AsorHuil aplicativo web para el reconocimiento del abecedario de señas colombiano

AsorHuil web application for the recognition of the alphabet of Colombian signs

[1]Daniela Alarcón Sepúlveda, [2]José Luis Baquero Mora

[1,2]Universidad Surcolombiana

Recibido: 15/12/2021 Aprobado: 20/01/2022

Resumen

El lenguaje de señas colombiano es utilizado por personas sordas para su propia comunicación. Consiste en movimientos y expresiones de diferentes partes del cuerpo, especialmente las manos. Actualmente, en Colombia, hay mucha falta de tecnología para el aprendizaje y la interpretación de este lenguaje, por tanto, es un compromiso social llevar a cabo iniciativas encaminadas a mejorar la calidad de vida de la población sorda del país. Este artículo presenta el proceso de diseño e implementación de un sistema de reconocimiento de gestos no móvil con entorno web mediante las tecnologías de Spring Tools y Angularjs y con el microservicio en el lenguaje python que implementa una red neuronal convolucional (Convolutional Neural Network) que utiliza las librerías de Keras y Tensorflow para el reconocimiento de imágenes. El Sistema permite visualizar las imágenes adquiridas y traducirlas al lenguaje de señas colombiano. Además, tiene la capacidad de consultar multimedia de ayuda y visualizar libros o productos que se ofrecen en la Fundación AsorHuil de la ciudad de Neiva-Huila (Colombia).

Palabras clave: lenguaje de señas colombiano, Python, red neuronal convolucional, Keras, Tensorflow, reconocimiento de imágenes.

Abstract

Colombian sign language is used by deaf people for their own communication. It consists of movements and expressions of different parts of the body, especially the hands. Currently, in Colombia, there is a great lack of technology for learning and interpretation, therefore, it is a social commitment to carry out initiatives aimed at improving the quality of life of the country's deaf population. This article presents the process of designing and implementing a non-mobile gesture recognition system with a web environment using Spring Tools and Angularjs technologies and with the microservice in the python language that implements a convolutional neural network (Convolutional Neural Network)

Citación: Baquero Mora, J. L., & Alarcón Sepulveda, D. . (2022). Abecedario lengua de señas Aplicativo web reconocimiento del abecedario de lenguaje de señas: Asorhuil . Publicaciones E Investigación, 16(1). https://doi.org/10.22490/25394088.5638

[1] Facultad de Ingeniería, Programa de Software, U20181166335@usco.edu.co
[2] Facultad de Ingeniería, Programa de Software, U20181167874@usco.edu.co

https://doi.10.22490/25394088.5638

that uses Keras and Tensorflow libraries for image recognition. The System allows visualizing the acquired images and translating them into Colombian sign language. In addition, it has the ability to consult multimedia help and view books or products offered by the AsorHuil Foundation in the city of Neiva- Huila.

Key words: *Colombian Sign Language, Python, Convolutional Neural Network, Keras, Tensorflow, Image Recognition.*

1. Introducción

En la Asociación de Sordos AsorHuil de Neiva-Huila (Colombia) se emplean cursos de lenguaje de signos o señas para discapacitados auditivos. En este tipo de enseñanza, los aprendices enfrentan una curva de aprendizaje muy alta. Los tutores son escasos y el proceso de evaluación es muy complejo y lento. Por otro lado, la comunidad sufre mucho por la discriminación y rechazo de la sociedad, en la cual enfrentan a una patente falta de accesibilidad en lo que se refiere a los planes de formación, lo que limita a la igualdad de oportunidades en el mercado laboral. Principalmente, son agredidos físicamente o despedidos injustificadamente de un empleo; no pueden hacer nada por falta de intérpretes, cuando el gobierno debe ofrecer una atención de calidad a la comunidad.

¿Cómo se puede desarrollar una aplicación web para cualquier sistema operativo que permita ayudar al aprendizaje del lenguaje de señas en la Asociación de Sordos AsorHuil de la ciudad de Neiva?

Además ¿Cómo se puede identificar el sistema de reconocimiento del abecedario de lengua de señas colombianomediante la construcción de un dataset para ayudar al aprendizaje de la lengua de señas colombiana?

