Fecha de publicación junio 03, 2024, Fecha de la versión actual junio 03, 2024.

Sistema de entrenamiento y generación de modelos para SLR (Sign Langua Recognition)

Flores Hernández Jaime Azael
[1] Universidad Autónoma de Baja California (Facultad de Ingeniería Diseño y Arquitectura)

Autor correspondiente: Flores Hernández Jaime Azael (e-mail: jaime.flores.hernandez@uabc.edu.mx).

ABSTRACT This study investigates the development and implementation of a system to generate, train and use K-Nearest Neighbors (KNN) models to recognize and label hand gestures, applied to sign language. The objective is to produce a functional model for sign language recognition and thus develop analytical and data science knowledge for future projects in the area. The model achieved an accuracy of 99.82% in the classification of static gestures, validating its effectiveness. OpenCV was used for data capture and Scikit-learn for model development, with metrics such as precision, recall and F1 Score to evaluate performance. This study contributes to the advancement in sign language recognition and suggests future improvements in the expansion of the gesture dictionary and the integration of additional technologies for more robust and accessible applications.

Español: Este estudio investiga el desarrollo e implementación de un sistema para generar, entrenar y utilizar modelos de K-Nearest Neighbors (KNN) para reconocer y etiquetar gestos de manos, aplicado al lenguaje de señas. El objetivo es producir un modelo funcional para el reconocimiento de lenguaje de señas y así desarrollar conocimiento análitico y de ciencia de datos para futuros proyectos en el área.. El modelo alcanzó una precisión del 99.82% en la clasificación de gestos estáticos, validando su eficacia. Se utilizó OpenCV para la captura de datos y Scikit-learn para el desarrollo del modelo, con métricas como precisión, recall y F1 Score para evaluar el rendimiento. Este estudio contribuye al avance en el reconocimiento de lenguaje de señas y sugiere mejoras futuras en la expansión del diccionario de gestos y la integración de tecnologías adicionales para aplicaciones más robustas y accesibles.

INDEX TERMS Reconocimiento de Lenguaje de Señas, K-Nearest Neighbors (KNN), Computer Vision, Modelos de Aprendizaje Automático, Precisión en Modelos de Lenguaje de Señas, Gestión de Datos de Lenguaje de Señas, OpenCV en Reconocimiento de Gestos, Interfaz de Usuario en Aplicaciones de Lenguaje de Señas, Evaluación de Modelos de Lenguaje de Señas, Implementación de Modelos de KNN

I. INTRODUCCIÓN

En el mundo se sabe que no todas las personas disponen de los medios para poder comunicarse de forma “adecuada” siendo así el tener alguna discapacidad auditiva o verbal, por ende se suelen incorporar a otro medio de comunicación como lo es el lenguaje de señas, se puede entender que el aprendizaje de esta forma de comunicación conlleva tiempo y esfuerzo, no solo para aquellos con la discapacidad del habla o escucha, sino también para aquellos que desean comunicarse con estas personas, por ende el problema a resolver sería el cerrar esa brecha entre el conocimiento y aprendizaje que requiere el aprender una nueva forma de comunicación o al menos hacerla un poco más accesible, esto por medio de un software que utiliza ciencia de datos.

Este documento presenta un estudio de investigación enfocado en el desarrollo e implementación de un sistema de software para entrenar y utilizar modelos de K-Nearest Neighbors (KNN) para etiquetar e identificar señas de manos, con el fin de aplicarlo en lenguajes de señas. El uso de la tecnología para facilitar la comunicación y el aprendizaje de lenguajes de señas es una necesidad creciente, especialmente para mejorar la accesibilidad y la inclusión de personas con discapacidad auditiva.

El objetivo principal es evaluar la factibilidad y eficacia de un modelo construido con herramientas y tecnologías simples. Además, se pretende generar conocimientos que sean útiles para trabajos futuros y otras aplicaciones tecnológicas. A largo plazo, el desarrollo y mejora continua

de este modelo buscará aumentar su alcance y utilidad en contextos educativos y de estudio independiente.
La sección II cubre el trabajo relacionado y los estudios previos en el campo del reconocimiento de señas de manos. La sección III detalla la metodología utilizada para desarrollar el modelo de KNN, incluyendo la recopilación y el procesamiento de datos. La sección IV presenta los resultados obtenidos y su análisis. La sección V provee una discusión general del proyecto. Finalmente, la sección VI ofrece las conclusiones y recomendaciones para futuros trabajos.

