***Reconocimiento De Señas Con Python***

***Análisis numérico***

***Santiago Escobar Mejía***

***Santiago Elias Rodriguez Hawasly***

***Luis Fernando Posada Cano***

***Universidad EAFIT***

***Medellín***

***2021***

***Abstract***

El reconocimiento de señas por medio de la visión artificial es un tema de interés a nivel mundial con el cual se busca ayudar para una integración adecuada de la población con discapacidad auditiva, logrando disminuir la brecha de comunicación existente entre las personas con esta discapacidad y el resto de la sociedad, por lo cual se busca que no se les dificulte la interacción en sociedad. Para esto en este proyecto mediante la implementación de librerías como OpenCV, Tensorflow, mediapipe y en un lenguaje no complejo como lo es Python se desarrollo un sistema el cuál por medio del reconocimiento de patrones se reconocen las principales características emuladas en video compuestos por frames de un grupo pequeña de señas (5), la utilización de redes neuronales para el entrenamiento y el aprendizaje del sistema para el reconocimiento de estas.

Se plantea un sistema donde se ingresa un total de 300 fotos por seña a reconocer las cuales nos servirán para el entrenamiento y la validación de las diferentes señas, donde a estás fotos se les extraerá y se cambiara el espacio de color bgr por el espacio rgb, donde por medio de la red neuronal y un número de interacciones el cuál sea el óptimo se buscó lograr una precisión mayor al 0.8. En la etapa final del desarrollo del sistema nos encontramos que para tener está precisón se debe realizar la ejecución con mínimo 128 neuronas, con 30 interacciones para así reconocer las 5 señas las cuales son las letras a,e,i,o,u.

Palabras claves: visión artificial, red neuronal, machine learning, espacio de color, reconocimiento de patrones, Python, lenguaje de señas.

***índice***

***1.Planteamiento del tema……………………………………………… 5***

***1.1 Problemática.......................................................................... 5***

***1.2 Posible solución…………………………………………… 5***

***1.3 Definición de objetivos…………………………………….. 5***

***1.3.1 Objetivo general………………………………….. 5***

***1.3.2 Objetivos específicos……………………………... 5***

***2. Estado del arte………………………………………………………. 5***

***2.1 OpenCV……………………………………………………. 5***

***2.2 Redes neuronales………………………………………….. 6***

***2.3 Espacio de color RGB…………………………………….. 8***

***2.4 Espacio de color BGR…………………………………….. 9***

***2.5 Tensorflow………………………………………………… 9***

***2.6 Mediapipe…………………………………………………. 9***

***3. Marco Teórico……………………………………………………... 9***

***3.1 Machine learning………………………………………… 9***

***3.2 Reconocimiento de patrones……………………………… 10***

***3.3 Python……………………………………………………... 10***

***3.4 Lenguaje de señas………………………………………… 11***

***4. Desarrollo………………………………………………………….. 11***

***4.1 Dedos.py……………………………………………………. 11***

***4.2 RedNeuronal.py……………………………………………. 11***

***4.3 Predicción.py………………………………………………... 11***

***5. Resultados……………………………………………………………. 12***

***6. Conclusiones…………………………………………………………. 12***

***7. Referencias…………………………………………………………… 13***

***1.Planteamiento del tema:***

***1.1 Problemática***

A nivel global una gran parte de la población tienen una discapacidad auditiva o del habla por lo cual hacen uso del lenguaje de señas, lenguaje que además de este gran número de personas muy pocas personas lo dominan, por tal motivo la interacción con otras personas se les dificultad.

La única manera de interactuar con este grupo de personas es por medio del lenguaje de señas, las maneras de aprenderlo son por medio de cursos presenciales o por medio de videos u otros materiales audiovisuales que ayuden como apoyo para el aprendizaje lo cuál se vuelve tedioso y un poco complicado de entender y acceder a ellos.

***1.2 Posible solución***

La solución planteada en el desarrollo de esta investigación es la inclusión de este lenguaje de señas por medio del uso de las nuevas tecnologías donde el aprendizaje y el reconocimiento de este lenguaje sea de manera más autodidacta apoyándonos de librerías OpenCV que se pueden enlazar a sistemas de inteligencia artificial que permiten el desarrollo en Python de algoritmos de procesamiento de imagen óptimos.

