



Reconocimiento de lengua de señas colombiana mediante redes neuronales convolucionales y captura de movimiento

Colombian sign language recognition using convolutional neural networks and motion capture

Jimmy Alejandro Plazas López ¹, Juan José Gutiérrez Leguizamón ², Marco Javier Suárez Barón ³ Juan Sebastián González Sanabria ⁴

Fecha de Recepción: 20 de enero de 2022

Fecha de Aceptación: 04 de julio de 2022

Cómo citar: Plazas López, J.A. Gutiérrez Leguizamón, J.J. Suárez Barón, M.J. y González Sanabria, J.S. (2022). Reconocimiento de lengua de señas colombiana mediante redes neuronales convolucionales y captura de movimiento. *Tecnura*, 26(74), 70-86. <https://doi.org/10.14483/22487638.19213>

Resumen

Contexto: Este artículo presenta el diseño de un modelo predictivo computacional que facilita el reconocimiento de la lengua de señas colombiana (LSC) en un entorno hotelero y turístico.

Método: Se aplicaron técnicas de inteligencia artificial y redes neuronales profundas en el aprendizaje y la predicción de gestos en tiempo real, los cuales permitieron construir una herramienta para disminuir la brecha y fortalecer la comunicación. Se implementaron algoritmos de redes neuronales convolucionales sobre captura de datos en tiempo real. Se capturó movimiento mediante cámaras de video de dispositivos móviles; así, se obtuvieron las imágenes que forman el conjunto de datos. Las imágenes se utilizaron como datos de entrenamiento para un modelo computacional óptimo que puede predecir el significado de una imagen recién presentada.

Resultados: Se evaluó el rendimiento del modelo usando medidas categóricas y comparando diferentes configuraciones para la red neuronal. Adicional a esto, todo está soportado con el uso de herramientas como *Tensorflow*, *OpenCV* y *MediaPipe*.

Conclusiones: Se obtuvo un modelo capaz de identificar y traducir 39 señas diferentes entre palabras, números y frases básicas enfocadas al sector hotelero, donde se logró una tasa de éxito del 97,6 % en un ambiente de uso controlado.

Agradecimientos: Universidad Pedagógica y Tecnológica de Colombia (UPTC).

¹Estudiante de Ingeniería de Sistemas y Computación. Sogamoso, Colombia.

Email: jimmy.plazas@uptc.edu.co

²Estudiante de Ingeniería de Sistemas y Computación. Sogamoso, Colombia.

Email: juan.gutierrez04@uptc.edu.co

³Ingeniero de Sistemas, magíster en Gestión de Información, doctor en Planeación Estratégica y Dirección de Tecnología. Docente asociado de la Universidad Pedagógica y Tecnológica de Colombia. Sogamoso, Colombia.

Email: marco.suarez@uptc.edu.co

⁴Ingeniero de Sistemas y Computación, magíster en Ingeniería del Software. Docente asistente de la Universidad Pedagógica y Tecnológica de Colombia. Tunja, Colombia.

Email: juansebastian.gonzalez@uptc.edu.co

Palabras clave: ambiente hotelero, analítica de datos, aprendizaje profundo, comunicación, discapacidad fonoauditiva, lengua de señas colombiana, redes neuronales.

Abstract

Context: This article presents the design of a computational predictive model that facilitates the recognition of Colombian Sign Language (LSC) in a hotel and tourism environment.

Method: Artificial intelligence techniques and deep neural networks were applied in the learning and prediction of gestures in real time, which allowed the construction of a tool to reduce the gap and strengthen communication. Convolutional neural network algorithms were applied to real-time data capture. Movement was captured using mobile device video cameras, thus obtaining the images that make up the data set. The images were used as training data for an optimal computational model that can predict the meaning of a newly presented image.

Results: The performance of the model was evaluated using categorical measures and comparing different configurations for the neural network. In addition to this, everything is supported with the use of tools such as Tensorflow, OpenCV and MediaPipe.

Conclusions: Finally, a model capable of identifying and translating 39 different signs between words, numbers and basic phrases focused on the hotel sector was obtained, where a success rate of 97.6% was obtained in a controlled use environment.

Acknowledgements: Pedagogical and Technological University of Colombia – UPTC

Keywords: hotel environment, data analytics, deep learning, communication, phono-hearing disability, colombian sign language, convolutional neural network.

Tabla de Contenidos

	Página
Introducción	72
Metodología	74
Captura de datos	74
Preprocesamiento de datos	76
Implementación de la red neuronal convolucional	76
Clasificación y entrenamiento	77
Métricas de calidad	78
Resultados	79
Conclusiones	84
Financiamiento	84
Agradecimientos	84
Referencias	85

INTRODUCCIÓN

La comunicación es de vital importancia en nuestra vida diaria, ya que permite expresar necesidades, ideas, entre otras emociones, además de los sentidos como medio que facilita la relación con el entorno turístico. Ahora bien, en el sector hotelero la población con discapacidad fonoauditiva aún tiene barreras de comunicación. Aunque algunos hoteles han mejorado sus instalaciones al hacerlas accesibles, el personal no se encuentra capacitado para atender los diferentes tipos de discapacidades que puede presentar una persona. De otra parte, algunos establecimientos hoteleros expresan que para ellos no es necesario hacer capacitaciones debido a la alta rotación de personal, la mayoría de veces no contratan por más de tres meses o permiten la práctica de aprendices para solventar algunos cargos.

Este proyecto está limitado al desarrollo de un prototipo que reconozca y traduzca en tiempo real el lenguaje de señas colombiana, posteriormente se desarrolló la aplicación de un algoritmo de aprendizaje profundo enfocado en el modelo de predicción a partir del conjunto de datos haciendo uso de Python, OpenCV, Tensorflow y Keras como base para la construcción y entrenamiento del modelo predictivo, donde se implementaron métodos para las etapas de procesamiento de imágenes y aplicación de una red neuronal convolucional (Llanos Mosquera, 2021). A su vez, se analizó el desempeño del algoritmo implementado para conocer el nivel de eficacia.

El impacto que se obtiene al establecer una comunicación asertiva es la fuente de oportunidades para la comunidad sorda en Colombia. Sumado a lo anterior, los hoteles deben solventar las necesidades de accesibilidad no solo físicas sino de cualquier tipo de discapacidad, mediante la oferta de equidad tanto a clientes como a empleados (Congreso de la República de Colombia., 2020). En este sentido, se espera que las personas se involucren más en el turismo, sin miedo a no poder comunicarse en los hoteles que se implemente el modelo.