1.1 Definición de objetivos

1.1.1 Objetivo general

Diseñar y desarrollar una aplicación web que permita enseñar a la comunidad de manera dinámica y muy práctica el lenguaje de señas a través de un sistema de reconocimiento del abecedario de lenguaje de signos colombiano que implementa algoritmos de procesamiento y segmentación de imágenes, utilizando librerías de OpenCV y para su entrenamiento se utiliza las librerías de Tensorflow, a través de un lenguaje de programación y de una cámara digital.

1.1.2 Objetivos específicos

Los objetivos específicos del proyecto son:

1. Aprender y comprender el uso de OpenCV y Tensorflow, implementado mediante algoritmos de reconocimiento de patrones.
2. Aprender a usar máquinas de soporte de vectores para entrenamiento y software de inteligencia artificial autodidacta, en nuestro caso, reconocimiento de patrones, proporcionado por Tensorflow y Keras.
3. Entrenar mediante el software de aprendizaje las letras del abecedario colombiano en el lenguaje de señas.
4. Analizar los resultados obtenidos en la investigación y proponer una solución web que se ajuste a las necesidades que se determinaron.
5. Conseguir una tasa de error baja.

1.2 Justificación e importancia

Es importante que la curva de aprendizaje de la lengua de señas disminuya a un gran porcentaje para una mejor calidad de vida.

En 1996, el congreso de Colombia reconoció el lenguaje de señas como un idioma oficial por medio de la ley 324, esta permite que las personas que tienen discapacidad auditiva exijan una educación inclusiva y de alta calidad.

Por lo mencionado anteriormente, se necesita una innovación y automatización de los procesos de

enseñanza y evaluación de los procesos que se implementan en la asociación de sordos AsorHuil. Con el fin de que los aprendices mejoren su comunicación, sus relaciones sociales, sus oportunidades laborales y personales.

1.3 Marco teórico

1.3.1 Prathamesh Timse, Pranav Aggarwal, Prakhar Sinha, Neel Vora y Ortega

Tomando como referencia dos trabajos hechos con la librería de OpenCv se obtiene una aproximación del tiempo (Timse, Aggarwal, Sinha & Neel Vora, 2014) y muestras (Ortega, 2013) que se requieren para entrenar el sistema y su porcentaje de precisión las cuales se muestran en la Tabla 1.

Tabla 1.
Porcentaje de precisión

Esta imagen esta muy pixelada. Enviar la tabla editable

Muestras	Imágenes		
15	Números	44	88%
	Letras	42	84%
30	Números	46	92%
	Letras	45	90%
50	Números	44	88%
	Letras	40	80%

1.3.2 Diseño e implementación de un sistema traductor de lenguaje de señas de manos a un lenguaje de texto mediante visión artificial en un ambiente controlado (Chiguano, 2018). El artículo está basado en realizar un sistema que traduce de lenguaje de señas a lenguaje de texto usando la visión artificial. Este proyecto ofrece un entrenador, para que aprendan cada uno de los símbolos y se adquiera una habilidad para el programa.

El sistema adquiere la imagen que pasa por un procesamiento digital de imágenes y por último se realiza la traducción.

El programa fue desarrollado en Labview 2009 incluyendo su toolkit de visión artificial. En el procesamiento digital de imágenes se aplicaron algunos filtros y operaciones morfológicas para resaltar las características de la imagen y eliminar información innecesaria como ruido. También se eliminaron objetos extraños en la imagen mediante un recortado del área de interés. Con la ayuda del Vision Assitant de Labview se elaboraron las bases de datos, para llevar a cabo la comparación con la imagen recortada y de esta manera asignar la clase (letra) correspondiente a cada imagen. Con la clase asignada se forma el texto que se muestra en forma escrita en la pantalla o a su vez se puede enviar a un documento de Word, además es posible reproducir en audio el texto formado con la ayuda de la herramienta "texto a voz" de Windows.