II. TRABAJO RELACIONADO

Para el reconocimiento de señas aplicado en lenguaje de señas se han realizado múltiples estudios e investigaciones previas de forma activa con diversos enfoques y metodologías aplicados. Varios de los trabajos previos han tenido un alto éxito en producir modelos totalmente funcionales y efectivos para su tarea.

A) Varios estudios se han centrado en el análisis e identificación del lenguaje de señas, desarrollando modelos que traducen gestos a texto o incluso a voz de manera efectiva. Philippe Dreuw y su equipo (2007)[1] lograron un 17.9% de Word Error Rate (WER) utilizando un sistema de reconocimiento del habla aplicado al lenguaje de señas. Sin embargo, su enfoque se vio limitado por la cantidad reducida de datos disponibles para el entrenamiento del modelo. A pesar de emplear un sistema complejo, la escasez de datos afectó significativamente los resultados obtenidos. Aunque el proyecto mostró potencial, no fue completamente explorado, dejando espacio para mejoras y mayores resultados en investigaciones futuras.

B)Se han desarrollado numerosos modelos utilizando algoritmos de aprendizaje automático, incluido K-Nearest Neighbors (KNN) debido a su sencilla implementación. Ahmed Sultan y su equipo (2022)[2] llevaron a cabo un estudio comparativo en el que se emplearon varios algoritmos, incluido KNN, para evaluar su efectividad en la tarea de reconocer e identificar el lenguaje de señas. Inicialmente, emplearon KNN únicamente para la detección de la mano derecha, obteniendo una precisión del 85%. Posteriormente, ampliaron su enfoque para detectar ambas manos, alcanzando una precisión del 99.8%.

III. METODOLOGÍA

Esta sección describe los métodos y procedimientos utilizados para desarrollar y evaluar un modelo de reconocimiento de lenguaje de señas utilizando K-Nearest Neighbors (KNN) y técnicas de computer vision.

El estudio se diseñó como una investigación cuantitativa con un enfoque experimental para evaluar la eficacia del modelo KNN en la identificación de señas de manos. Para el desarrollo y entrenamiento del sistema, se adoptó un enfoque flexible que incluyó la obtención y manejo de

C) La implementación de métodos de captura es fundamental en los procesos de computer vision, especialmente para garantizar la obtención de imágenes de calidad y datos suficientes para alimentar modelos de aprendizaje automático de manera eficiente. Jorge Casas (2019)[3] describe cómo uno de los modelos desarrollados utilizó el Leap Motion Controller, Microsoft Kinect y TensorFlow para reconocer 17 signos estáticos con una precisión cercana al 100% mediante el uso de machine learning y redes neuronales. Este enfoque demuestra una de las múltiples formas de lograr una captura de datos efectiva en aplicaciones de reconocimiento de señas.

D) A pesar de los grandes avances tecnológicos en hardware y software, aún existen puntos de oportunidad y limitaciones significativas en el campo del reconocimiento del lenguaje de señas. Mazen Selim (2019)[4] abordó estos aspectos en un ensayo que destacó varios puntos críticos para el avance tecnológico en el reconocimiento de señas. Entre estos desafíos se incluyen el reconocimiento y seguimiento dinámico de las manos, la integración de gestos y rasgos no manuales como el cuerpo o el rostro, y la validación a gran escala de modelos de Continuous Sign Language Recognition (CSLR). Uno de los desafíos más importantes no solo implica el reconocimiento individual de los signos y su significado, sino también la composición de frases complejas utilizando estos signos.

E) El objetivo de este estudio es proporcionar un conjunto de pasos para desarrollar modelos de Recognition of Sign Language (RSL) utilizando herramientas, materiales y equipos accesibles. Se establece una guía que promueve el uso de K-Nearest Neighbors (KNN) como base y computer vision en tiempo real para la implementación, aplicación y adquisición de datos sin la necesidad de una base de datos previamente construida. Además, se recomienda aplicar técnicas de imputación de datos y calibración del modelo para mejorar los resultados.