El modelo predictivo interpreta señas básicas de conversaciones cotidianas en un ambiente de recepción hotelera. El propósito es llegar a personas de diferentes edades y estratos sociales que sean identificados como personas con discapacidad fonoauditiva, teniendo en cuenta que en Boyacá (Colombia) hay 5644 personas sordomudas, de los cuales el 48 % son mujeres y el 52 % hombres (Insoy, 2020), sin embargo, esta situación se replica a nivel país, donde según el mismo Instituto existen 166 619 personas con limitación y de las cuales 47,8 % son de género femenino y el 52,2 % restante hombres. El uso de tecnologías emergentes debe ser un elemento vital en el diseño de los componentes físicos y lógicos de modelos computacionales. Por consiguiente, la población con discapacidad se verá beneficiada, ya que se proyectará un patrón de reconocimiento de información en los diferentes ambientes sociales, para que así se puedan utilizar servicios de almacenamiento e inteligencia artificial (IA) y dar una correcta optimización de cada lenguaje a la hora de tener una comunicación asertiva.

En este proyecto, se dan a conocer artículos y trabajos relacionados directa o indirectamente con una solución tecnológica al problema de comunicación entre personas oyentes que no dominan la lengua de señas colombiana, y aquellas con discapacidad fonoauditiva. En particular, se basa en el uso y la forma del lenguaje de señas al no ser universal; el cual, cada región o país puede tener diversas formas de expresarse, lo que hace complejo el desarrollo de un modelo tecnológico (Rostand *et al.*, 2019).

Para disminuir el problema de comunicación entre personas con deficiencia auditiva y personas oyentes. Se propone el diseño y desarrollo de un modelo predictivo computacional de un intérprete de lengua de señas colombiana (LSC) enfocado en el sector hotelero que permita la interpretación de señas en tiempo real (Mishra *et al.*, 2019).

Jiménez-Forero y Moreno-Mosquera, 2020 exploraron el desarrollo de un método automático para el reconocimiento del lenguaje de señas colombiano en cuanto a la traducción de las vocales y números. Este modelo está implementado que consta de seis etapas: a) procesamiento de datos (imágenes); b) preprocesamiento; c) un debido muestreo; d) extracción de las características más específicas posibles de cada imagen o conjunto de datos; e) obtención de una clasificación para la identificación del gesto con un porcentaje de reconocimiento óptimo, y f) medición estadística sobre el rendimiento del clasificador.

Las mediciones para mostrar retroalimentación parten por las imágenes que ellos mismos capturan y que utilizan como datos de entrenamiento para un modelo computacional óptimo que puede predecir el significado de una imagen recién presentada (Ortiz Farfán y Camargo Mendoza, 2020). Evalúan el rendimiento del método usando medidas categóricas y comparando diferentes modelos. Una vez seleccionados los mejores modelos, se prueban con nuevas imágenes, similares a las de entrenamiento. Se puede observar que el mejor modelo logra una tasa de éxito de alrededor del 68 % de las 22 clases utilizadas en el sistema.

La librería *OpenCv* y el *IDE QT Creator* tienen capacidad para realizar el seguimiento de las manos con un algoritmo que facilita el proceso a través del color; dicho algoritmo consta de otros ya desarrollados, como SVM y el KNN, con el objetivo de reducir errores porcentuales durante el proceso predictivo (Ortiz García, 2021). Así mismo, se recurre a redes neuronales convolucionales (CNN) y aceleración de la GPU, con el fin de automatizar el proceso de la creación de funciones y métodos de aprendizaje (Herazo, 2020). Pudieron implementar un modelo predictivo capaz de reconocer 20 gestos en lenguaje de señas italiano con gran precisión sobre usuarios y entornos que no ocurrieron durante el entrenamiento de la red neuronal y dando así un valor máximo del 91,7 %.

El principal resultado con diferentes configuraciones de entrenamiento, con y sin validaciones, alcanzó precisiones finales; basado en 8 sujetos de estudio y 9400 imágenes, es del 85 % para el modelo observado en comparación con nuevos modelos (Halvardsson *et al.*, 2021). El conjunto de datos se creó con cada imagen configurada a 320×240 píxeles.

También señalaron como fortaleza del modelo y de la implementación su costo, que es asequible a cualquier persona con el interés de investigar estas necesidades. Estos resultados indicaron que el

uso de CNN es un enfoque prometedor para interpretar los lenguajes de señas, y el aprendizaje de transferencia se puede usar para lograr una alta precisión en las pruebas, a pesar de usar un pequeño conjunto de datos de entrenamiento. Además, se describen detalles de implementación del modelo para interpretar signos como una aplicación web fácil de usar.

Por último, [Vázquez-Enríquez et al., 2021](#) mencionan las limitaciones que se tienen en cuanto a señas que involucran el contacto del cuerpo, especialmente el rostro, ya que el modelo está diseñado para el reconocimiento de las manos y solo hace un análisis de imágenes 2D. De esta manera se resalta la importancia de las características únicas que tiene cada gesto y significado que dificultan extrapolar una solución para la traducción del lenguaje ([Mustafa, 2020](#)).

METODOLOGÍA

La metodología de desarrollo se basa en una arquitectura dividida en cinco fases que se observan en la figura 1.

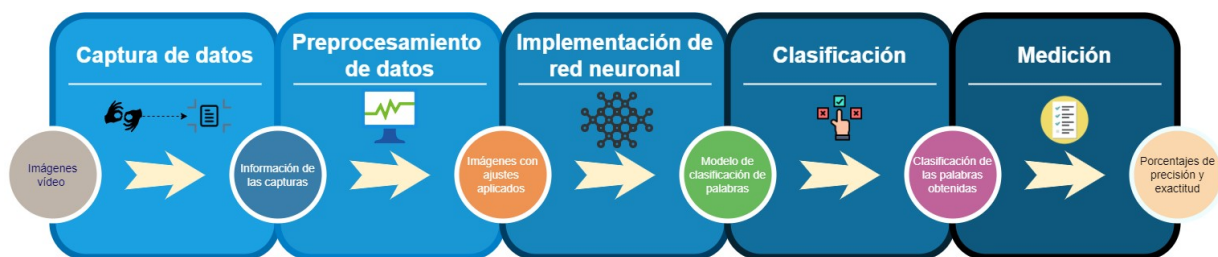


Figura 1. Metodología de desarrollo

Fuente: elaboración propia.