Tabla 2.
Resultados- vecino más cercano

Esta imagen esta muy pixelada. Enviar la tabla editable

K-Vecino más Cercano (in				
Letra	Euclidean	Maximum	Sum	Total
W	500,00	No reconoce	500,00	500,05
X	500,00	500,00	1000,00	749,94
Y	1000,00	1000,00	1000,00	749,94

Los métodos de clasificación de Vecino más Cercano, K- Vecino más Cercano y Mínima Distancia; de los cuales se concluye: para aplicaciones donde existe ruido el método el vecino más cercano es poco efectivo, en aplicaciones con mucho ruido y cuyos patrones tiene variaciones muy pequeñas entre sí es mejor usar el método de la mínima distancia. Por lo tanto, para esta aplicación con presencia de ruido y cuyos patrones en su mayoría están bien diferenciados se usa el método del K-Vecino más Cercano.

1.3.3 Signame es una aplicación que ayuda a desarrollar las habilidades comunicativas.

Proporciona videos populares, rápidos y fáciles que muestran las diferentes etiquetas utilizadas en la comunicación. Estos vídeos han sido elaborados por los expertos del Centro María Cordentura en España a partir de la experiencia diaria y la dilatada trayectoria del centro.

Se basa principalmente en la lengua de signos española, pero se han citado varias fuentes. Debido a la complejidad estructural de muchas de las señales que deben hacer los niños, algunas se han adaptado simplificándolas o eligiendo alternativas.

2. Materiales y métodos

La Tabla 3 enumera los materiales utilizados para hacer el sistema. Contiene las respectivas definiciones para una mejor comprensión.

Tabla 3.
Materiales para el desarrollo del sistema

Materiales	Descripción
Open CV	Provee infraestructura para el sistema. La librería tiene más de 2.500 algoritmos, que incluye algoritmos de machine learning.
TensorFlow	Plataforma para construir y entrenar redes neuronales que permitan la detección y descifrado de patrones y correlaciones, similar al aprendizaje y razonamiento que utilizan los humanos.
Keras	Framework de alto nivel para el aprendizaje, facilitar un proceso de entrenar modelos de clasificación de imágenes.
Python	Lenguaje de programación que se implementa el algoritmo en el sistema.

Fuente: Alarcón & Baquero (2021)

2.1 Obtención y creación del dataset

El dataset se ha creado con 5.250 imágenes, clasificadas en 21 carpetas correspondientes a cada letra del abecedario excepto "g, h, j, ñ, s y z". La fórmula empleada fue L∈R^(Hxw) donde H es el tamaño de las filas H=80 y N es el tamaño de las columnas N=80.

2.2 Preprocesamiento

Los datos se estandarizan y se limpian para aumentar su volumen de utilidades, antes de aplicar transformaciones a los datos incluidos en los algoritmos. Permite convertir los datos sin procesar en un conjunto limpio de datos.

Contornos con algoritmo Canny: el algoritmo de detección de bordes Canny propone un método de localización de bordes de acuerdo con dos criterios para bordes arbitrarios. Estos criterios son denominados "Signal- to-noise ratio" (SRN) y localization. Un paso crucial para el algoritmo es capturar ambos criterios intuitivos de forma matemática. En la siguiente ecuación se representa el modelo de SRN como primer criterio de localización.

Ecuación 1. Ecuación modelo SRN

$$SNR = \frac{\left|\int_{-w}^{+w}(G-x)f(x)\,dx\right|}{n_0\sqrt{\int_{-w}^{+w}f^2(x)dx}}$$

Todas las formulas estan muy pixeladas, enviar imagen en buena calidad

Con este criterio, se localizan los bordes de los objetos con la presencia de ruido utilizando como operador de respuesta la f(x). De manera consecutiva, se emplea el segundo criterio de localización (localization) de bordes que es representado con la siguiente ecuación.

Ecuación 2. Ecuación de localización

$$Localization = \frac{\left|\int_{-w}^{+w}G'(-x)f'(x)dx\right|}{n_0\sqrt{\int_{-w}^{+w}f^2(x)dx}}$$

En síntesis, el modelo general se basa en un criterio de especificación y localización matemática, usando una optimización numérica para encontrar los operadores óptimos de los bordes inferiores y superiores de la imagen, por lo que, excluye los pixeles encontrados dentro y fuera de los bordes, permitiendo definir la forma geométrica del conjunto de objetos en la imagen.