***1.3 Definición de objetivos***

***1.3.1 Objetivo general***

Desarrollar un sistema de visión artificial para la detección de señas, implementando algoritmos de procesamiento de imagen, utilizando un lenguaje de programación como lo es Python y haciendo utilización de librerías OPENCV

***1.3.2 Objetivos específicos***

Los objetivos específicos de este proyecto son:

* Clasificación de las vocales con una alta precisión por medio del conocimiento del gasto neuronal.
* Conocer y aprender sobre el uso de librerias OPENCV.
* Aprender del desarrollo del entrenamiento y el autoaprendizaje de un software de inteligencia artificial.
* Entrenamiento y autoaprendizaje con una precisión mayor o igual al 80% para lograr una buena predicción.

***2. Estado del arte***

***2.1 OpenCV*** es una biblioteca de código abierto que contiene implementaciones que abarcan 2500 y más algoritmos. Es gratuita para uso comercial y la investigación bajo una licencia BSD. Se especializa en el sistema de visión artificial y machine learning.

La utilización principal va desde la detección de objetos y rostros, especialmente en campos como la seguridad, el marketing o incluso la fotografía.

Entre sus principales ventajas contamos con:

* Funciona como proveedor de infraestructura para las aplicaciones relacionadas con la visión artificial.
* Es multiplataforma, se ejecuta en sistemas operativos como Windows, Mac OS X, Linux,etc.
* Se puede utilizar con distintos lenguajes como Java, C# y Python.
* Tiene gran eficiencia por tal motivo desde su fundación solo ha tenido 3 versiones.
* Tiene una gran comunidad, lo que ayuda a su desarrollo permanente.

[5]

***2.2 Redes neuronales,*** también conocidas como redes neuronales artificiales (ANN) o redes neuronales simuladas (SNN), son un subconjunto de machine learning y están en el núcleo de los algoritmos de deep learning.

Las redes neuronales artificiales (ANN) están formadas por capas de nodos, que contienen una capa de entrada, una o varias capas ocultas y una capa de salida. Cada nodo, o neurona artificial, se conecta a otro y tiene un peso y un umbral asociados. Si la salida de un nodo individual está por encima del valor de umbral especificado, dicho nodo se activa y envía datos a la siguiente capa de la red. De lo contrario, no se pasan datos a la siguiente capa de la red.

Diagrama

Descripción generada automáticamente

*(Imagen tomada de* [*¿Qué son las redes neuronales? - España | IBM*](https://www.ibm.com/es-es/cloud/learn/neural-networks)*)*

Las redes neuronales se basan en entrenar datos para aprender y mejorar su precisión con el tiempo. No obstante, una vez que estos algoritmos de aprendizaje se ajustan de manera precisa, son potentes herramientas de informática y inteligencia artificial, lo que nos permite clasificar y agrupar los datos a una alta velocidad.

De esta manera funcionan las redes neuronales: Se debe de ver cada nodo individual como su propio modelo de regresión lineal, formado por datos de entrada, ponderaciones, un sesgo (o umbral) y una salida. La fórmula sería similar a la siguiente:

∑wixi + bias = w1x1 + w2x2 + w3x3 + bias

output = f(x) = 1 if ∑w1x1 + b> = 0; 0 if

∑w1x1 + b < 0

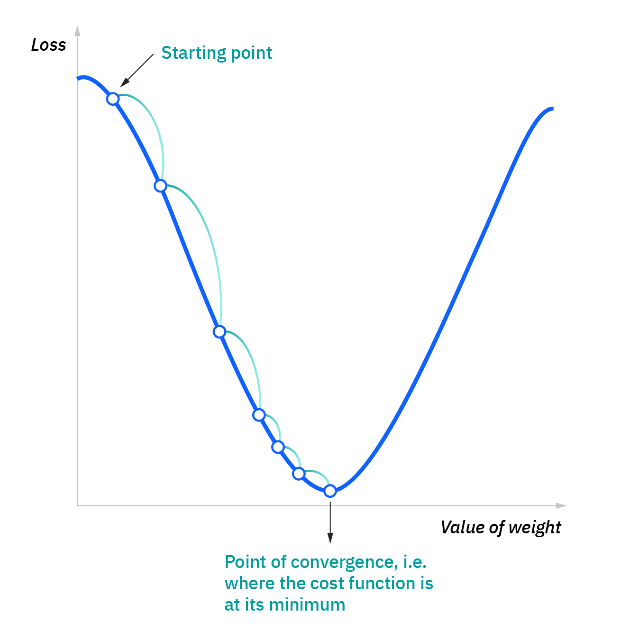
Una vez que se determina una capa de entrada, se asignan ponderaciones. Estas ponderaciones permiten determinar la importancia de cualquier variable, donde las más grandes contribuyen más significativamente a la salida en comparación con otras entradas. A continuación, todas las entradas se multiplican por sus respectivas ponderaciones y se suman. A continuación, la salida se pasa a través de una función de activación, que determina la salida. Si la salida supera un determinado umbral, activa el nodo y pasa los datos a la siguiente capa de la red. Como resultado, la salida de un nodo se convierte en la entrada del nodo siguiente. Este proceso de pasar datos de una capa a la siguiente define esta red neuronal como una red de propagación hacia delante.

Como las redes neuronales se comportan de forma similar a los árboles de decisiones, los datos en cascada de un nodo a otro con x valores entre 0 y 1 reducirán el impacto de cualquier cambio de una única variable en la salida de cualquier nodo y, posteriormente, la salida de la red neuronal.

A medida que empezamos a pensar en casos de uso más prácticos de redes neuronales, como el reconocimiento de imágenes o la clasificación, podremos aprovechar el aprendizaje supervisado o los conjuntos de datos etiquetados para entrenar el algoritmo. Cuando entrenamos un modelo, nos interesa evaluar su precisión usando una función de coste (o pérdida). Esto también se conoce comúnmente como el error cuadrático medio (MSE). En la siguiente ecuación:

𝐶𝑜𝑠𝑡 𝐹𝑢𝑛𝑐𝑡𝑖𝑜𝑛 = 𝑀𝑆𝐸 =1/2𝑚 ∑129\_(𝑖 =1)^𝑚▒(𝑦 ̂^((𝑖) )−𝑦^((𝑖) ) )^2

En última instancia, el objetivo es minimizar nuestra función de coste para garantizar que el ajuste es correcto para cualquier observación. A medida que el modelo ajusta sus ponderaciones y sesgos, utiliza la función de coste y el aprendizaje de refuerzo para alcanzar el punto de convergencia, o el mínimo local. El proceso con el que el algoritmo ajusta sus ponderaciones es con un descenso de gradiente, lo que permite al modelo determinar la dirección a seguir para reducir los errores (o minimizar la función de coste). Con cada ejemplo de entrenamiento, los parámetros del modelo se ajustan para converger gradualmente al mínimo.



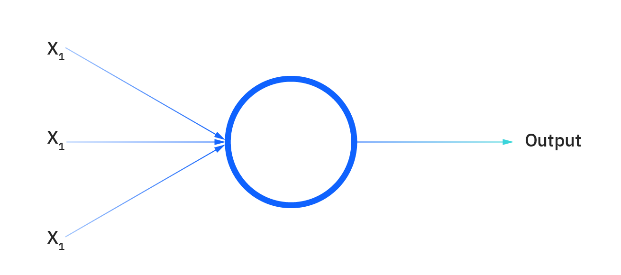
*(Imagen tomada de* [*¿Qué son las redes neuronales? - España | IBM*](https://www.ibm.com/es-es/cloud/learn/neural-networks)*)*

La mayoría de las redes neuronales profundas son de propagación hacia delante, lo que significa que fluyen en una sola dirección, de la entrada a la salida. Sin embargo, también puede entrenar su modelo mediante la retropropagación; es decir, moverse en la dirección opuesta, de la salida a la entrada. La retropropagación permite calcular y atribuir el error asociado a cada neurona, de forma que podemos ajustar y adecuar los parámetros de los modelos según corresponda.

