

Hand and Gestures Tracking with AI & OpenCV

Riveros Pérez O. L.¹ and Vargas Gutiérrez Y. E.²

¹oriverosp@unal.edu.co

²yvargasgu@unal.edu.co

Abstract Hand Tracking, is a constantly evolving technology that allows detecting and tracking the movement of hands in real time. This capability not only facilitates the control of interfaces, but also opens up a range of possibilities from sign language translation to the creation of digital art through intuitive gestures. In a first approach, by means of segmentation, morphology, convolution and filtering techniques, the ideal way to characterize hand gestures is identified.

Keywords

Hand tracking, Artificial Intelligence, OpenCV, Threshold segmentation, Color space filters.

Introducción

El Hand Tracking, o seguimiento de manos, es una tecnología en constante evolución que permite detectar y rastrear el movimiento de las manos en tiempo real. Esta capacidad no solo facilita el control de interfaces, sino que también abre un abanico de posibilidades que van desde la traducción del lenguaje de señas [1], [2], [3], [4] hasta la creación de arte digital mediante gestos intuitivos [5], [6].

En este contexto, se han explorado diversos enfoques para identificar patrones de manera óptima a partir de las poses de la mano y sus significados. Entre estos enfoques, destaca el uso de la API Mediapipe de Google [1], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14] que se ha convertido en una herramienta popular. Mediante esta API, se obtiene un modelo de puntos de referencia que permite estimar con precisión la pose de la mano. Sin embargo, también se han desarrollado otros métodos que emplean técnicas avanzadas de Machine Learning y Deep Learning [2], [3], [15], [16], [17], [18], [19], [20] con el objetivo de estimar directamente los patrones de la mano o incluso de obtener modelos de marcas de referencia de manera similar a la API Mediapipe [21].

Aunque los enfoques previamente mencionados han logrado avances significativos, es crucial tener en cuenta que todos ellos se basan en técnicas avanzadas que, con frecuencia, conllevan desafíos adicionales en cuanto a rendimiento y eficiencia. Esto puede dificultar la adecuada evaluación del problema de extracción de características de la mano. Por esta razón, también se exploran enfoques más simples basados en técnicas de segmentación y morfología para desarrollar modelos de mano más simples y eficaces [22], [23], [24], [25], [26].

El artículo propone un ejercicio académico para implementar y comparar varios enfoques del reconocimiento de la mano, con el objetivo de evaluar su rendimiento computacional, precisión y limitaciones. Se comenzará con una revisión del estado del arte en la siguiente sección, seguida de la presentación de diversas arquitecturas para el reconocimiento de la mano. Posteriormente, se analizarán los resultados obtenidos de cada arquitectura y se proporcionarán conclusiones correspondientes.

Antecedentes

El Hand tracking, como se ha mencionado anteriormente, tiene una amplia gama de aplicaciones, y una de las más notables es la detección del lenguaje de señas en diferentes idiomas. Este desafío particular implica tanto poses estáticas como dinámicas de las manos. En un estudio reciente [1], los autores proponen el uso de la API Mediapipe de Google para extraer características de la imagen. Estas características se someten a procesamientos geométricos

y temporales para clasificar gestos estáticos y dinámicos utilizando algoritmos de Support Vector Machine (SVM). Como resultado de este enfoque, logran una precisión del 97.20% en la detección de gestos estáticos y dinámicos en el lenguaje de señas japonés. Este avance representa un paso significativo hacia la comprensión y aplicación efectiva del lenguaje de señas en entornos tecnológicos.

En otro estudio sobre el reconocimiento del lenguaje de señas [3], los autores proponen un enfoque innovador que combina la segmentación de la imagen con el uso de Redes Neuronales Convolucionales (CNN). En particular, emplearon MobileNetV3 para adaptarlo a dispositivos móviles. Este enfoque les permitió lograr una precisión del 75.38% en la detección de gestos estáticos en el lenguaje de señas indonesio. Este método destaca por su capacidad para ser implementado en dispositivos móviles, lo que lo hace accesible y práctico para aplicaciones cotidianas.

Otro aspecto importante del Hand Tracking es el reconocimiento de gestos estáticos para la interacción con el entorno. En un estudio reciente [20], los autores presentan una Red Neuronal Convolutacional (CNN) llamada RGRNet, diseñada específicamente para imágenes desenfocadas. Este enfoque logra una precisión del 78.2%; no obstante, se destaca que requiere cierta capacidad computacional para su ejecución.

Un artículo reciente [21] destaca un enfoque investigativo innovador en el cual los autores proponen el uso de una Red Neuronal Convolutacional (CNN) para reconocer marcas de referencia de manera similar a la API MediaPipe. Su propuesta incluye el uso de imágenes con información de profundidad y un Detector de Disparo Único de Redes Neuronales Convolucionales (SSD-CNN) para extraer los puntos de referencia de la mano. Posteriormente, emplean un algoritmo de Máquinas de Vectores de Soporte (SVM) para clasificar los gestos. Los resultados obtenidos muestran una precisión del 83.45% en la identificación de gestos.

Los métodos actuales comúnmente emplean algoritmos de machine learning; sin embargo, es útil explorar enfoques que estimen las características de las manos a partir de la geometría y operaciones básicas. En este sentido, el enfoque descrito en el artículo [22] propone la estimación de la pose de la mano utilizando operaciones típicas de procesamiento de imágenes. Este método comienza obteniendo con precisión el contorno de la mano mediante varios filtros. Luego, se realiza un proceso de convolución utilizando una matriz invariante a la rotación, seguido de una umbralización de la imagen. Posteriormente, se aplica una operación de dilatación para obtener el perfil de los dedos de la mano. A partir de esta información, se puede estimar el ángulo de los dedos con respecto a la muñeca. Este enfoque simple proporciona información suficiente para estimar poses básicas.

Metodología

Para abordar el problema planteado, se propone utilizar una arquitectura basada en los sistemas de visión por computadora convencionales. Esta arquitectura, ampliamente utilizada en el campo, se detalla en la Figura 1.