Algoritmo de detección automática de umbral Otsu: es un algoritmo que realiza una evaluación sobre la factibilidad de un umbral en base a las condiciones de intensidad luminosa que presenta una imagen; es decir, analiza de manera estadística cada imagen de entrada (en niveles de grises) y retorna de manera automática el valor de umbral óptimo para una imagen.

Para evaluar la "efectividad" de un umbral, se utilizan las siguientes medidas de separación de clases, representada en las siguientes ecuaciones:

Ecuación 3. Ecuaciones para evaluar la efectividad de un umbral

$$\sigma_B^2 = w_0(\mu_0\mu_T)^2 + w_1(\mu_1-\mu_T)^2$$

$$\sigma_B^2 = w_0w_1(\mu_1-\mu_T)^2$$

$$\sigma_w^2 + \sigma_B^2 = \sigma_T^2$$

$$\sigma_B^2(K^*) = \max\sigma_B^2(k)$$

Donde K es el nivel de umbral óptimo al maximizar la última ecuación; OB2 la función del nivel de umbral; por último, u y w las probabilidades de ocurrencia de las clases. Se asume entonces, que el valor de K es el nivel de gris adecuado para la imagen que se procesa.

Algoritmo de detección de bordes CannySu. Consiste en analizar la imagen dada como entrada, con el algoritmo Otsu para calcular el valor de umbral que, de acuerdo con su histograma, sea el adecuado para una mejor separabilidad de clases. Este valor de umbral es proporcionado por las ecuaciones que utilizamos en el algoritmo de Otsu, modificando el operador de respuesta f(x) de la ecuación de la Figura 19 del detector Canny. De tal forma que la localización en los bordes de las imágenes, incluso en presencia de ruido, sea realizado de una manera adecuada.

Experimentación: la metodología seguida para la evaluación del algoritmo de SR y CannySu se realizó como lo muestra el diagrama de bloques en la Imagen 1:

- Se obtiene una imagen con la que se desea trabajar.
- Se evalúa si la imagen está en escala de grises; de ser válida entrará directamente al proceso de SR y detección de bordes, en caso contrario, se transformará la imagen a escala de grises para luego ser procesada por el sistema.
- Sobre la imagen validada, se emplea el algoritmo de SR para mejorar sus características.
- Posteriormente, esta imagen es analizada con el algoritmo CannySu.
- Finalmente, se obtiene como salida una imagen con bordes mejor definidos

Imagen 1. Diagrama de bloques de la experimentación

Esta imagen esta muy pixelada. no se lee nada, enviar en buena calidad

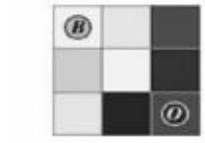
2.3 Detección de la mano

Detección del fondo y mano por separados. El principal problema del reconocimiento con cámaras digitales se encuentra en el fondo irregular de la imagen. Una forma de resolver este problema es detectar el fondo antes de enfocar la mano. Por lo tanto, cuando se detecta una mano, se calculará la diferencia entre el fondo y la parte inferior de la mano, y el resultado será una mano sin fondo.

Este método es teóricamente efectivo, pero hay algunos factores que afectan y pueden afectar los resultados; que incluyen luz, movimiento de la cámara y movimiento de la mano.

Algoritmo GrabCut. El algoritmo lo segmenta de forma iterativa para obtener el mejor resultado. Sin embargo, en algunos casos, la segmentación puede no estar bien. Por ejemplo, puede haberse marcado alguna región de primer plano como fondo y viceversa. En ese caso, el usuario debe hacer retoques finos. Sólo debe dar algunos trazos en las imágenes donde hay algunos resultados defectuosos.

Imagen 2. Proceso del algoritmo GrabCut en una imagen



(a) Image with seeds.

Esta imagen esta muy pixelada.

Técnica de Histogramas de Gradientes Orientados (HOG–Histograms of Oriented Gradients). Captan mejor las propiedades de las manos, como el contorno, los dedos, así como algunos gradientes sutiles en la palma. El gradiente de la imagen puede calcularse de diferentes maneras. Una de ellas es convirtiendo primero, la imagen a una imagen en escala de grises, para luego calcular el gradiente. Otra manera es calculando directamente el gradiente de la imagen a color tomando en cuenta los tres canales de color y utilizando el enfoque de gradiente de color basado en vector o gradiente vectorial, este último método ha demostrado una mejor relación señal a ruido que el gradiente escalar.