El propósito no es superar otras tecnologías o modelos existentes, sino contribuir al estudio académico y científico en el campo de la ciencia de datos y análisis de soluciones mediante algoritmos de aprendizaje automático y computer vision.

datos específicos, en lugar de utilizar un dataset preexistente. Esto permitió una mayor flexibilidad en el desarrollo y calibración inicial del modelo.

Para la recopilación de datos, se utilizó OpenCV y una webcam básica de ordenador, permitiendo capturar datos de forma estructurada y ordenada en tiempo real para su posterior almacenamiento. La decisión de obtener datos de manera práctica y directa respondió a la necesidad de ajustar y calibrar el modelo de forma dinámica.

Además del modelo, se desarrolló un software con una interfaz de usuario para facilitar la interacción con el modelo. Python[5] fue el lenguaje de programación elegido por su capacidad de integrar fácilmente software visual con software matemático para el modelo. Las principales librerías utilizadas fueron:

- CustomTkinter: para la interfaz de usuario.
- OpenCV[6]: para el manejo de la webcam.
- MediaPipe: para el reconocimiento básico de puntos clave (en este caso, manos).
- Scikit-learn[7]: para el desarrollo del modelo KNN.

Se probó la funcionalidad del modelo con una cámara web con resolución de 1920x1080 píxeles, escalada a 640x360 píxeles. El software con interfaz gráfica puede ejecutarse en una computadora con un procesador de 4 núcleos a 2.4 GHz y 8 GB de memoria RAM.

La captura de datos se realizó teniendo en cuenta la visibilidad y posición de las manos, así como si ambas manos son visibles o solo una. La captura de datos consideró cuatro aspectos fundamentales para la generación de las colecciones de datos:

1. Visibilidad de las manos.
2. Posición de las manos.
3. Visibilidad de ambas manos.
4. Identificación de cuál mano es visible (izquierda o derecha).

Atributo	Tipo de Dato	Descripción
photo_path	Cualitativo nominal	Ruta de la imagen del gesto de la mano
gesture_label	Cualitativo nominal	Etiqueta del gesto de la mano
keypoints_left	Cuantitativo continuo	Coordenadas continuas de los puntos clave de la mano izquierda
keypoints_right	Cuantitativo continuo	Coordenadas continuas de los puntos clave de la mano derecha

Siendo así que tenemos los datos independientes, que son los puntos clave (keypoints) de ambas manos (keypoints_left y keypoints_right), y la variable dependiente, que es el gesto (gesture_label), para el entrenamiento del modelo y funcionalidades específicas se requieren estos tres últimos, ya que son los que definen la base necesaria para el modelo. Cabe resaltar que los keypoints son vectores; cada mano posee 21 puntos clave (vectores) y cada punto clave tiene coordenadas en tres orientaciones: horizontal, vertical y profundidad.

La forma de integrar estos datos entre sí fue el cerciorarse de que estaban normalizados y en el caso de no encontrarse una mano poder asegurarse de que el modelo no tuviera conflicto con esto dándole la capacidad no requerir todos los puntos clave, siendo que no se necesita la ruta de la imagen debido a que los datos necesarios se encuentran en los keypoints esta columna puede ser descartada, algo que se pudo haber probado fue el utilizar coordenadas relativas a un punto clave inicial en cada mano como otra forma de procesar los datos previos a su uso, esta idea queda disponible para futuras pruebas.

Los datos utilizados para el modelo base de pruebas fueron 800 imágenes de 8 gestos, 100 imágenes por gesto, los cuales incluyen las etiquetas de [alegre, azúcar, caída, comida, conmigo, contra, mio, primero], este modelo base se realizó con el proposito de aplicar una comparativa posterior a un modelo más robusto con características similares, siendo así que el segundo modelo utiliza la misma cantidad de gestos pero con un dataset de 8000 imágenes, 1000 imágenes por cada gesto.

La integración de estos datos se realizó asegurándose de que estaban normalizados. En caso de que una mano no fuera detectada, se garantizó que el modelo no tuviera conflictos al respecto, permitiéndole operar sin requerir todos los puntos clave.