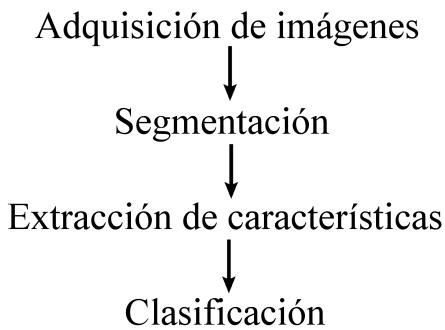


Figura 1. Arquitectura básica de un sistema de visión máquina.

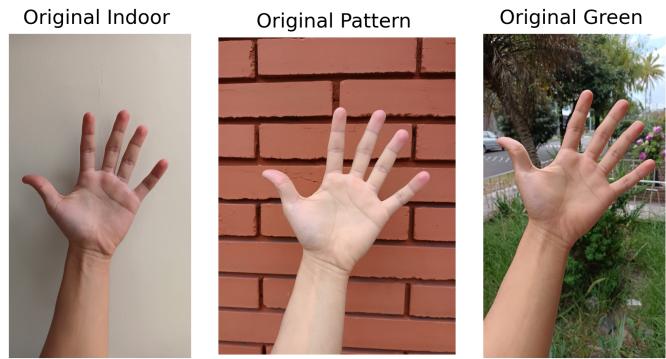


Figura 2. Imágenes de una mano en diferentes entornos de prueba.

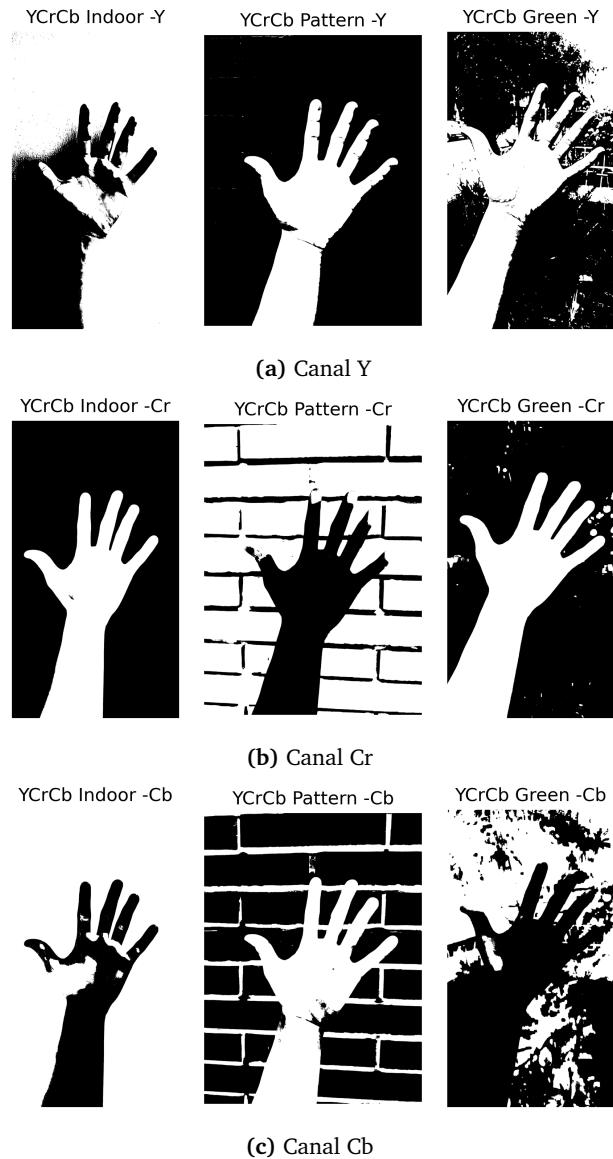


Figura 3. umbralización OTSU para diferentes canales del espacio de color YCrCb.

1 Segmentación

1.1 Umbralización OTSU y aplicación en diferentes espacios de color

El primer enfoque propuesto para desarrollar la extracción de la mano en relación con la información de fondo en la imagen consideró diferentes espacios de color. A través de la revisión del estado del arte, se observa que en [22] se utiliza el espacio de color YCrCb, en [26] se implementa el espacio de color HSV, y en [27] se lleva a cabo una revisión estadística de los espacios de color para segmentar la piel en una imagen.

Considerando lo anteriormente mencionado, se llevó a cabo una revisión del comportamiento del método de umbralización de OTSU en diferentes canales de los espacios de color YCrCb, HSV, RGB y escala de grises. Las imágenes de referencia utilizadas para evaluar la umbralización se presentan en la Figura 2., mientras que los resultados obtenidos para los espacios de color YCrCb, HSV, RGB y escala de grises se muestran en las Figuras 3, 4, 5 y 6, respectivamente.