Los métodos que se van a desarrollar incluyen:

- Características basadas en gradientes, específicamente HOG features, ya que codifican mejor las propiedades de las manos.
- Bloques descriptores de diferentes tamaños y proporciones, ya que tienen mayor potencial para capturar patrones útiles del objeto.
- Histogramas integrales, para calcular los features de manera eficiente.
- Un clasificador dispuesto en cascada, que permite enfocar la atención del detector en regiones "prometedoras" de la imagen, y descartar patrones negativos fáciles de manera eficiente.

Detección de la letra mediante el threshold tamaño de áreas aun en análisis. Este algoritmo es una extensión de la segmentación connected-component, dentro del marco de algoritmos de level sets. El paradigma de la segmentación por level sets consiste en un método numérico para controlar la evolución de contornos y superficies. En lugar de manipular el contorno directamente, el contorno se introduce como la curva de nivel cero dentro de una función de orden superior llamada level-set function, $\psi(X,t)$. Esta función se hace evolucionar bajo el control de una ecuación diferencial. En cualquier momento puede obtenerse el contorno envolvente extrayendo la curva de nivel cero $\Gamma((X),t) = \{\psi(X,t) = 0\}$ a partir de la salida, como muestra la Imagen 3.

Imagen 3. Concepto de curva cero en un conjunto de level sets (Anónimo, 2019)

2.4 Reconocimiento de gestos estáticos

Los gestos estáticos son aquellos en los que no se tienen en cuenta los cambios en el tiempo.

Semafóricos (semaphoric): grupo de gestos que se utilizan para transmitir significados específicos a partir de la postura del gesto. Formalmente, los enfoques semafóricos pueden ser referidos como "comunicativos" y que sirven como un universo de símbolos para comunicar a la máquina. Se distinguen tres tipos de gestos semafóricos: estáticos, dinámicos y de "golpe" (stroke). El primero se refiere a una postura específica de la mano, como por ejemplo el pulgar hacia arriba para indicar aprobación.

Simbólicos (iconic): utilizados para demostrar la forma, el tamaño o la curvatura de un objeto. Los gestos simbólicos se pueden dividir en estáticos compuesto por posturas de la(s) mano(s), como por ejemplo un rectángulo formado por los pulgares y los índices de ambas manos.

Según lo anterior, se implementará y se trabajará el proyecto con las librerías de Tensorflow y Keras.

2.5 Entrenamiento

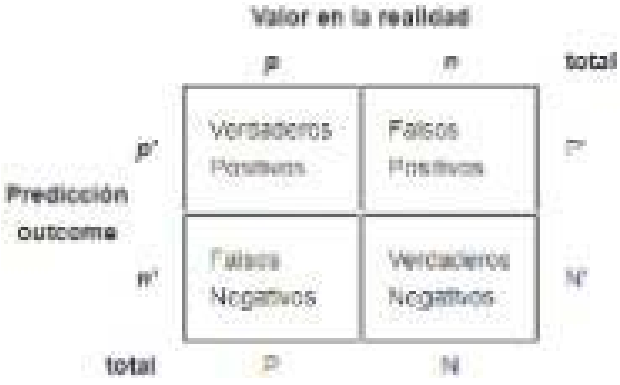
El proceso de entrenamiento de basa en realizar el gesto, guardarlo y luego entrenar; en la cual el sistema guarda las coordenadas de la mano en un archivo en una carpeta específica.

Curva ROC. Es una representación gráfica de la relación entre la sensibilidad y la especificidad del sistema clasificador binario cuando cambia el umbral de discriminación.

NOTA GENERAL: Estas imagenes estan todas muy pixelada.

Otra interpretación de este gráfico, es la representación de la razón o proporción de verdaderos positivos (VPR = Razón de Verdaderos Positivos) frente a la razón o proporción de falsos positivos (FPR = Razón de Falsos Positivos), también según se varía el umbral de discriminación (valor a partir del cual decidimos que un caso es un positivo).