Tipos de redes neuronales:

la siguiente lista sería representativa de los tipos más comunes de redes neuronales que encontrará en sus casos de uso:

El perceptrón es la red neuronal más antigua, creada por Frank Rosenblatt en 1958. Tiene una sola neurona y es la forma más simple de una red neuronal:



*(Imagen tomada de* [*¿Qué son las redes neuronales? - España | IBM*](https://www.ibm.com/es-es/cloud/learn/neural-networks)*)*

Las redes neuronales de propagación hacia delante o perceptrones multicapa (MLP)

Están formadas por una capa de entrada, una capa o varias capas ocultas y una capa de salida. Aunque estas redes neuronales también se conocen como MLP, es importante tener en cuenta que en realidad están formadas por neuronas sigmoides, no perceptrones, ya que la mayoría de los problemas del mundo real son no lineales. Normalmente, se proporcionan datos a estos modelos para entrenarlos. Son la base para la visión artificial, el procesamiento del lenguaje natural y otras redes neuronales.

Las redes neuronales convolucionales (CNN) son similares a las redes de propagación hacia delante, pero normalmente se utilizan para el reconocimiento de imágenes, el reconocimiento de patrones y/o la visión artificial. Estas redes aprovechan los principios del álgebra lineal, particularmente la multiplicación de matriz, para identificar patrones dentro de una imagen.

Las redes neuronales recurrentes (RNN) se identifican por sus bucles de retroalimentación. Estos algoritmos de aprendizaje se utilizan principalmente con datos de series temporales para hacer predicciones sobre resultados futuros, por ejemplo, predicciones del mercado de acciones o previsiones de ventas.

[*4* ]

***2.3 Espacio de color RGB*** se basa en la combinación de tres señales de luminancia cromática distinta: rojo, verde y azul (Red, Green, Blue). La manera más intuitiva de conseguir un color concreto es determinar la cantidad de color rojo, verde y azul que se necesita combinar, para ello se realiza la suma aritmética de las componentes: X = R + G + B.

Si tomamos en cuenta que cada una de las componentes tiene variaciones de tonalidad que van desde el 0 al 255 (28), entonces el cubo posee (28)3 = 16,777,216 colores representados vectorialmente; de esta manera podemos distinguir los extremos como el rojo (255,0,0), verde (0,255,0), azul (0,0,255), negro (0,0,0) y blanco (255,255,255) [6].

Imagen que contiene Diagrama

Descripción generada automáticamente

***(****Imagen tomada de* [*UCC0990\_01.pdf (pucv.cl)*](http://opac.pucv.cl/pucv_txt/txt-0500/UCC0990_01.pdf) ***)***

***2.4 Espacio de color BGR*** es el formato en el que OpenCV almacena el color, aunque el espacio de color predeterminado de OpenCV es RGB. Este es un modelo de color aditivo donde las diferentes intensidades de Azul, Verde y Rojo dan distintas tonalidades de color.

***2.5 Tensorflow*** es una plataforma de código abierto de extremo a extremo para el aprendizaje automático. Cuenta con un ecosistema integral y flexible de herramientas, bibliotecas y recursos de la comunidad que permite que los investigadores innoven con el aprendizaje automático y los desarrolladores creen e implementen aplicaciones con tecnología de AA fácilmente.[7]

***2.6 Mediapipe*** es un paquete de Python prediseñado en PyPI. También proporciona herramientas para que los usuarios creen sus propias soluciones. El paquete MediaPipe Python está disponible en PyPI para Linux, macOS y Windows.