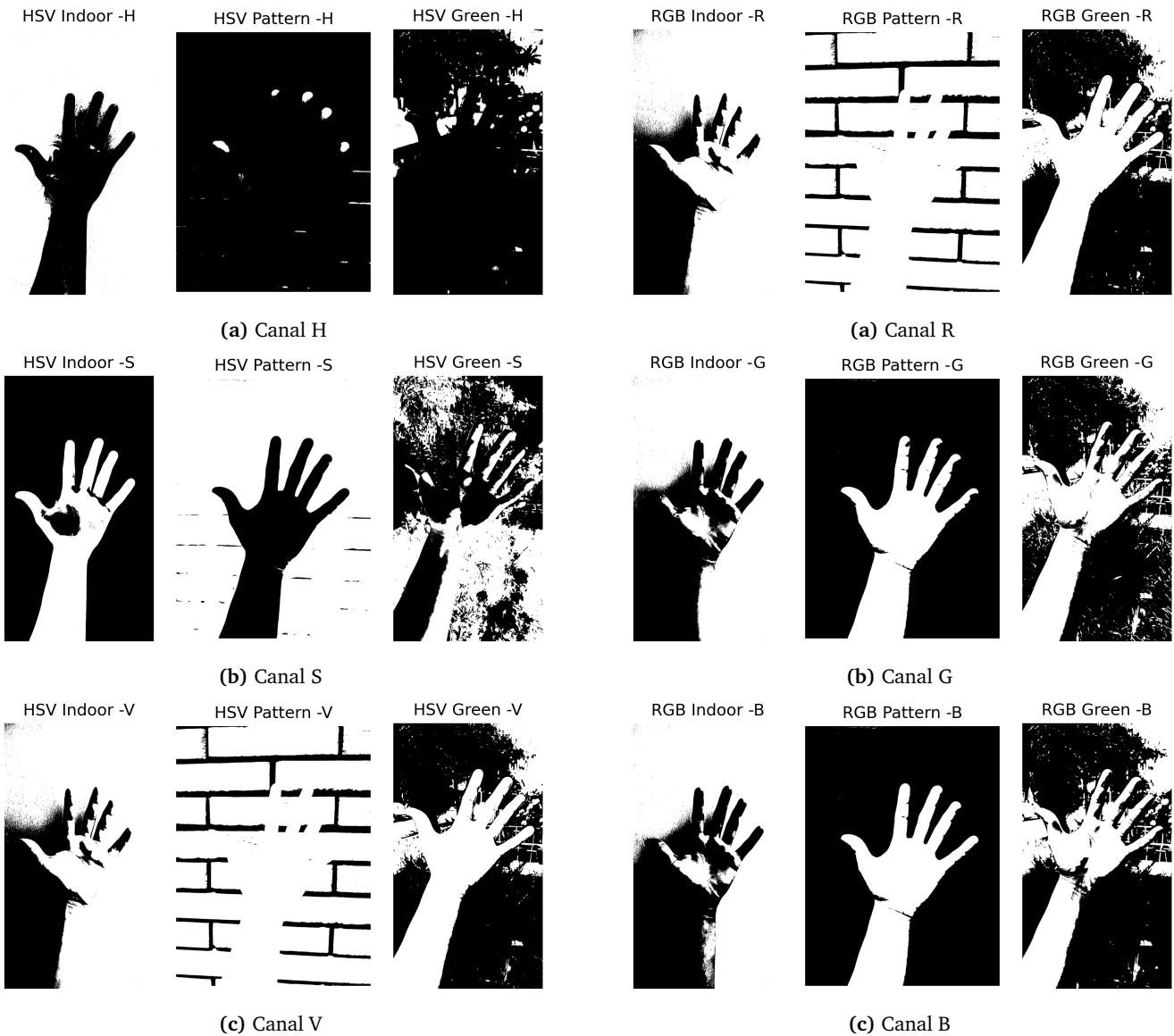


Figura 4. umbralización OTSU para diferentes canales del espacio de color HSV.

Figura 5. umbralización OTSU para diferentes canales del espacio de color RGB.

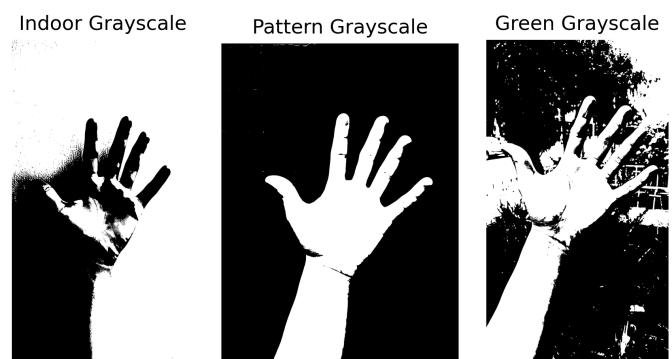


Figura 6. umbralización OTSU para la escala de grises.

Al analizar el comportamiento de la umbralización OTSU en los diferentes espacios de color, se concluye que el canal Cr del espacio de color YCrCb logra una mejor segmentación de la mano respecto al fondo en los tres escenarios evaluados. Además, es importante destacar que este método de umbralización se ejecuta en aproximadamente 13ms, lo que lo convierte en una solución computacionalmente eficiente. Para complementar estos resultados, se observó en [27] que el número de bins utilizados para normalizar el histograma de la imagen impacta significativamente en la segmentación de la piel. Por ende, al aplicar la umbralización de OTSU en el canal Cr del espacio de color YCrCb, se probó reducir el número de bins del histograma a 128 y 16, respectivamente. Los resultados de estos experimentos se muestran en la Figura 7.

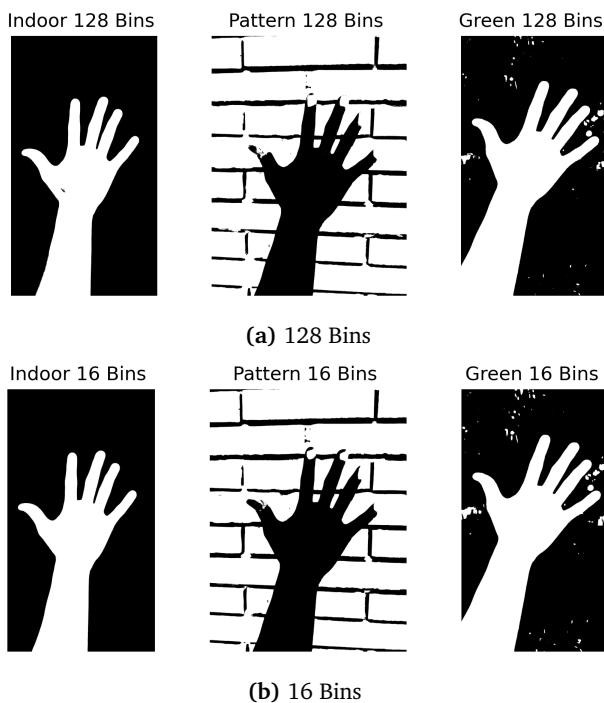


Figura 7. Reducción del número de bins del histograma en la umbralización OTSU del canal Cr.