Captura de datos

El proceso para realizar las señas y movimientos correctos se basó en el diccionario de LSC ([Insor, s. f.](#)). Estos datos ingresaron al sistema a través del uso de una cámara que registró los cuadros (*frames*) de video. Luego, el algoritmo reconoció las manos y dibujó el *landmark* con los 21 puntos de interés; seguido de esto, se recortó la imagen de manera que quedaran únicamente las manos. Una vez se obtuvo la imagen recortada, se aplicó un redimensionamiento de (200 px, 200 px) y por último, se almacenó en la carpeta correspondiente a la clase de la seña, como se puede observar en la figura 2. De esta manera fue posible gestionar los datos para su posterior procesamiento. Para el proceso se tomaron 39 palabras incluyendo algunos números y frases básicas en lengua de señas.

En la tabla 1 se listan las señas seleccionadas.

Adicionalmente, se recurrió a un algoritmo de reconocimiento de manos que, a su vez, capturó automáticamente la imagen de la cámara y la almacenó en un directorio específico con la etiqueta de la seña que se capturó. La etiqueta de cada imagen es igualmente la clasificación final que van a tener las imágenes en la red neuronal.



Figura 2. Flujo de captura de datos

Fuente: elaboración propia.

Tabla 1. Lista de señas que reconoce el modelo

Señas seleccionadas para entrenar el modelo			
1	Sábado	Septiembre	Gracias
2	Domingo	Octubre	Habitación
3	Enero	Noviembre	Hola
4	Febrero	Diciembre	Hotel
5	Marzo	Adulto	Mal
Lunes	Abril	Bien	Niño
Martes	Mayo	Bienvenido	No
Miércoles	Junio	Cama	Por favor
Jueves	Julio	¿Cómo estás?	Sí
Viernes	Agosto	Con mucho gusto	

Fuente: elaboración propia.

La base primordial fue el procesamiento de imágenes y captura en tiempo real; por esto, el código se estructuró en Python. Así mismo, para el reconocimiento y seguimiento se aplicó la librería *MediaPipe Hands* (Google, s. f.) que detectó las manos en tiempo real, las palmas y los dedos.

Preprocesamiento de datos

Las imágenes almacenadas fueron preprocesadas aplicando tres diferentes ajustes. Para esta fase, primero se leyó el conjunto de datos a partir de la ruta donde se encuentra almacenado. Luego se aplicaron los ajustes iniciando con un cambio de tamaño para trabajar con imágenes cuadradas, específicamente con dimensiones (200px, 200px), haciendo uso del método *resize* de la librería de *OpenCV* (Medina Rojas, 2017). Luego se aplicó un reescalado a los píxeles de cada imagen, es decir que se cambió el valor de cada píxel de 0 y 255 a un valor decimal entre 0 y 1. Esto se hace para normalizar los datos. Y finalmente, se aplicó un porcentaje de *zoom*.

Ahora bien, los ajustes de reescalado y *zoom* se aplicaron aleatoriamente a imágenes para que el modelo no terminara aprendiendo la posición de los píxeles, sino para que fuera capaz de entender lo que había en la imagen sin importar el tamaño, posición o color. El flujo completo en secuencia se puede observar en la figura 3.

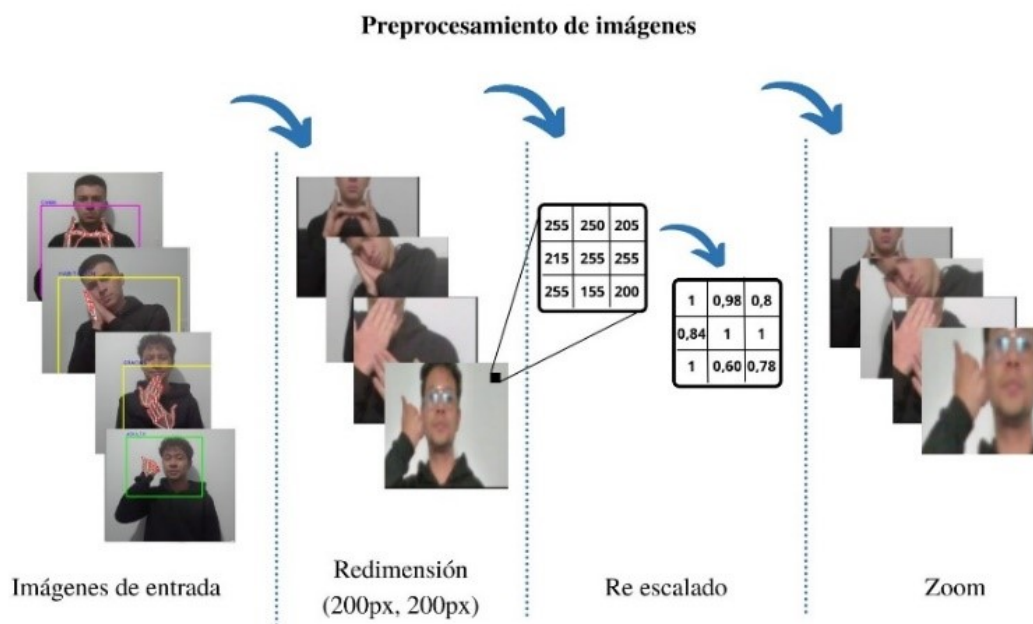


Figura 3. Flujo completo de preprocesamiento de imágenes

Fuente: elaboración propia.

Implementación de la red neuronal convolucional

Para la implementación de la red neuronal, se hicieron pruebas modificando valores en el tamaño del lote (*batch size*), tamaño del núcleo (*kernel*), iteraciones (*epochs*), número de capas y número de neuronas. De esta manera se pudo establecer la mejor configuración para que el modelo se entrenara, aprendiera y el resultado fuera óptimo para su uso. Finalmente, la estructura del modelo con su respectiva configuración se muestra a continuación en la figura 4.