Cosas que podemos realizar con la ayuda de MediaPipe

* Detección de rostros y malla de rostros
* Pose y detección holística
* Detección y seguimiento de objetos
* Reconocimiento de manos
* plantear estimación
* Conteo de dedos
* Ajuste de volumen

[8]

***3. Marco Teórico*** (Machine learning, Pattern recognition, Python y otras tecnologías,)

***3.1 Machine learning*** es una disciplina de la inteligencia artificial, mediante la identificación de patrones complejos entre millones de datos, se crea un sistema que aprende este contexto creado por esos patrones. Un algoritmo es el que realmente entre comillas aprende, revisando datos y prediciendo comportamientos futuros, frente a esto notamos 2 faces en el funcionamiento del machine learning, la primera es el aprendizaje, en la que el algoritmo lee e identifica los patrones de los datos y una segunda en la que el algoritmo arroja una predicción o posible resultado a partir de los datos anteriores [1].

La fase de aprendizaje al alimentarse con datos será mejor ejecutada mientras mayor cantidad de datos se le ingresen, es decir no es lo mismo entrenar un algoritmo con diez mil datos que con millones, ya que en este caso la cantidad acompañada de la calidad de los datos va a hacer la diferencia y a generar un mejor entrenamiento en el algoritmo. Por su parte resultado o predicción puede ser acertado o no, pero mientras mejor entrenado sea el algoritmo más factores tendrá en cuenta por lo que su calidad también depende en su origen de los datos.

***3.2 Reconocimiento de Patrones*** (Pattern Recognition) es el uso de modelos computacionales para encontrar patrones de comportamiento en datos, imágenes y cualquier otro tipo de entradas. Este tipo de modelos han sido utilizado en una amplia variedad de aplicaciones, como la visión por computadora, análisis estadístico de datos, análisis médicos, análisis de imágenes, reconocimiento de voz, entre otros [2]. Esta versatilidad lo ha convertido en uno de los modelos computacionales más utilizados como base para todo tipo proyectos grandes y ambiciosos como Boston Dynamics [3], una compañía que investiga y desarrolla robots con grandes capacidades en cuanto autonomía, desempeño, resolución de problema y reconocimiento del terreno, este ultimo permitiendo que estos robots sean capaces de calcular autónomamente como moverse en el entorno manteniendo su estabilidad y orientación, siendo esto posible gracias a al análisis de imágenes y el reconocimiento de patrones.

La tecnología detrás de la gran eficacia del reconocimiento de patrones es el Machine Learning, por esto mismo sigue el mismo patrón de desarrollo con dos fases, la primera para el entrenamiento de modelo y la segunda en la que el algoritmo arroja las predicciones. Es importante hacer un énfasis en la calidad y la buena elección de los datos de entrenamiento para que de los mejores resultados posibles.

***3.3 Python*** es un lenguaje de programación de código abierto, orientado a objetos, muy simple y fácil de entender. Tiene una sintaxis sencilla que cuenta con una vasta biblioteca de herramientas, que hacen de Python un lenguaje de programación único. Una de las ventajas principales de aprender Python es la posibilidad de crear un código con gran legibilidad, que ahorra tiempo y recursos, lo que facilita su comprensión e implementación.

La utilización de Python para proyectos de machine learning y de inteligencia artificial es muy común y útil, es un lenguaje de escritura rápido, escalable, robusta y de código abierto, ventajas que hacen de Python un aliado perfecto para la Inteligencia Artificial. Permite plasmar ideas complejas con unas pocas líneas de código, lo que no es posible con otros lenguajes.

Existen bibliotecas como «Keras» y «TensorFlow”, que contienen mucha información sobre las funcionalidades del aprendizaje automático. Además, existen bibliotecas proporcionadas por Python, que se usan mucho en los algoritmos AI como Scikitl, una biblioteca gratuita de aprendizaje automático que presenta varios algoritmos de regresión, clasificación y agrupamiento. También con bibliotecas de procesamiento de datos como ‘Pydoop‘ , que son de gran ayuda para los profesionales, ya que puede escribir un código de MapReduce en Python y procesar los datos en el clúster HDFS.

Python junto a NumPy, scikit-learn, iPython Notebook y matplotlib, son la base para iniciar un proyecto de IA.