Al analizar los resultados obtenidos al normalizar el histograma con una menor cantidad de bins, se observa una imagen más suave en general, que incluso corrige imperfecciones de la umbralización en el caso de utilizar 16 bins. Además, es importante mencionar que la reducción del número de bins del histograma puede disminuir el tiempo de ejecución de la umbralización de OTSU hasta en 9ms cuando se utilizan 16 bins.

1.2 Segmentación en Color

Un método bastante popular, como se demuestra en los artículos [26], [25] y [23] es la segmentación en color utilizando el espacio de color HSV. Este enfoque se basa en la suposición de que la tonalidad de la piel presenta mayormente un componente rojizo, por lo que se busca umbralizar principalmente el componente H del espacio HSV para aislar los tonos rojizos. Al implementar este método con las imágenes de la Figura 2, los resultados se pueden observar en la Figura 8.

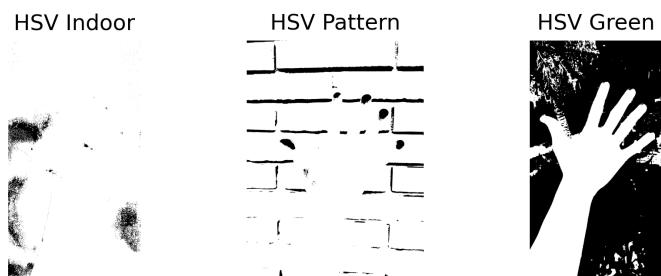


Figura 8. Segmentación en color de la mano en diferentes entornos de prueba.

Como se puede observar, este método es útil únicamente cuando el fondo tiene tonalidades diferentes al color predominante de la mano, como en entornos exteriores. Además, el tiempo de ejecución es de aproximadamente 10ms, lo que lo hace tan rápido como el método de OTSU, pero con resultados inferiores.

1.3 Superpixeles - SLIC

El último método considerado para evaluar la segmentación de la mano es el método de segmentación basada en regiones SLIC. Esto se debe a que, en [24], se implementan métodos similares para segmentar la mano, demostrando su eficacia en este contexto. El algoritmo SLIC se aplicó y comparó en tres espacios de color diferentes: RGB, HSV y YCrCb, como se muestra en la figura 9. Esta comparación permite observar cómo se comporta la segmentación y el tiempo de ejecución del algoritmo en cada espacio de color.

Al analizar la segmentación realizada por el algoritmo en los diferentes espacios de color, se observa que al aplicarlo al espacio de color RGB, la segmentación es más uniforme y logra una mejor diferenciación de los colores en comparación con los otros espacios de color. Sin embargo, el tiempo de ejecución del algoritmo es de aproximadamente 17 segundos, lo que lo hace extremadamente ineficiente para ejecutar en una CPU en una aplicación de tiempo real.

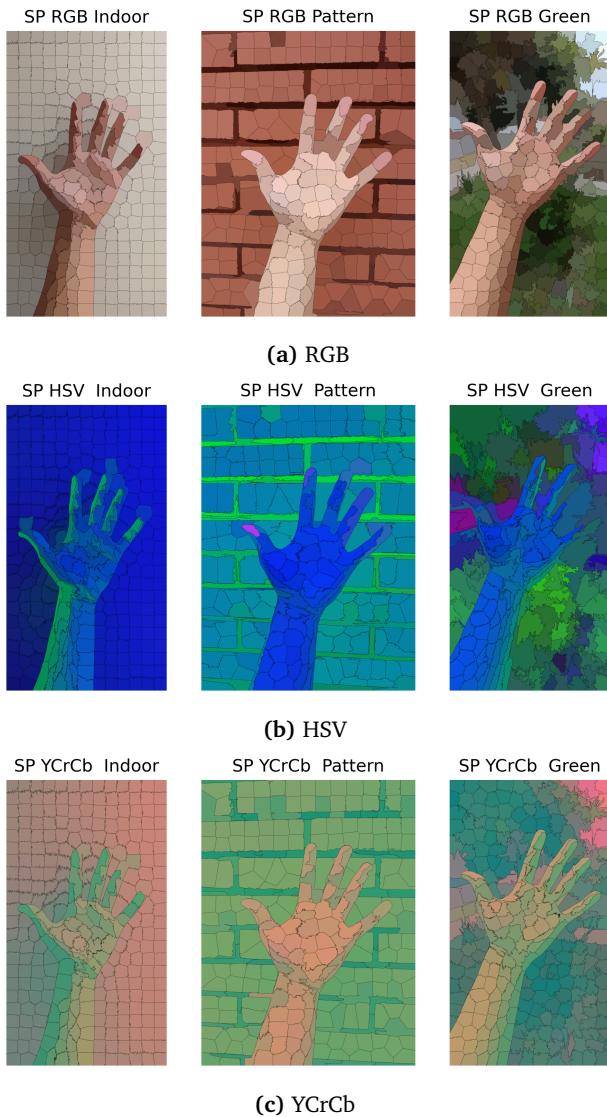


Figura 9. Algoritmo SLIC de segmentación en diferentes espacios de color.

1.4 Filtrado del ruido de fondo

Al analizar los métodos propuestos, se hace evidente que aquel que ofrece un tiempo de ejecución óptimo y buenos resultados corresponde a la umbralización OTSU en el canal Cr del espacio de color YCrCb, utilizando una normalización del histograma de 16 bins. Sin embargo, aún persisten algunos artefactos de fondo. Por ende, se propone el procesamiento que se puede observar en la Figura 11. para mejorar la segmentación. Los resultados de este proceso se muestran en la Figura 10.