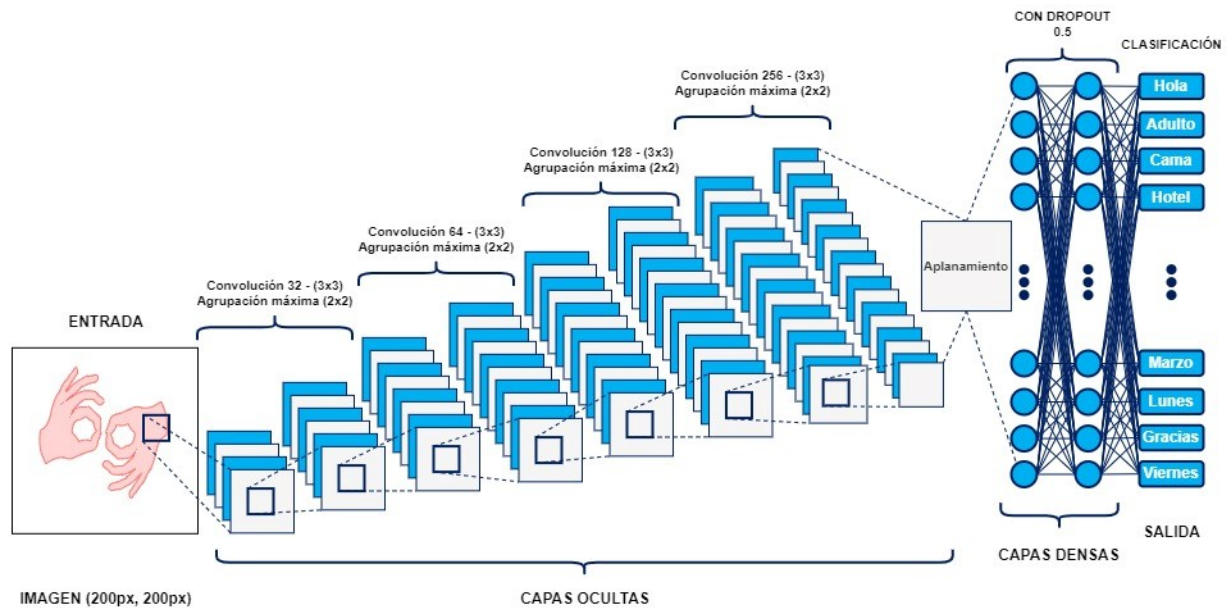


Figura 4. Estructura del modelo de red neuronal

Fuente: elaboración propia.

La red neuronal se compuso de cuatro capas convolucionales y cuatro de agrupación máxima, un aplanamiento de todas las capas, dos capas densas, un *dropout*, y finalmente la capa de clasificación. Además, se aplicaron funciones de activación Relu y en la última capa Softmax.

El modelo recibió como entrada una imagen de (200px, 200px), la cual pasa por 4 capas ocultas. La primera estuvo definida con 32 núcleos, la segunda con 64, la tercera con 128 y la última con 256. En todas las capas convolucionales anteriores la dimensión de cada núcleo fue definida como [3,3]. De igual manera en las capas de agrupación máxima la dimensión de cada núcleo fue definida como [2,2]. Luego de que la imagen pasó por las 4 capas ocultas, se hizo un aplanamiento para que tener una sola dimensión con toda la información. Luego pasó por 2 capas densas donde se definió un *dropout* de 0,5, esto quiere decir que en cada iteración se apagaba la mitad de las neuronas con el fin de no sobreajustar la red neuronal. Finalmente, se estableció una capa densa donde la cantidad de neuronas fue igual a la cantidad de señas usadas, y se estableció la activación Softmax. De esta manera, el modelo devolvió las probabilidades donde el número mayor fue la predicción final.

Clasificación y entrenamiento

En esta fase, se obtuvo una clasificación de las palabras identificadas en el modelo predictivo. A partir de aquí se pueden aplicar mediciones con el fin de obtener el mejor resultado como traducción. En el modelo de entrenamiento que se aplicó se establece la forma como se entrena y se predicen los resultados. Para el modelo se estableció el optimizador como "adam", la pérdida de la red neuronal es "categorical_crossentropy", y la métrica para medir la eficiencia del modelo es "accuracy".

Para el entrenamiento se definieron 3000 épocas o iteraciones, donde los resultados del modelo fueron el promedio de cada una de las combinaciones por iteración. Los pasos por cada iteración se definieron de acuerdo con la ecuación (1):

$$\text{Pasos por iteración} = \frac{\text{Número de imágenes por clase}}{\text{Tamaño de lote}} \quad (1)$$

$$31,25 = \frac{1000}{32}$$

Aplicando la ecuación (1), por cada clase se tomaron 1000 imágenes, y como tamaño de lote se estableció 32 que por lo general es el valor predeterminado. Con esto se obtuvo un valor para los pasos por iteración de 31,25.

Métricas de calidad

En esta última fase se establecieron los métodos de medición de calidad para conocer los porcentajes de precisión y exactitud que el modelo arroja como resultado al aplicarse. Las métricas seleccionadas fueron el uso de matriz de confusión, exactitud, F1-score, precisión y pérdida del modelo. Para poder medir los resultados de aprendizaje del modelo se recurrió a la herramienta *TensorBoard*. Esta permitió graficar los resultados de precisión y pérdida de todo el proceso de entrenamiento del modelo. Para esto, se usó el atributo *callback* en la función de entrenamiento del modelo, de esta manera en cada iteración se almacenaban los logs en una carpeta definida con *TensorBoard*.

Adicionalmente se definieron las siguientes ecuaciones como métodos de medición de resultados:

Precisión (p). Consiste en el conjunto de registros verdaderos positivos (VP) dividida entre la suma de verdaderos positivos (VP) y falsos positivos (FP) logrados en la ejecución del modelo.

$$P = \frac{VP}{VP + FP} \quad (2)$$

Exactitud (accuracy, ACC). Porcentaje de casos en los que el algoritmo ha acertado positivamente, en pocas palabras, es la medición del número de predicciones correctas sobre el número total de predicciones.

$$ACC = \frac{TP + TN}{FP + FN + TP + TN} \quad (3)$$

Exhaustividad (recall, R). Medición de integridad definida como el fragmento de la cifra de registros verdaderos positivos (VP) fraccionados por la adición de los verdaderos positivos (VP) y registros clasificados como falsos negativos (FN).

$$R = \frac{VP}{VP + FN} \quad (4)$$

F1-score. Función de exhaustividad y precisión que busca buscar un equilibrio entre la precisión y la exhaustividad.

$$F1 = 2 \cdot \frac{PRE \cdot REC}{PRE + REC} \quad (5)$$

RESULTADOS

Se desarrolló un algoritmo para el reconocimiento de las manos de cualquier persona en tiempo real, con el cual se hicieron las respectivas pruebas en diferentes ángulos, fondos e iluminación, donde la librería de *MediaPipe* brinda eficacia al momento de poner los 21 puntos en la mano de distintas personas que hagan la prueba frente a la cámara del ordenador portátil, a pesar de la baja calidad en cuanto a píxeles que esta brinde, como se muestra en la figura 5.