Dado que los datos necesarios se encuentran en los keypoints, la columna que contiene la ruta de la imagen pudo ser descartada. Una posible mejora futura podría ser el uso de coordenadas relativas a un punto clave inicial en cada mano como una forma alternativa de procesar los datos antes de su uso.

Para clasificar los gestos de las manos, se utilizó el algoritmo K-Nearest Neighbors (KNN). El modelo se entrenó y validó utilizando un conjunto de datos dividido en datos de entrenamiento y de prueba. También se realizaron pruebas con random forest y regresión múltiple; sin embargo, se observó que los algoritmos de regresión y estadística predictiva presentaban desventajas significativas en comparación con los algoritmos de aprendizaje automático más avanzados.

Para la validación del modelo, se aplicó validación cruzada para evaluar su estabilidad. En el software final se incluyó la capacidad de entrenar y generar modelos para su análisis y uso posterior, permitiendo evaluar su rendimiento y capturar datos con el mismo software. Esto facilita probar distintas configuraciones y ambientes, o incluso entrenar diferentes modelos para diversos tipos de lenguajes de señas. Los modelos se pueden entrenar y evaluar para generar métricas que incluyen la precisión y eficiencia.
Una de las limitaciones fue el tiempo de desarrollo establecido, lo cual restringió algunas características

planeadas para el sistema. Las funcionalidades implementadas se limitaron a la generación de modelos, captura de datos, imputación de datos, validación de modelos y graficación de resultados, así mismo otra de las limitantes significativas fue la cantidad de gestos seleccionados para probar la eficacia de los modelos durante las pruebas.

IV. RESULTADOS

Para la realización tanto del modelo (en funcionalidad presente) como del software se utilizaron los mismos datos producidos por versiones pre-eliminares del software al tomar captura de la webcam en tiempo real para así evaluar la eficacia, las métricas principales empleadas para evaluar el desempeño de los modelos incluyen precisión, recall y F1 Score así como matrices de confusión.

El modelo KNN alcanzó una precisión promedio del 99.82% en la clasificación de gestos de manos. Se observó una mejora con respecto a los resultados obtenidos con otros modelos siendo muy por debajo de este valor, aun así esto no implica la total precisión del modelo, ya que si bien tiene una alta taza de éxito al identificar gestos estáticos aún queda a la espera de aplicarlo para gestos dinámicos, la evaluación se realizó tomando en cuenta la precisión por medio de un reporte de clasificación y validación cruzada, de esta forma se obtuvieron los siguientes datos:

```
Validación cruzada (5-fold): [0.99910714 0.99821429 1.         0.99821429 0.99910714]
Promedio de validación cruzada: 0.9989285714285714
```

FIGURA 1. Datos de validación cruzada para el último modelo estable.

En la figura 1 se puede observar el rango de precisión después de 5 fold (pases de evaluación), además se produjeron métricas de precisión.

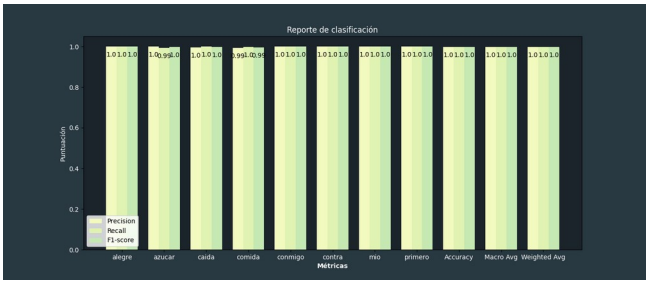


FIGURA 2. Reporte de clasificación para las 8 palabras previamente seleccionadas con el dataset de 8000 imágenes [modelo N2].

Si observamos la figura número 2 se puede observar que se obtuvo una precisión considerablemente alta para cada uno de los gestos, esto se puede comparar al modelo anterior en su reporte de clasificación.