De los resultados obtenidos, es crucial destacar cómo el proceso detrás de la operación cierre alternativo, la cual logra eliminar casi por completo el ruido generado por el patrón de la segunda imagen, dicha operación consiste en erosionar la imagen como si de una obtención del esqueleto de la misma se tratara, y una vez eliminado el ruido, dilatar

la imagen el mismo número de iteraciones que se erosionó. De esta manera, al obtener el contorno más grande de la imagen, se puede aislar totalmente el ruido y, aunque con información recortada, obtener la silueta de la mano de forma efectiva.

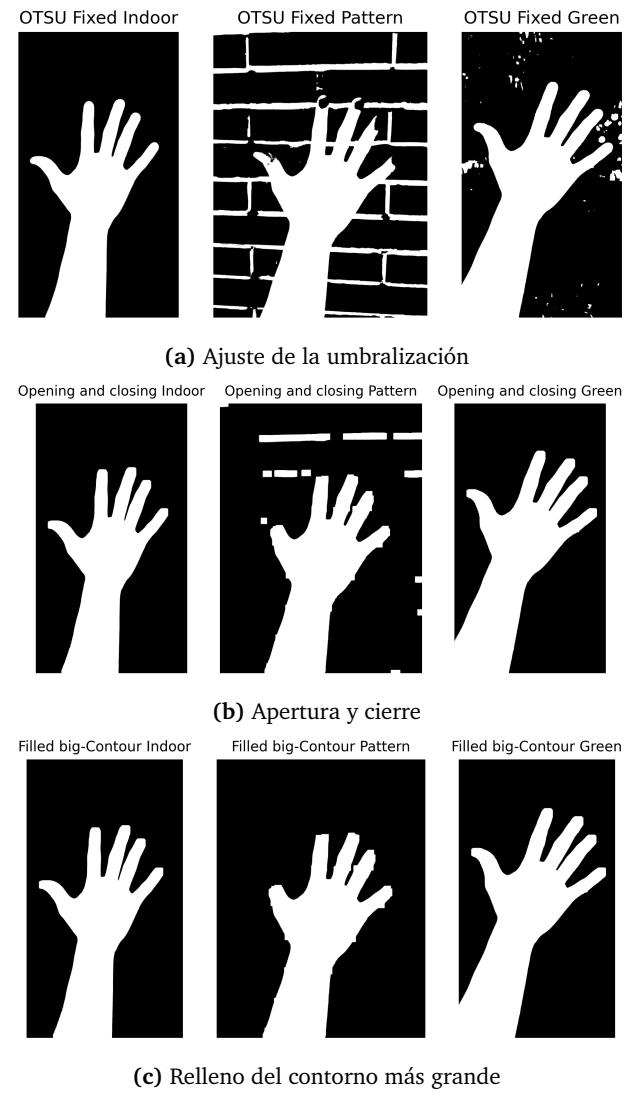


Figura 10. Mejoras propuestas para la umbralización planeteada.

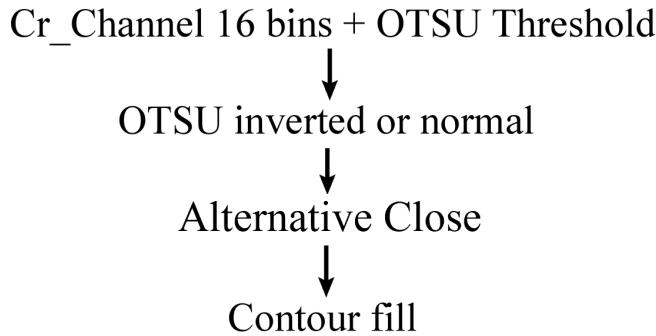


Figura 11. Algoritmo implementado para mejorar la segmentación.

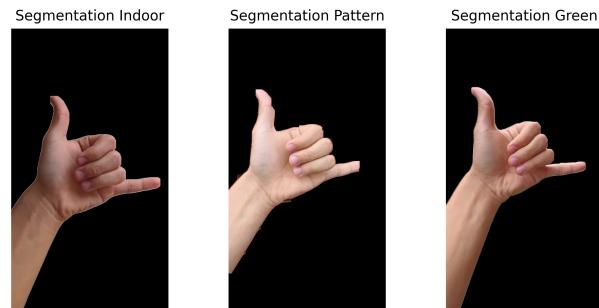


Figura 13. Resultado del algoritmo de segmentación empleado anteriormente.

2 Extracción de características

La extracción de características de la mano se revela como una tarea compleja, especialmente en la búsqueda de un algoritmo capaz de capturar múltiples características y, al mismo tiempo, ser invariante a la rotación, escala y traslación. Por lo tanto, se propuso seguir métodos observados en diferentes artículos con el fin de obtener una variedad de características siempre que sea posible.

Sin embargo, para la implementación final se utiliza un método de extracción sencillo e intuitivo, el cual presenta un alto desempeño. Como ha de esperarse, este utiliza conceptos de los métodos utilizados por los autores referenciados, por ello no se entrará en detalle a la hora de describir de las operaciones utilizadas.

2.1 A novel finger and hand pose estimation technique for real-time hand gesture recognition [22]

Para las técnicas y resultados presentados en este artículo, es crucial disponer de una imagen que muestre un gesto con los dedos flexionados. Por lo tanto, las imágenes utilizadas para evaluar los resultados son diferentes, como se puede observar en la Figura 12, y sus respectivas segmentaciones se muestran en la Figura 13.



Figura 12. Imágenes de una mano con dedos flexionados en diferentes entornos de prueba.

