana por entidad federativa y tipo de actividad realiza según sexo, 2020. https://www.inegi.org.mx/app/tabulados/interactivos/?pq=Discapacidad_Discapacidad_02_2c111b6a-6152-40ce-bd39-6fab2c4908e3&idrt=151&opc=t, fecha de consulta: 19 de octubre de 2022.

- [2] Diario Oficial de La Federación. (2011). Ley General para la Inclusión de las Personas con Discapacidad. Secretaría de Gobernación. https://dof.gob.mx/nota_detalle.php?codigo=5191516&fecha=30/05/2011#gsc.tab=0, fecha de consulta 19 de octubre de 2022.
- [3] Cámara de Diputados. (10 de febrero de 2021). Aprueban reformas para que personas con discapacidad auditiva reciban educación bilingüe en lengua de señas. Honorable Cámara de Diputados. <https://comunicacionnoticias.diputados.gob.mx/comunicacion/index.php/boletines/aprueban-reformas-para-que-personas-con-discapacidad-auditiva-reciban-educacion-biling-e-en-lengua-de-se-as#gsc.tab=0>, fecha de consulta: 19 de octubre de 2022.
- [4] Alpaydin, E. (2010). Introduction to machine Learning. The MIT Press. Massachusetts. ISBN 978-0-262-01243
- [5] Mejía, K. (2022). Sistema de reconocimiento del Lenguaje de Señas Mexicano basado en una cámara RGB-D y aprendizaje automático. (Tesis inédita de maestría). Universidad Autónoma Querétaro. Querétaro, México.
- [6] Trujillo, F., García, G. (2021). Reconocimiento de palabras de la Lengua de Señas Mexicana utilizando información RGBD. RECIBE 10(2). DOI: <https://doi.org/10.32870/recibe.v10i2.209>
- [7] Jin, M., Omar, Z. & Hisham M. (2016). A Mobile Application of American Sign Language Translation via Image Processing Algorithms. 2016 IEEE Region 10 Symposium, DOI: 10.1109/TENCONSpring.2016.7519386
- [8] CONADIS. (2016). Lengua de Señas Mexicana (LSM). Gobierno de México. <https://www.gob.mx/conadis/articulos/lengua-de-senas-mexicana-lsm?idiom=es#:~:text=La%20Lengua%20de%20Se%C3%B1as%20Mexicana,propia%20sintaxis%2C%20gram%C3%A1tica%20y%201%C3%A9xico>, fecha de consulta: 13 de marzo de 2023.
- [9] Calvo, M. T. (2004). Diccionario Español-Lengua de Señas Mexicana (DIELSEME): estudio introductorio. Dirección de Educación Especial: México.
- [10] Tang, D. (2019). Empowering novices to understand and use machine learning with personalized image classification models, intuitive analysis tools, and MIT App Inventor. (Thesis, Massachusetts Institute of Technology). <https://dspace.mit.edu/handle/1721.1/123130>, fecha de consulta: 10 de noviembre de 2022.
- [11] Torres, S., García, C., Cabeza, C. y Docío, L. (2020). “LSE_Lex40_UVIGO: Una base de datos específicamente diseñada para el desarrollo de tecnología de reconocimiento automático de LSE”. Revista de Estudios de Lenguas de Signos REVLES, 2: 151-172.
- [12] Teachable Machine. (2023)- ¿Qué es Teachable Machine? <https://teachablemachine.withgoogle.com/>
- [13] Yolo. (2023). YOLOv5: The friendliest AI architecture you'll ever use. <https://ultralytics.com/yolov5>.
- [14] MediaPipe. (2023). Hand landmarks detection guide. https://developers.google.com/mediapipe/solutions/vision/hand_landmarker
- [15] YoloV5. (2023). Yolo V5 Github Repository. <https://github.com/ultralytics/yolov5>