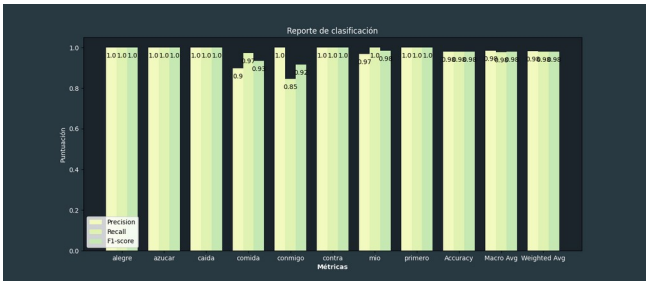


FIGURA 3. Reporte de clasificación para las 8 palabras previamente seleccionadas con el dataset de 800 imágenes [modelo N1].

En la figura número 3 podemos observar el comportamiento previo para las mismas 8 palabras, pero esta vez aplicado a 800 imágenes en vez de 1000, aquí se puede observar la limitante en cuanto a cantidad de información para el entrenamiento de un modelo con el software final, si hay diferencia significativa entre la cantidad de imágenes disponibles, obteniendo así una precisión de 98%.

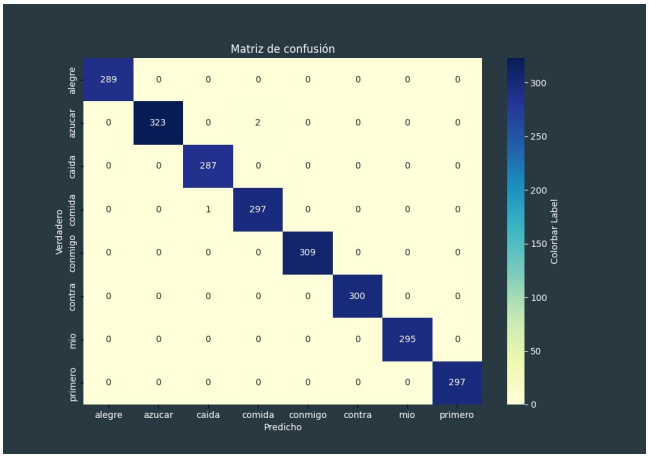


FIGURA 4. Matriz de confusión para el modelo N2 con 8000 imágenes.

En la figura 4 podemos observar la cantidad de fallos aproximados para la distinción entre palabras para el segundo modelo entrenado con el software usando KNN, se puede ver que la taza de predicciones es alta en relación con los datos verdaderos.

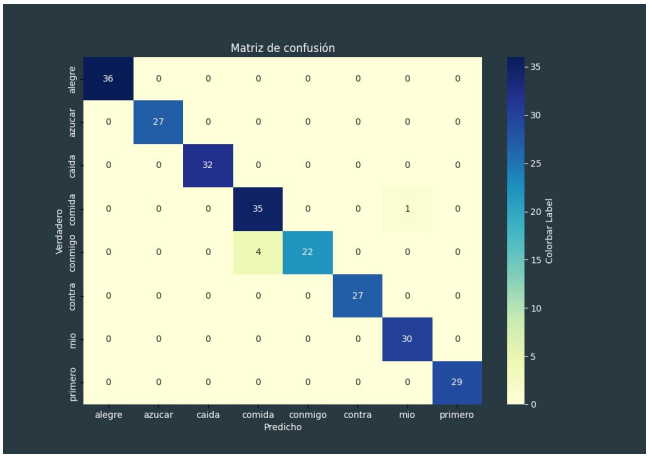


FIGURA 5. Matriz de confusión para el modelo N1 con 800 imágenes.

Si observamos la figura 5 con la matriz de confusión para el modelo entrenado con las 800 imágenes podemos ver que

el rango limitado de datos de entrenamiento sí que afecta significativamente a la distinción entre datos.

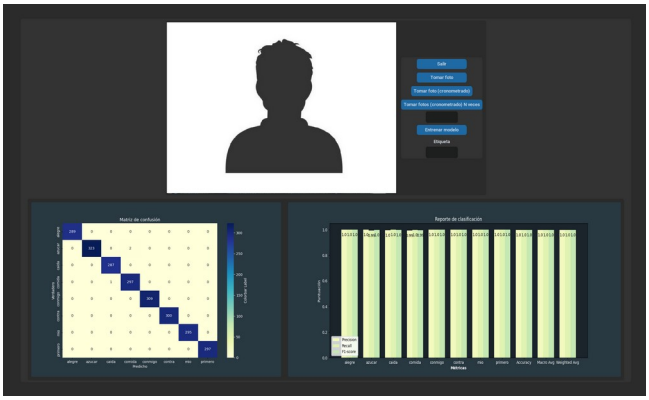


FIGURA 6. Software final mostrando las características a disposición del 3 de junio del 2024.