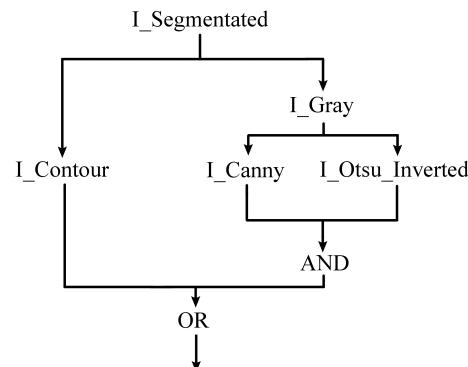
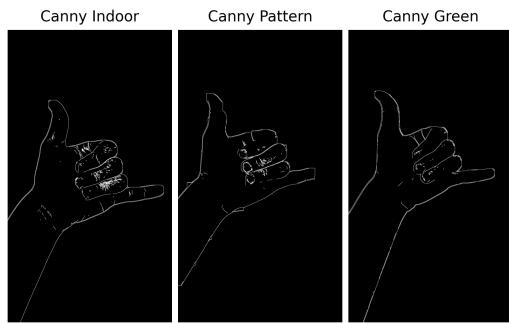


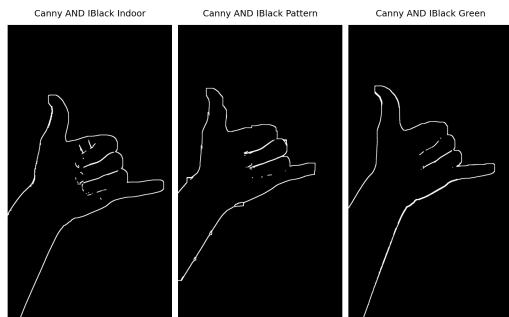
Figura 14. Arquitectura planteada en [22] para extraer el contorno de la mano y los dedos.

Una vez se dispone de las imágenes a analizar, se empleó la arquitectura propuesta en el artículo para intentar recuperar los bordes de la mano, incluyendo, si es posible, los bordes de los dedos. Esta arquitectura se presenta en la Figura 14, y los resultados del proceso se pueden observar en la Figura 15.

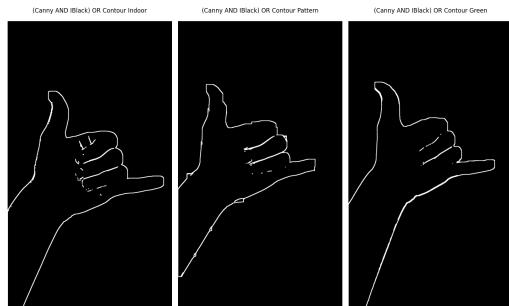
A través de los resultados se puede observar cómo el filtro Canny captura muchos detalles de contorno; sin embargo, hay información que puede afectar los resultados obtenidos. Al combinar el filtro Canny con la umbralización OTSU, se logra capturar principalmente la información de los dedos, aunque con ciertas fallas dependientes de la iluminación de la imagen. Esto se debe principalmente a las sombras generadas por los dedos sobre la palma de la mano. A pesar de estas limitaciones, este método permite obtener información sobre los dedos cuando están flexionados.



(a) Filtro Canny



(b) Otsu Invertido y Canny



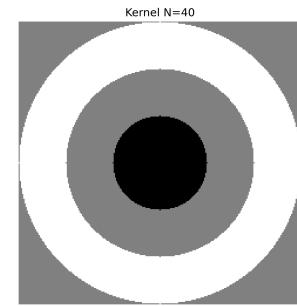
(c) Contorno + OTSU y Canny

Figura 15. Resultados obtenidos en diferentes entornos para la arquitectura propuesta en [22].

Una vez se cuenta con el contorno de los dedos, los autores de [22] proponen realizar el proceso de convolución con un Kernel invariante a la rotación que presenta la siguiente función:

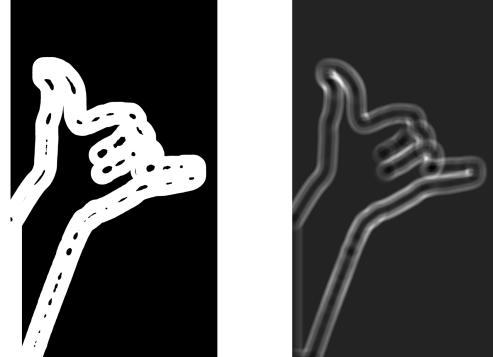
$$g(x, y) = \begin{cases} -1, & 0 \leq D(x, y) \leq N \\ 1, & 2N < D(x, y) \leq 3N \\ 0, & \text{else} \end{cases} \quad (1)$$

En el kernel propuesto, N es el grosor de los círculos y se define como $N=0.25D$, donde D se refiere al grosor de los dedos en la imagen. Esto presenta un problema, ya que hace que el kernel sea invariante a la rotación y traslación, pero no a la escala. Sin embargo, se ajustó el kernel como se muestra en la Figura 16a, obteniendo el resultado ilustrado en la Figura 16b.



(a) Kernel N=40

Contour * G - Filter2D openCV Contour * G - Convolve2D Scipy



(b) Resultado de la convolución

Figura 16. Resultados de la convolución para diferentes librerías de python.

Como se puede observar, el resultado de la convolución varía entre las bibliotecas de Python implementadas. El resultado más cercano a lo planteado en el artículo corresponde a la convolución de SciPy. Sin embargo, el principal problema de este método es su tiempo de ejecución, que es de 13.4 minutos, lo que lo hace inviable para su implementación en un CPU, descartando así esta alternativa.

2.2 Real-Time Hand Gesture Recognition Using Finger Segmentation [26]

En el enfoque de este artículo, se parte de la imagen segmentada que se observa en la Figura 7. A esta imagen se le calcula la transformada de distancia, generando el resultado mostrado en la Figura 17. A partir de este resultado, se identifica el punto central de la mano al localizar el valor máximo y su posición en la imagen.