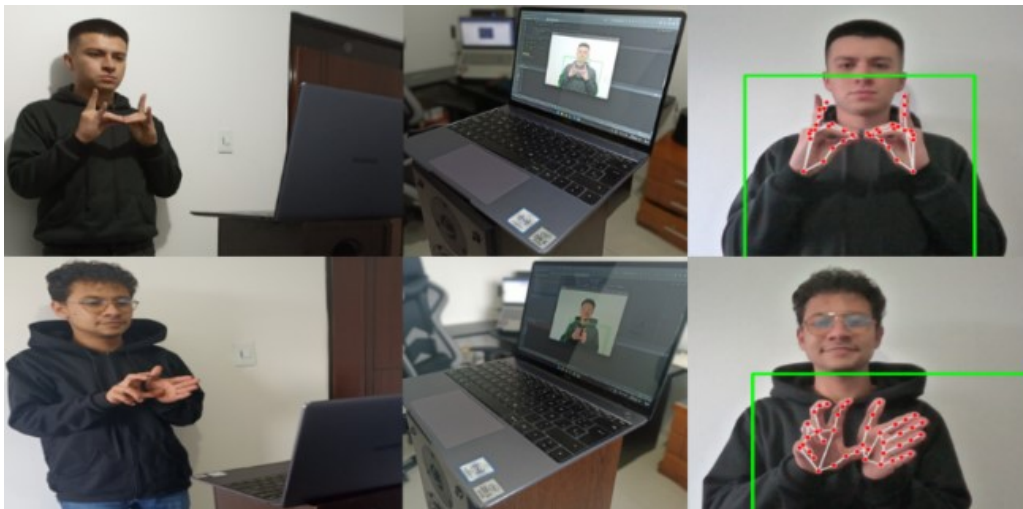


Figura 5. Prueba del detector de manos en tiempo real

Fuente: elaboración propia.

Como medida de desempeño del modelo se generó una matriz de confusión con base en la información que arrojaron las pruebas con 50 imágenes por cada señal del conjunto de datos. La matriz de confusión se produjo de manera que las columnas eran los datos que predijo, y las filas, los datos verdaderos, como se observa en la figura 6.

Para verificar la precisión del modelo creado se usó la herramienta *TensorBoard* y se importaron los *logs* y se generó la gráfica de precisión, lo que arrojó como resultado la figura 7.

A partir de la gráfica de precisión se puede observar que el modelo fue aprendiendo tanto con los datos de validación como con los datos de entrenamiento. A su vez se observó que aproximadamente en la iteración 1400 el modelo mantenía la precisión de aprendizaje, es decir que llegó a su punto máximo de aprendizaje, en este caso al 100 %.

De la misma manera, se generó la gráfica de pérdida (figura 8). En esta gráfica se muestra la tasa de pérdida tanto de los datos de entrenamiento como los datos de validación.

De la figura de la pérdida del modelo se puede observar que el modelo disminuye la tasa de pérdida a medida que incrementa el número de iteraciones, y que la pérdida del modelo tiende a ser 0. Además, a pesar de que aproximadamente a partir de la iteración 1100 el modelo aumenta la tasa de pérdida, sigue siendo la tasa bajo el 2 %.

Matriz de confusión

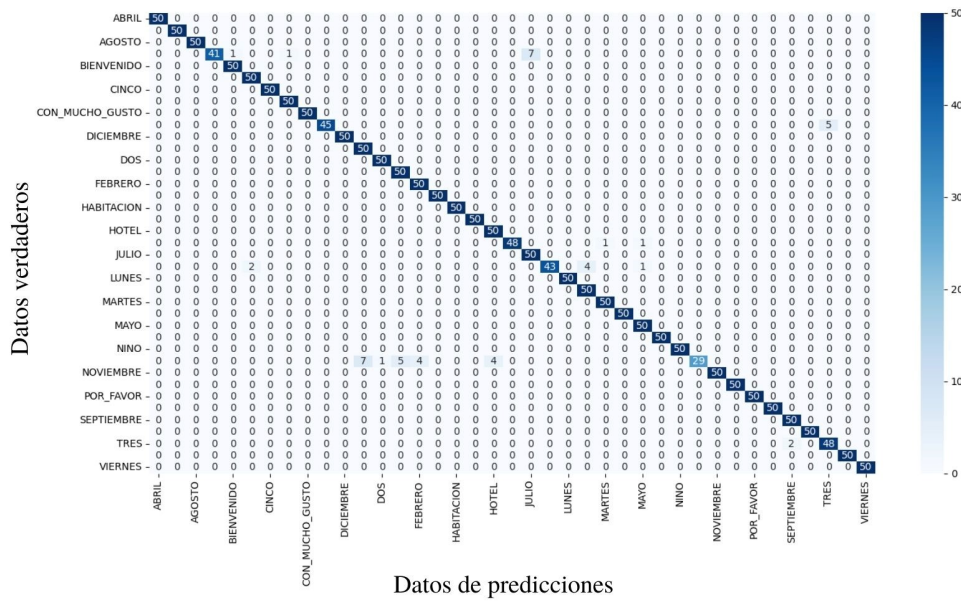


Figura 6. Matriz de confusión

Fuente: elaboración propia.

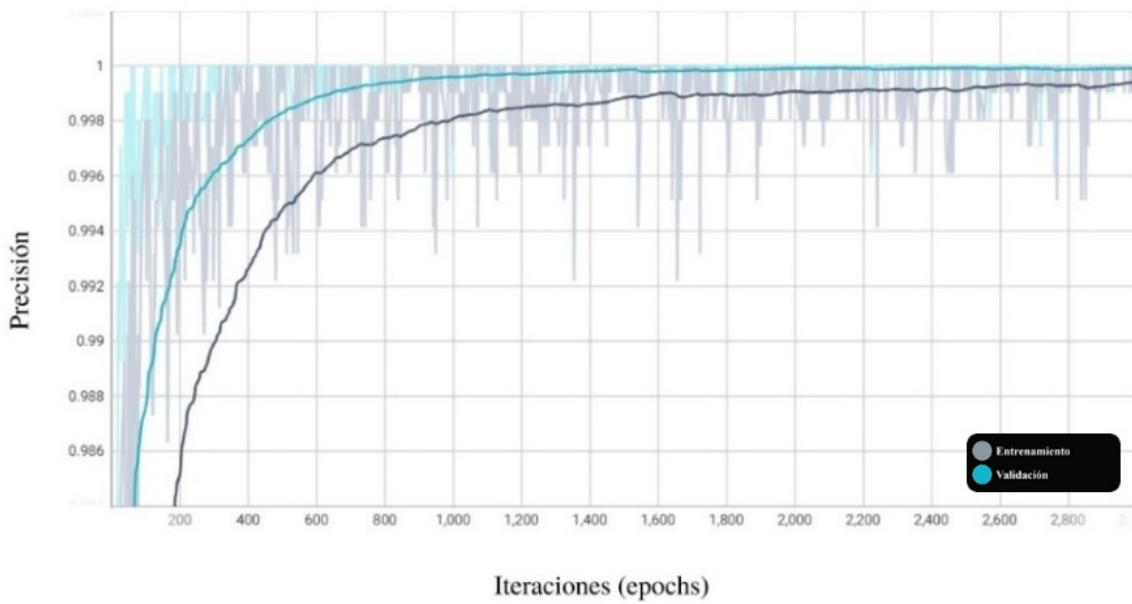


Figura 7. Precisión del modelo de red neuronal

Fuente: elaboración propia.