En la figura 6 se muestra el software final, el cual permite tomar fotografías para almacenarlas en una base de datos, capturar imágenes según un cronometraje predefinido y configurar un rango de fotos a capturar con etiquetas personalizadas. Además, incluye una sección para visualizar en tiempo real la webcam, los puntos clave detectados y los resultados de los gestos clasificados por el modelo. También se presentan los gráficos correspondientes a la matriz de confusión y al reporte de clasificación.

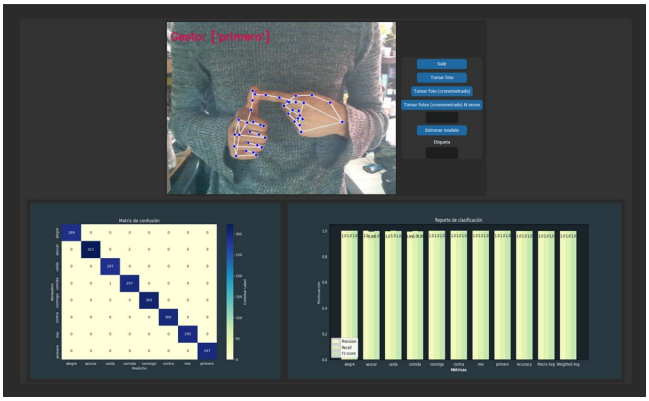
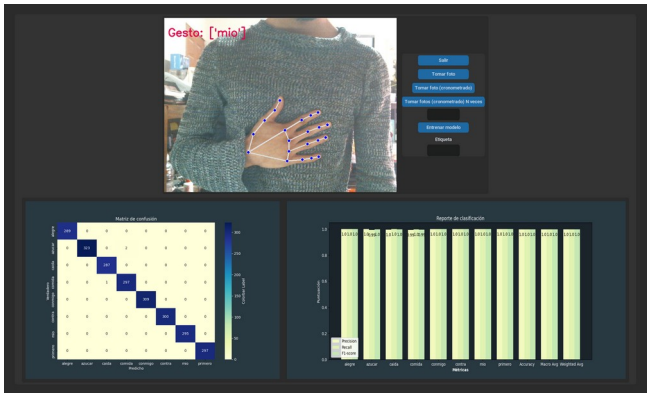


FIGURA 7. Software final en uso para la detección de gestos en el último modelo entrenado (ambas manos).

VI. CONCLUSIONES



En la figura 7 se puede observar el sistema funcionando con el último modelo entrenado para la fecha, mostrando el reconocimiento de la seña de “primero”, y dibujando los puntos clave (Key points) de ambas manos, siendo esta una de las primeras condiciones para el correcto funcionamiento, poder sumar ambas manos para gestos específicos y su reconocimiento.

FIGURA 8. Software final en uso para la detección de gestos en el último modelo entrenado (una mano).

En la figura 8 comprobamos la funcionalidad del modelo con una sola mano sin tener que estar presente la mano derecha, de esta forma validando su funcionamiento para los distintos casos y condiciones.

Al final el sistema completo se publicó en github[8] de forma libre para su distribución como código abierto.

V. DISCUSIÓN

Con base en los resultados obtenidos mediante el software y el método de entrenamiento utilizado para desarrollar modelos de reconocimiento, logramos alcanzar una precisión máxima del 99.82%.

Esto confirma la efectividad del software y del último modelo para cumplir con el objetivo inicial, estableciendo así una sólida base para investigaciones futuras.

Los principales factores que influyeron en estos resultados fueron la cantidad y complejidad de los datos obtenidos de las imágenes, así como la diversidad de gestos aplicados. Con un período adicional de tiempo, sería factible completar la integración de un diccionario completo de lenguaje de señas.

Los hallazgos de este estudio pueden tener aplicaciones significativas en la adquisición de conocimientos fundamentales para el desarrollo de sistemas inteligentes mediante algoritmos de aprendizaje automático y visión por computadora. Además, contribuyen de manera notable al campo de la ciencia de datos.