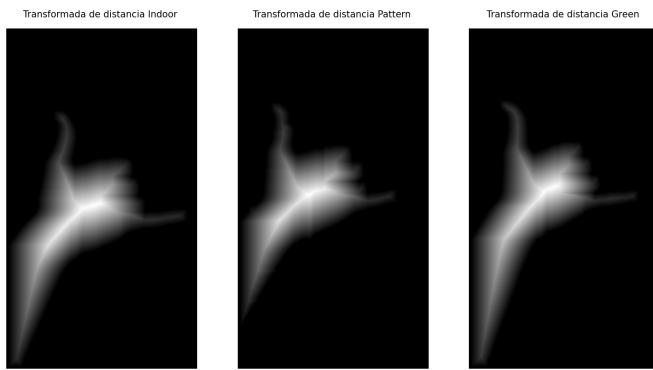
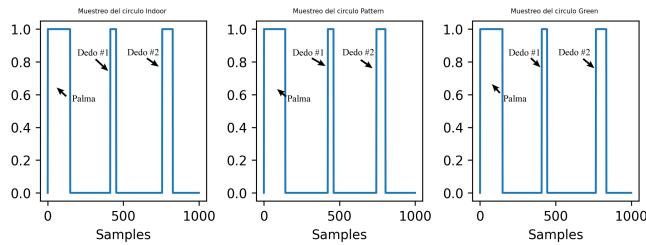
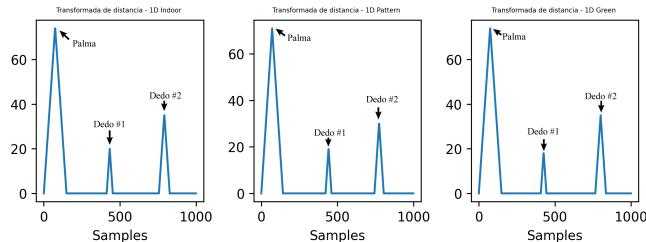


Figura 17. Transformada de distancia de la segmentación de la mano.



(a) Muestreo de la imagen segmentada a través del círculo



(b) Grosor y punto central

Figura 18. Muestreo de los dedos, determinación de su ubicación y grosor.

Con la información obtenida, se genera un círculo con un radio 1.5 veces mayor que el obtenido en la transformada de distancia, con el fin de capturar únicamente la información de los dedos y la palma de la mano. Esto se logra muestreando la información de la segmentación a lo largo del perímetro del círculo. A esta información se le calcula nuevamente la transformada de distancia, con el objetivo de identificar el punto central de los dedos o la palma y su anchura, tal como se observa en la Figura 18.

A través de toda la información mencionada, se obtienen características como el ancho y el ángulo relativo entre los dedos, lo cual permite dibujar una línea para verificar visualmente la precisión de los datos obtenidos, como se muestra en la Figura 19.

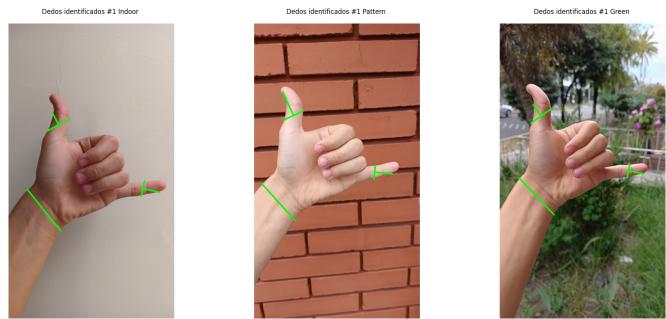
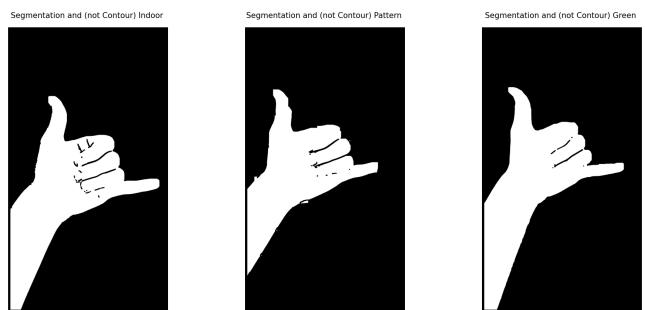
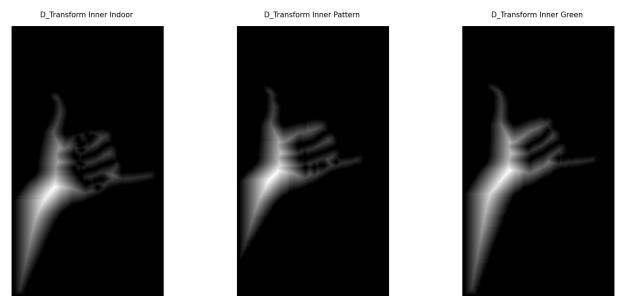


Figura 19. Dedos detectados a través del procedimiento explicado anteriormente.



(a) Segmentación sin la información del contorno de los dedos



(b) Transformada de distancia resultante

Figura 20. Alteración de la transformada de distancia para detectar los dedos restantes.

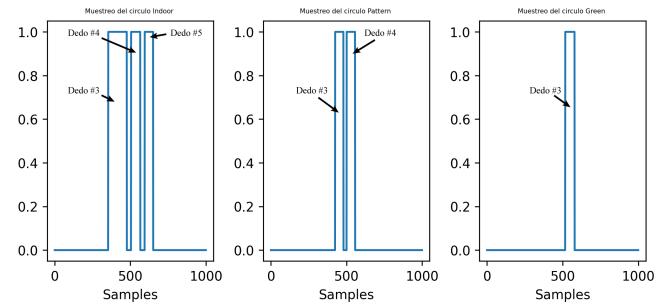
Sin embargo, esto solo es útil cuando los dedos están estirados; en el caso de dedos flexionados, como en la imagen utilizada, no se obtiene información relevante. Por ello, se propuso reutilizar el contorno obtenido del enfoque anterior (Figura 15) y restarlo de la imagen segmentada, como se observa en la Figura 20a. Esto altera notablemente la transformada de distancia obtenida, tal como se muestra en la Figura 20b.