Se produjo una lista de resultados, como se puede observar en la tabla 2. Estos fueron generados a partir de una prueba con 50 imágenes de cada una de las señas que reconoce el modelo que se entrenó.

Tabla 2. Resultados de prueba del modelo de predicción en ejecución

Clase	Precisión	Recall	F1-score	Datos
ABRIL	1,000	1,000	1,000	50
ADULTO	1,000	1,000	1,000	50
AGOSTO	1,000	1,000	1,000	50
BIEN	1,000	0,820	0,901	50
BIENVENIDO	0,980	1,000	0,990	50
CAMA	0,961	1,000	0,980	50
CINCO	1,000	1,000	1,000	50
COMO_ESTA	0,980	1,000	0,990	50
CON_MUCHO_GUSTO	1,000	1,000	1,000	50
CUATRO	1,000	0,900	0,947	50
DICIEMBRE	1,000	1,000	1,000	50
DOMINGO	0,877	1,000	0,934	50
DOS	0,980	1,000	0,990	50
ENERO	0,909	1,000	0,952	50
FEBRERO	0,925	1,000	0,961	50
GRACIAS	1,000	1,000	1,000	50
HABITACION	1,000	1,000	1,000	50
HOLA	1,000	1,000	1,000	50
HOTEL	0,925	1,000	0,961	50
JUEVES	1,000	0,960	0,979	50
JULIO	0,877	1,000	0,934	50
JUNIO	1,000	0,860	0,924	50
LUNES	1,000	1,000	1,000	50
MAL	0,925	1,000	0,961	50
MARTES	0,980	1,000	0,990	50
MARZO	1,000	1,000	1,000	50
MAYO	0,961	1,000	0,980	50
MIERCOLES	1,000	1,000	1,000	50
NIÑO	1,000	1,000	1,000	50
NO	1,000	0,580	0,734	50
NOVIEMBRE	1,000	1,000	1,000	50
OCTUBRE	1,000	1,000	1,000	50
POR FAVOR	1,000	1,000	1,000	50
SABADO	1,000	1,000	1,000	50
SEPTIEMBRE	0,961	1,000	0,980	50
SI	1,000	1,000	1,000	50
TRES	0,905	0,960	0,932	50
UNO	1,000	1,000	1,000	50
VIERNES	1,000	1,000	1,000	50
Exactitud			0,976	1950

Fuente: elaboración propia.

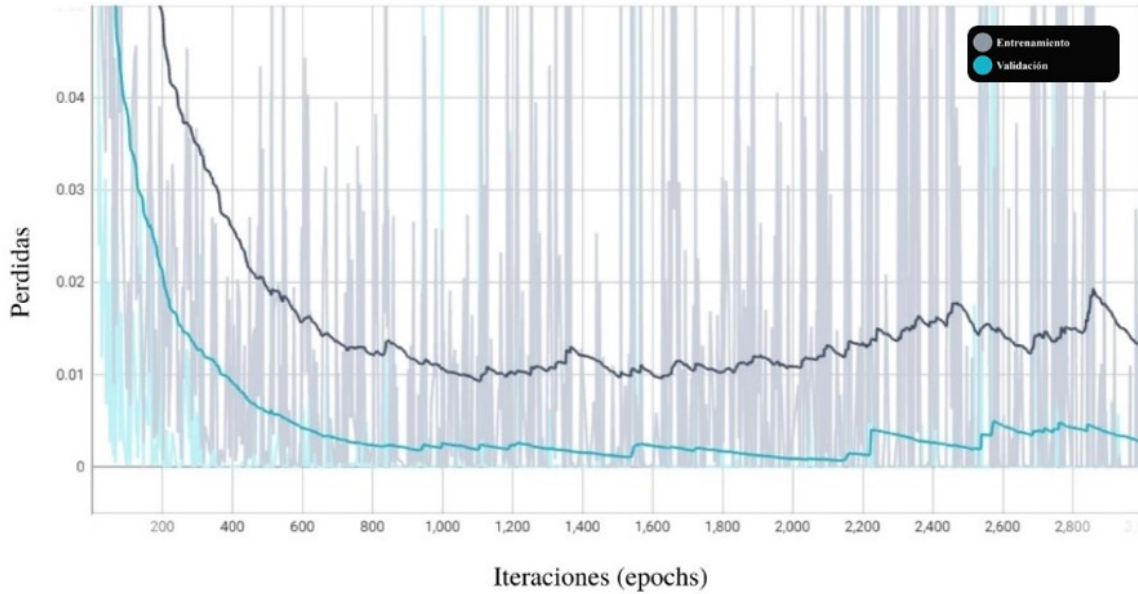


Figura 8. Taza de pérdida del modelo de red neuronal

Fuente: elaboración propia.

En los resultados se encuentran las señas usadas en el modelo con los respectivos resultados de cada uno. Entre los resultados se encuentran la precisión, *Recall* o exhaustividad, *F1-score*, y por último la cantidad de datos usados para la prueba (Trejos Buriticá 2018). Al final se muestra el total de la precisión y la totalidad de datos usados. En la figura 9 se recopilan todos los datos presentados en la tabla 2. Se pueden observar las tres mediciones donde el *F1-score* presenta el valor más bajo con

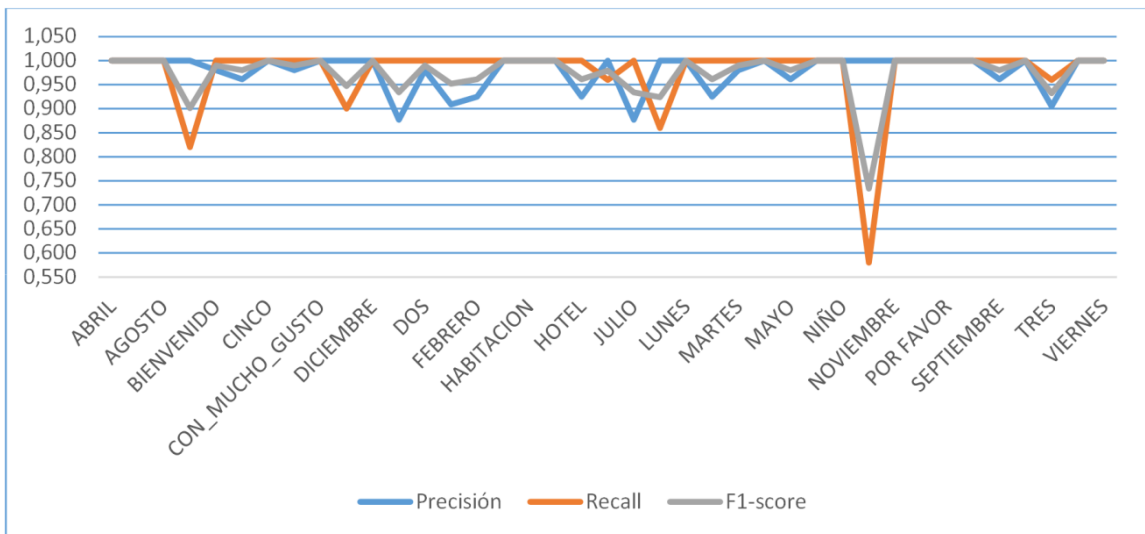


Figura 9. Resultados de rendimiento del modelo

Fuente: elaboración propia.