***3.4 Lenguaje de señas*** es el tipo de comunicación que se da entre las personas que sufren de sordera. Al igual que otros tipos de lenguajes, posee una serie de reglas lingüísticas que han sido establecidas con el objetivo de poder ayudar a resolver las necesidades de comunicación que existen entre los seres humanos. Esta lengua está al alcance, actualmente, de todas las personas con problemas auditivos y gracias a ellas, el proceso de comunicación logra establecerse de forma efectiva y certera.

El lenguaje de señas es una forma de expresión y de comunicación basada en la configuración gestual, espacial y en la percepción visual o táctil en las personas que padecen de hipoacusia o sordera por medio de la cual logran establecer un adecuado canal de comunicación con el entorno social.

***4. Desarrollo:***

Se elaboran 3 clases principales:

***4.1 Dedos.py***: En este se ejecuta los métodos para el entrenamiento y la validación la cual se hace 5 veces (las 5 vocales) por el cual se toman 300 fotos por cada una, este las guarda en una carpeta dentro del programa, se utiliza varias herramientas de mediapipe para el reconocimiento de la mano; Se cambia el color de bgr a rgb. Se hace una copia para el dibujado del rectángulo, se procesan las fotos. Se busca el punto central de la palma para así realizar un nuevo rectángulo donde quedaría solo la imagen de la mano. Se corta el video siempre en 200\*200 para evitarnos problemas con otro tipo de resoluciones.

***4.2 RedNeuronal.py*** Se reciben las fotos y se procesan para hacer las interacciones necesarias para una mejor precisión, se tienes 3 capas de redes neuronales una de 32,64 y 128; Se configuran las imágenes haciendo un reescalado a los colores y la posición de la imagen, se añade la red neuronal a utilizar(sequencial), se añade los tipos de perdidas a tener encuenta, las métricas y parámetros de configuración, guarda el modelo y los pesos relacionados con la red.

**4.3 Predicción.py** se carga el modelo y los pesos, lista los nombres para las etiquetas, declara las variables y para cada frame se hace lo mismo que en el entrenamiento haciendo el rectángulo para tener la imagen 200\*200 y así hacer la predicción sobre esa imagen, en el método resultado se sacó el mayor del array para darnos nuestra respuesta. Todo esto para cada seña (las 5 vocales).

Diagrama, nombre de la empresa

Descripción generada automáticamente

***(****Desarrollo del sistema****)***

***5. Resultados:***

Se adjuntan imágenes sobre lo realizado:

***Texto

Descripción generada automáticamente***

*Carpeta donde se guardan las imágenes para el entrenamiento.*

*Imagen que contiene pantalla, oscuro, monitor, tabla

Descripción generada automáticamente*

*Foto con la letra A*

*Texto

Descripción generada automáticamente*

*Foto del entrenamiento*

*Una foto de un hombre con lentes

Descripción generada automáticamente con confianza media*

*Reconocimiento de la letra A*

***6. Conclusiones:***

* *En el desarrollo del sistema las redes neuronales la trabajamos con 2*^N con esto nos dimos cuentas que para que reconozca una sola seña se debe hacer a partir de 2^7, a lo cual hicimos un grupo de pruebas para saber que cantidad de neuronas es necesario utilizar para tener un entrenamiento optimo logrando una mejor precisión.

Tabla

Descripción generada automáticamente

*Tabla de neuronas recomendadas para la detección de señales*

*Grafico de la tabla de neuronas recomendadas para la detección de señales*

* La precisión del modelo tiene una relación proporcional con el número de interacciones realizadas en el entrenamiento ya que a mayor interacciones mayor va ser la precisión de este, para tener un sistema más completo es recomendable realizarlo con mayor número de fotos teniendo en cuenta que el número de imágenes con la cual se realizo este proyecto fue de 300 por vocal, por otro lado se recomienda que se tenga una buena variadad de fotos donde cambie el color de la piel, el fondo, el grado de inclinación, etc. Esto con el fin de lograr una precisión increíble del sistema.
* El rendimiento del computador también es muy importante para que el sistema se ejecute de una buena manera ya que el entrenamiento puede tomar mucho tiempo teniendo en cuenta el número de neuronas a utilizar, la cantidad de fotos para el entrenamiento y por ultimo el número de interacciones.
* Este tipo de sistemas son aportes muy importantes para la inclusión en la sociedad de personas discapacitadas, como también puede ser de gran apoyo para el estudio del lenguaje de señas volviendo su enseñanza más interactiva.