Imagen 4. Gráfico ROC



Matriz de confusión (false positive, true positive). Consiste una de las métricas más sencillas e intuitivas, que se utiliza para encontrar la exactitud del modelo. Se utiliza para el problema de clasificación donde la salida puede ser dos o más tipos de clases.

La exactitud de una clasificación puede evaluarse calculando el número de objetos de la clase que han sido reconocidos correctamente (verdaderos positivos), la cantidad de objetos reconocidos correctamente, pero que no pertenecen a la clase (verdaderos negativos) y los objetos que no fueron asignados incorrectamente a la clase (falsos positivos), o los que no fueron reconocidos como objetos de la clase (falsos negativos).

Imagen 5. Matriz de confusión

		Predicción	
		Positivos	**Negativos**
Observación	**Positivos**	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

2.6 Test

Probar el sistema y obtener las métricas de predicción (accurancy).

Medidas de desempeño. Los indicadores de desempeño se pueden evaluar según el modelo, basado en el uso de un conjunto de indicadores que permiten medir el desempeño del algoritmo, en un grupo de datos utilizados para las pruebas. El proyecto involucró problemas de clasificación, por lo que se hace uso de indicadores de desempeño que se dividen en dos etapas, la primera etapa es:

Entrenamiento (proceso de aprendizaje), la segunda etapa es prueba. En la fase de entrenamiento se implementarán indicadores o métricas para elegir el mejor modelo para la fase de clasificación. En otras palabras, las medidas de evaluación se utilizan para distinguir y seleccionar la mejor solución para proporcionar con mayor precisión la evaluación futura y la predicción. En la etapa de prueba se utilizan las métricas de evaluación para medir la efectividad que produce el clasificador, cuando se prueba con los datos destinados para ese fin.

Precisión. Es una medida que dice qué de la proporción de objetos que se dejaron en una clase, cuales en realidad pertenecen a esta.

Ecuación 4. Ecuación de precisión binaria

$$Precision\ binaria = \frac{VP}{VP + FP}$$

El enfoque de evaluación para el modelo binario en términos de las etiquetas de objetos que son positivas.

Ecuación 5. Ecuaciones de precisión multivariado

$$Precision_{\mu}\ multivariado = \frac{\sum_{i=1}^{l} vp_i}{\sum_{i=1}^{l}(vp_i + fp_i)}$$

$$Precision_{M}\ multivariado = \frac{\sum_{i=1}^{l} \frac{vp_i}{(vp_i + fp_i)}}{l}$$

NOTA GENERAL: Estas imagenes estan todas muy pixelada.

Implementación del aplicativo web: el sistema de reconocimiento de lenguaje de señas debe estar ligado a un aplicativo amigable, didáctico y muy sencillo para que el usuario (la persona discapacitada) proporcione imágenes de su lenguaje de señas e identifique que tipo de letra del abecedario o número está realizando.

Es útil para la evaluación y el ejercicio del aprendizaje obtenido en la asociación AsorHuil.

La imagen capturada se almacena en el dataset para así luego realizar los respectivos procesos para la identificación de la seña descrita.

3. Resultados

Imagen 6. Constantes para creación del modelo

```
INIT_LR = 1e-3
epochs = 1000
batch_size = 24
```

Imagen 7. Resultados de precisión de cada una de las categorías de letras del dataset

```
140/140 [==============================] - 1s 7ms/step - loss: 0.8838 - accuracy: 0.7359 - val_loss: 0.5719 - val_accuracy: 0.8542
Epoch 999/1000
140/140 [==============================] - 1s 7ms/step - loss: 0.9126 - accuracy: 0.7353 - val_loss: 0.5724 - val_accuracy: 0.8566
Epoch 1000/1000
140/140 [==============================] - 1s 7ms/step - loss: 0.9300 - accuracy: 0.7317 - val_loss: 0.5691 - val_accuracy: 0.8530
```

4. Conclusiones y trabajos futuros

Al finalizar el estudio y desarrollo del sistema para el reconocimiento del lenguaje de señas colombiano se llegó a las siguientes conclusiones:

Se introdujo un enfoque para la detección y clasificación del lenguaje de señas colombiano a las letras del abecedario, se utilizó una arquitectura de redes neuronales convolucionales con visión artificial que permitieron la reducción del porcentaje de error durante la clasificación. Los cambios de iluminación se deben controlar sobre todo para la construcción de aplicaciones no industriales, ya que de ello depende el éxito o fracaso de los resultados que emita el sistema. OpenCV y Tensorflow cuentan con una extensa documentación, así como la fácil integración con el lenguaje de programación Python. Además, son una herramienta de software libre, lo que permitió reducir el costo de la aplicación sin perder la calidad. El desarrollo del presente sistema facilitó la enseñanza en la Asociación de Sordos AsorHuil de Neiva-Huila, permitiéndoles una manera más ágil de enseñar.