la seña “NO” con 0,734. De la misma manera el *Recall* tiene su valor más bajo de 0,580 con la misma palabra. Y a su vez, se puede observar la precisión del modelo que se mantiene sobre el 0,87 %.

En la figura 10 se observa la recopilación de capturas tomadas del modelo en funcionamiento, donde se presenta el reconocimiento de las manos dibujando en la pantalla el *landmark* con los puntos en las manos y el recuadro, la predicción y traducción de las señas hechas por los usuarios. Además, el modelo se usó con diferentes niveles de luminosidad y tonos de piel, lo que demuestra la eficacia del modelo predictivo.

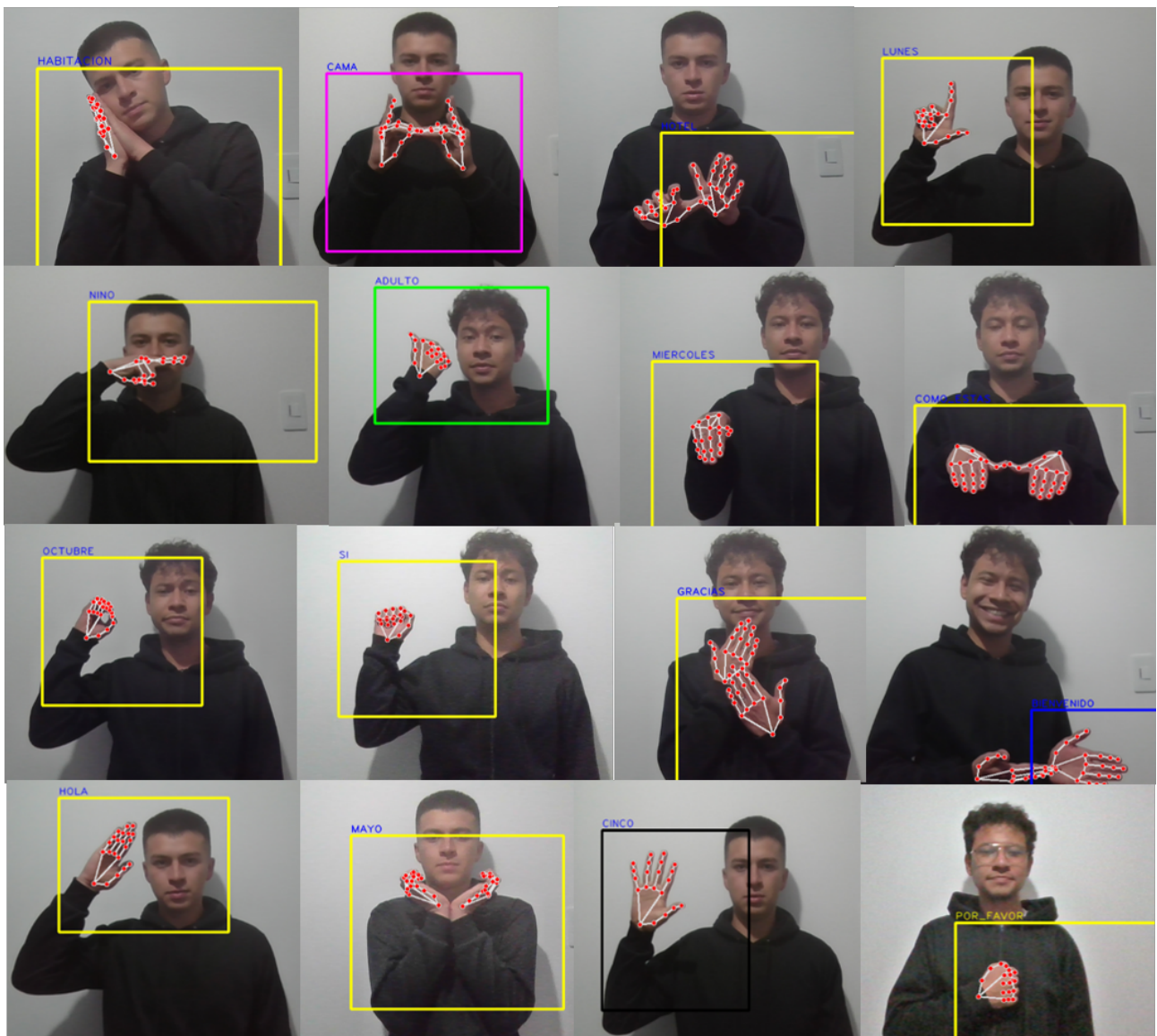


Figura 10. Resultados del modelo en ejecución

Fuente: elaboración propia.

CONCLUSIONES

Este proyecto dio a conocer que el modelo computacional predictivo utilizando la lengua de señas colombiana (LSC) enfocado en el sector hotelero, evidenció su efectividad para reducir la brecha y mejorar la comunicación entre personas con y sin pérdida auditiva. Esto se puede demostrar con el desarrollo del modelo predictivo, ya que al dar una clasificación de tipo múltiple comparando los resultados de la tabla 2 frente a las 39 clases (señas/palabras), se da respuesta de eficacia con alta puntuación predicha.

En cuanto al conjunto de datos creado, se dio como resultado un amplio número de imágenes y épocas procesadas por la red neuronal convolucional, dado que se tiene que procesar un alto volumen de datos donde se presenta alta complejidad por tratarse de imágenes. Además, se obtuvo como producto un conjunto de datos base para dar una predicción efectiva a la hora de aplicar el modelo en la LSC.