***7. Referencias:***

* [1]"¿Qué es Machine Learning? – Cleverdata". Cleverdata – Hacemos realidad la Excelencia Operacional a través del Decision Engineering. https://cleverdata.io/que-es-machine-learning-big-data/ (accedido el 9 de octubre de 2021).
* [2] "Understanding Pattern Recognition in Machine Learning". Engineering Education (EngEd) Program | Section. https://www.section.io/engineering-education/understanding-pattern-recognition-in-machine-learning/ (accedido el 9 de octubre de 2021).
* [3] "About". Boston Dynamics. https://www.bostondynamics.com/about (accedido el 9 de octubre de 2021).
* "¿Cómo se relaciona la inteligencia artificial (IA) con Python? - ▷ Cursos de Programación de 0 a Experto © Garantizados". https://unipython.com/como-se-relaciona-la-inteligencia-artificial-ia-con-python/ (accedido el 11 de octubre de 2021).
* "Python, los 5 usos más importantes de este lenguaje de programación". El Blog de Akademus. https://www.akademus.es/blog/programacion/principales-usos-python/ (accedido el 11 de octubre de 2021)
* I. C. T. López, “Sistema para el aprendizaje del lenguaje de señas colombiano usando visión por computador”, Edu.co. [En línea]. Disponible en: https://ciencia.lasalle.edu.co/cgi/viewcontent.cgi?article=1158&context=ing\_automatizacion. [Consultado: 21-nov-2021].
* [6]C. Gonzales, J. Yimes, “Sistema de reconocimiento gestual de lengua de señas chilena mediante cámara digital”, [Pucv.cl. [En línea]. Disponible en: http://opac.pucv.cl/pucv\_txt/txt-0500/UCC0990\_01.pdf. [Consultado: 21-nov-2021].
* Y. Muñoz, L. Moreno,

“Implementación de un algoritmo para la clasificación automática de lenguaje de señas colombiano en video usando aprendizaje profundo”, Edu.co. [En línea]. Disponible en: https://repository.ucatolica.edu.co/bitstream/10983/24980/1/Proyecto%20de%20grado.pdf. [Consultado: 21-nov-2021].

* J. A. Etxeberria, “Algoritmo de reconocimiento de forma y color para una plataforma robótica”, Ehu.eus. [En línea]. Disponible en: https://www.ehu.eus/documents/1545039/1570316/10jaristondo.pdf. [Consultado: 21-nov-2021].
* O. y. Silva-Martínez, “Análisis de algoritmos para reconocimiento de imágenes digitales”, Ecorfan.org. [En línea]. Disponible en: https://www.ecorfan.org/spain/researchjournals/Sistemas\_Computacionales\_y\_TICs/vol2num5/Revista\_de\_Sistemas\_Computacionales\_y\_TIC%60S\_V2\_N5\_11.pdf. [Consultado: 21-nov-2021].
* “Home - OpenCV”, Opencv.org, 09-feb-2021. [En línea]. Disponible en: https://opencv.org/. [Consultado: 21-nov-2021].
* [5]“¿Qué es OpenCV y para qué sirve? 👁️ [2021]”, Crehana.com. [En línea]. Disponible en: https://www.crehana.com/mx/blog/desarrollo-web/que-es-opencv/. [Consultado: 21-nov-2021].
* [4]“¿Qué son las redes neuronales?”, Ibm.com. [En línea]. Disponible en: https://www.ibm.com/es-es/cloud/learn/neural-networks. [Consultado: 21-nov-2021].
* [7]“TensorFlow”, Tensorflow.org. [En línea]. Disponible en: https://www.tensorflow.org/?hl=es. [Consultado: 21-nov-2021].
* [8]“MediaPipe”, Mediapipe.dev. [En línea]. Disponible en: https://mediapipe.dev/. [Consultado: 21-nov-2021].