Referencias

Aguilera, L.-S. & Arias, H.P. (2016). Computer vision applied for recognition sign language.

Angel. (13 de octubre de 2009). Mapa de bit. http://mapajs.blogspot.com/2009/10/mapa-de-bits.html

Anónimo (2019). Algoritmos de segmentación evaluados.

Anónimo. (s.f.). Lenguaje de señas colombiana. *Un idioma para conocer.* https://sites.google.com/site/lenguasdecorazon/

Anónimo. (s.f.). Segmentación de imágenes. http://alojamientos.us.es/gtocoma/pid/tema4.pdf

Botina-Monsalve, D. J., Domínguez, M., Madrigal, C. & Castro, A. (2018). Clasificación automática de las vocales en el lenguaje de señas colombiano. *TecnoLógicas, 21*(41), 103-114. http://www.scielo.org.co/scielo.php?pid=S0123-77992018000100007&script=sci_abstract&tlng=es

Chiguano, E., Moreno, N. & Corrales, L. (2018). *Diseño e implementación de un sistema traductor de lenguaje de señas de texto mediante visión artificial en un ambiente controlado*. EPN, Quito.

Fotored (15 de mayo de 2016). Contraste. http://www.digitalfotored.com/imagendigital/contraste.htm

Guerrero-Balaguera, J. D., & Pérez-Holguín, W. J. (2015). Sistema traductor de la lengua de señas colombiana a texto basado en FPGA. *Dyna, 82*(189), 172-181.

Herrera, P. J. (enero de 2016). Segmentación de regiones. https://www.researchgate.net/figure/Figura-61a-Imagen-original-b-Imagen-resultante-binarizada_fig1_323200183

https://journal.espe.edu.ec

Ordoñez, C. (04 de junio de 2016). Formatos de imagen digital. *Revista Digital Universitaria, 5*(7). http://www.revista.unam.mx/vol.6/num5/art50/may_art50.pdf

Pajares Martinsanz, G. & de la Cruz García. J. M. (2001). *Visión por computador. Imágenes digitales y aplicaciones*. Madrid: RA-MA.

Pequero Núñez, D. (s.f.). Realce y restauración de imagen. http://www.lpi.tel.uva.es/muitic/pim/docus/Realce_y_restauracion.pdf

Pérez, J. B. (23 de junio de 2016). Espacio HSI. *Uso del sistema HSI para asignar falso color a objetos en imágenes digitales.* http://www.scielo.org.mx/pdf/rmfe/v54n2/v54n2a11.pdf

Prospectiva, *16*(2), 41-48. https://doi.org/10.15665/rp.v16i2.1488

Santamaría, B. A. (2002). Realce de imágenes: filtrado espacial. *Revista AET, 17,* http://www.aet.org.es/?q=revista17-4#:~:text=Santamar%C3%ADa.,procesamiento%20por%20grupo%20de%20pixeles.

Satorres, S. (s.f.). *Detección de bordes en una imagen.* Universidad de Jaén, http://www4.ujaen.es/~satorres/practicas/practica3_vc.pdf

Tirado, A. B. (2017). El ruido digital: qué es, por qué aparece y 3 métodos para eliminarlo. Foto24.

Universidad de Sevilla (s.f.). Textura. *Concepto de textura.* http://bibing.us.es/proyectos/abreproy/11494/fichero/PROYECTO%252FCapitulo+3.pdf

Univisión (8 de noviembre de 2013). Cómo funciona la lengua de señas. http://www.batanga.com/curiosidades/4910/como-funciona-la-lengua-de-senas

Villa, B., Valencia, V. & Berrio, J. (2018). Digital image processing applied on static sign language recognition system.