Cabe aclarar que este modelo aun no proporciona una exactitud del 100 % en cuanto a la traducción de la LSC, ya que hay señas similares en su forma y en ocasiones el modelo falla en la predicción, obteniendo resultados de porcentajes en efectividad un poco bajos, así como se logra evidenciar en la figura 6. Los productos obtenidos de las medidas de precisión, exactitud y exhaustividad fueron aceptables para el análisis manual visual y datos cuantitativos por el modelo logrado para el proceso de clasificación de cada seña que se logró traducir. Con los resultados de las medidas de rendimiento se dio a conocer que la red neuronal funcionó de manera óptima para la traducción del conjunto de datos establecidos.

Teniendo en cuenta lo anterior, este proyecto puede ser mejorado e implementado en más áreas. Se recomienda ampliar la cantidad de imágenes en cada clase y el vocabulario para tener un conjunto de datos más completo, con diferentes prendas de vestir, fondos y diferentes tonos de piel. En cuanto al ambiente de implementación se recomienda usar fondos blancos y ropa oscura para una mayor garantía del reconocimiento tanto de las manos como de la seña que se quiera traducir. Por otra parte, también se puede implementar haciendo uso del modelo con extensión "tflite" para aplicaciones móviles y el modelo en formato JSON para sistemas web. Además, resulta como herramienta atractiva para la promoción y aprendizaje de la LSC.

FINANCIAMIENTO

Recursos propios.

AGRADECIMIENTOS

A nuestras familias y amigos por el apoyo incondicional de todas las maneras posibles a lo largo de todo el desarrollo profesional.

A la Universidad Pedagógica y Tecnológica de Colombia, y a todos nuestros educadores, en especial a nuestro tutor de trabajo de grado ingeniero, el doctor ingeniero Marco Javier Suarez Barón, por su comprensión, apoyo, enseñanzas y quien compartió sus conocimientos a lo largo de nuestra carrera.

REFERENCIAS

- [Congreso de la República de Colombia., 2020] Congreso de la República de Colombia. *Ley 2068 de 2020*. <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=172558> ↑Ver página 72
- [Google, s. f.] Google. (s. f.). *MediaPipe*. <https://google.github.io/mediapipe/solutions/hands> ↑Ver página 75
- [Halvardsson *et al.*, 2021] Halvardsson, G., Peterson, J., Soto-Valero, C. y Baudry, B. (septiembre de 2021). Interpretation of Swedish Sign Language using convolutional neural networks and transfer learning. *SN Computer Science*, 2, 207. <https://doi.org/10.1007/s42979-021-00612-w> ↑Ver página 73
- [Herazo, 2020] Herazo, J. (1 de agosto de 2020). Sign language recognition using deep learning. *TowardsDataScience.com*. <https://towardsdatascience.com/sign-language-recognition-using-deep-learning-6549268c60bd> ↑Ver página 73
- [Insor, 2020] Instituto Nacional para Sordos (Insor). (2020). *Población sorda en Boyacá: perfil territorial*. <https://www.insor.gov.co/insorlab/wp-content/uploads/2021/12/BOYACA.pdf> ↑Ver página 72
- [Insor, s. f.] Instituto Nacional para Sordos (Insor). (s. f.). *Diccionario básico de la lengua de señas colombiana*. http://www.insor.gov.co/descargar/diccionario_basico_completo.pdf ↑Ver página 74
- [Jiménez-Forero y Moreno-Mosquera, 2020] Jiménez-Forero, G. y Moreno-Mosquera, E. E. (2020). *Método automático para el reconocimiento de gestos de manos para la categorización de vocales y números en lenguaje de señas colombiano* [Trabajo de grado]. Repositorio Institucional de la Universidad Católica de Colombia. <https://hdl.handle.net/10983/22601> ↑Ver página 73
- [Llanos Mosquera, 2021] Llanos Mosquera, J. M. (2021). Una revisión sistemática sobre aula invertida y aprendizaje colaborativo apoyados en inteligencia artificial para el aprendizaje de programación. *Tecnura*, 25(69), 196-214. <https://doi.org/10.14483/22487638.16934> ↑Ver página 72

- [Medina Rojas, 2017] Medina Rojas, F. A. (2017). A quantitative and qualitative performance analysis of compressive spectral imagers. *Tecnura*, 21(52), 53-67. <https://doi.org/10.14483/udistrital.jour.tecnura.2017.2.a04> ↑Ver página 76
- [Mishra *et al.*, 2019] Mishra, S., Sinha, S., Sinha, S. y Bilgaiyan, S. (2019). Recognition of hand gestures and conversion of voice for betterment of deaf and mute people. En *Advances in Computing and Data Sciences* (pp. 46-57). Springer Singapore. https://doi.org/10.1007/978-981-13-9942-8_5 ↑Ver página 73
- [Mustafa, 2020] Mustafa, M. (04 de marzo de 2020). A study on Arabic sign language recognition for differently abled using advanced machine learning classifiers. *Journal of Ambient Intelligence and Humanized Computing*, 12, 4101-4115. <https://doi.org/10.1007/s12652-020-01790-w> ↑Ver página 74
- [Ortiz Farfán y Camargo Mendoza, 2020] Ortiz Farfán, N. y Camargo Mendoza, J. E. (2020). Computational model for sign language recognition in a Colombian context. *Tecnológicas*, 23(23), 197-232. <https://doi.org/10.22430/22565337.1585> ↑Ver página 73
- [Ortiz García, 2021] Ortiz García, C. D. (16 de julio de 2021). *Traductor de letras en lenguaje de señas con redes neuronales convolucionales* [Trabajo de grado]. Repositorio Institucional de la Universidad de los Andes. <https://repositorio.uniandes.edu.co/handle/1992/53437> ↑Ver página 73
- [Rostand *et al.*, 2019] Rostand, C., De Araújo, T., Lima, M., Veríssimo, V., De Andrade, R., Vieira, S., Santos, A., Souza Filho, G. L., Soares, M. K. y Hanael, V. (agosto de 2019). Towards an open platform for machine translation of spoken languages into sign languages. *Machine Translation*, 33, 315-348. <https://doi.org/10.1007/s10590-019-09238-5> ↑Ver página 73
- [Trejos Buriticá 2018] Trejos Buriticá, O. I. (2018). Aprovechamiento de los tipos de pensamiento matemático en el aprendizaje de la programación funcional. *Tecnura*, 22(56), 29-39. <https://doi.org/10.14483/22487638.12807> ↑Ver página 82
- [Vázquez-Enríquez *et al.*, 2021] Vázquez-Enríquez, M., Alba-Castro, J. L., Docio-Fernández, L. y Rodríguez-Banga, E. (2021). Isolated sign language recognition with multi-scale spatial-temporal graph convolutional networks. En *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp. 3457-3466). <https://doi.org/10.1109/CVPRW53098.2021.00385> ↑Ver página 74