Dado que todo el desarrollo se realizó en un entorno donde la recolección de datos fue local, la herramienta desarrollada está disponible para su uso libre, facilitando su distribución bajo licencias de código abierto según el uso y consideración de cada individuo.

El estudio logró desarrollar un sistema capaz de utilizar y entrenar modelos de (LSR) utilizando KNN como base, alcanzando una precisión máxima del 99.82%.

Esto confirma que el software y el modelo desarrollados cumplen con el objetivo inicial de proporcionar una herramienta efectiva para el reconocimiento de lenguaje de señas, comprensión académica y producción de datos esenciales para el estudio de la ciencia de datos en modelos de aprendizaje automático con computer vision.

Los resultados obtenidos demuestran que el modelo de KNN es altamente eficaz para la clasificación precisa de gestos de manos. La comparación con investigaciones previas y la alta precisión alcanzada validan su utilidad y eficiencia en aplicaciones prácticas.

La calidad y diversidad del conjunto de datos utilizados, junto a la elección adecuada del algoritmo KNN, fueron fundamentales para los resultados positivos obtenidos. La capacidad de adaptarse a diferentes configuraciones y ambientes también contribuye significativamente al éxito del modelo, así como las comparativas entre modelos entrenados bajo diferentes rangos de datos.

Una limitación importante fue el tiempo limitado por el desarrollo y entrenamiento de los modelos, lo cual afectó la implementación completa de un diccionario extenso de gestos entre otras características esperadas. Mejoras futuras podrían centrarse en la expansión y pruebas con otros conjuntos de datos, optimización del software y modelos, inclusión de características de reconocimiento facial y corporal general así como de construcción de frases continuas basadas en gestos dinámicos para aumentar la utilidad y robustez tanto del sistema como de los modelos.

Este estudio proporciona una base sólida para aplicaciones prácticas en el desarrollo de sistemas de asistencia para personas con discapacidades auditivas y aquellos que quieran apoyarse de una herramienta útil para entender el lenguaje de señas. Las futuras investigaciones podrían explorar la integración de tecnologías adicionales para mejorar la usabilidad del software en diversos entornos.

REFERENCES

[1] P. Dreuw. “Speech recognition techniques for a sign language recognition system”. i6 - zentral: Language Processing and Pattern Recognition. Accedido el 3 de junio de 2024. [En línea]. Disponible: http://www-i6.informatik.rwth-aachen.de/publications/download/154/Dreuw-INTERSPEECH-2007.pdf

[2]A. Sultan, W. Makram, M. Kayed y A. A. Ali. “Sign language identification and recognition: A comparative study”. De Gruyter. Accedido el 3 de junio de 2024. [En línea]. Disponible: https://www.degruyter.com/document/doi/10.1515/comp-2022-0240/html

[3] J. Casas. “Motion capture methods and machine learning for sign language recognition | Archivo Digital UPM”. Archivo Digital UPM | Archivo Digital UPM. Accedido el 3 de junio de 2024. [En línea]. Disponible: https://oa.upm.es/56792/

[4] M. Selim. “Advances, challenges and opportunities in continuous sign language recognition”. Academia.edu - Share research. Accedido el 3 de junio de 2024. [En línea]. Disponible: https://www.academia.edu/95454489/Advances_Challenges_and_Opportunities_in_Continuous_Sign_Language_Recognition?uc-sb-sw=36428832

[5] “Welcome to python.org”. Python.org. Accedido el 3 de junio de 2024. [En línea]. Disponible: https://www.python.org/

[6] “Home”. OpenCV. Accedido el 3 de junio de 2024. [En línea]. Disponible: https://opencv.org/

[7] “Scikit-learn: Machine learning in Python — scikit-learn 1.5.0 documentation”. scikit-learn: machine learning in Python — scikit-learn 0.16.1 documentation. Accedido el 3 de junio de 2024. [En línea]. Disponible: https://scikit-learn.org/stable/

[8] A. Hernández. “GitHub - CodigoLasagna/reconPath”. GitHub. Accedido el 3 de junio de 2024. [En línea]. Disponible: https://github.com/CodigoLasagna/reconPath