Al analizar la imagen, se observa que la zona de mayor brillo corresponde a la palma. Por ende, se umbralizó y rellenó el contorno más grande para obtener la región de la palma y segmentarla correctamente, como se muestra en la Figura 21. Esto permite obtener parcialmente los dedos flexionados y aplicar la transformada de distancia para segmentar de

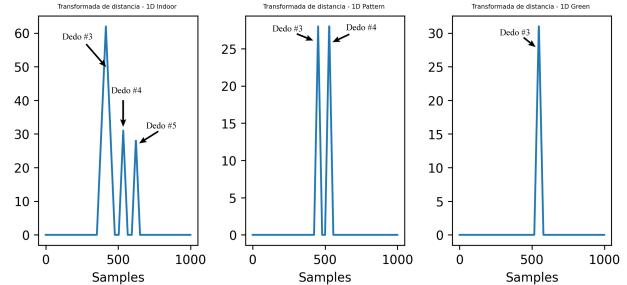
manera más efectiva los dedos, tal como se muestra en la Figura 21.



Figura 21. Segmentación de la palma.



(a) Muestreo de la imagen segmentada a través del círculo



(b) Grosor y punto central

Figura 23. Muestreo de los dedos, determinación de su ubicación y grosor.

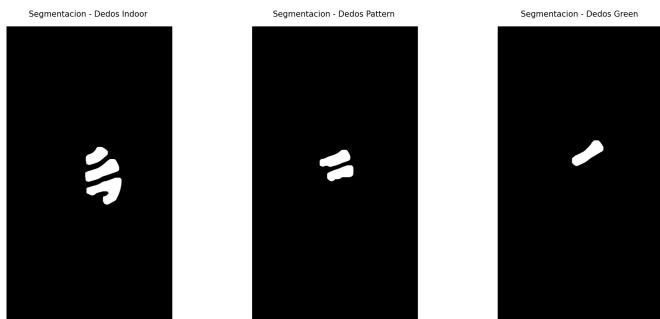


Figura 22. Segmentación de los dedos.



Figura 24. Dedos detectados a través del algoritmo en diferentes escenarios.

Una vez segmentados los dedos, se aplica la misma estrategia de mapeo a través de un círculo a la segmentación obtenida. Al resultado del muestreo se le calcula la transformada de distancia para determinar el punto central y la anchura de los dedos, como se observa en la Figura 23. Con esta información, se dibujó una línea para identificar visualmente los dedos flexionados detectados por el algoritmo, dando el resultado final mostrado en la Figura 24.

Como se puede observar, la cantidad de dedos detectados depende de la iluminación de la imagen. Sin embargo, mediante esta aproximación es posible obtener hasta 20 características, como se muestra en la Tabla 1.

Como se puede observar, las características que indican una medida de distancia están en unidades decimales. Esto se debe a que se normalizó la medida utilizando el radio de la palma para hacerla invariante a la escala. Adicionalmente, la medida de si el dedo está extendido es menor a 1 cuando no lo está y mayor a 1 cuando sí lo está. Por otra parte, los ángulos se calcularon de manera relativa al ángulo del pulgar.

Características	Indoor	Pattern	Green
Numero de dedos	5	4	3
Ángulo índice	6.49	3.96	43.96
Ángulo corazón	43.24	31.35	135.13
Ángulo anular	74.59	119.64	0
Ángulo menique	135.49	0	0
Ancho pulgar	0.685	0.356	0.338
Ancho índice	0.376	0.315	0.348
Ancho corazón	0.349	0.316	0.655
Ancho anular	0.315	0.563	0
Ancho menique	0.654	0	0
Pulgar extendido	0.9	1.4	1.4
Índice extendido	1.4	0.9	0.9
Corazón extendido	0.9	0.9	1.4
Anular extendido	0.9	1.4	0
Menique extendido	1.4	0	0
Largo Pulgar	0.898	0.444	0.405
Largo Índice	0.420	0.897	0.898
Largo Corazón	0.899	0.898	0.331
Largo Anular	0.899	0.639	0
Largo Menique	0.295	0	0

Cuadro 1. Características extraídas a través del algoritmo anteriormente mencionado.

Sin embargo, no es correcto afirmar que todas las manos presentan esta distribución entre grosor, largo y ancho. Lo que hace a estas características variantes y por tanto, no clasifican como *buenas características*. Una manera de comprobarlo es observar los resultados de los datos frente a un algoritmo de clasificación, se utiliza el método por distancias encadenadas y un clasificador bayesiano. No supervisado y supervisado respectivamente.

En la clasificación por distancias encadenadas, se observa una única clase al final del proceso, y el clasificador bayesiano entrega una precisión media del 0.23, indicios suficientes para descartar este segundo método de obtención de características.

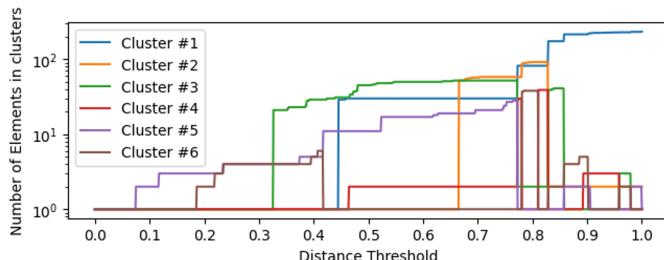


Figura 25. Clasificación mediante distancias encadenadas.

	precision	recall	f1-score	support
1.0	0.17	0.04	0.06	26
2.0	0.62	0.23	0.33	35
3.0	0.29	0.25	0.27	16
4.0	0.16	0.27	0.20	15
5.0	0.12	0.43	0.19	7
6.0	0.00	0.00	0.00	4

Figura 26. Clasificador Bayesiano.

2.3 Extracción de características mediante puntos clave de la mano

Los autores referenciados [22], [26], [27] se enfocan en la idea de que a mayor complejidad en las operaciones para identificar los dedos, obtendrán resultados más cercanos a la realidad, sin embargo, olvidan el objeto de estudio. La mano, tiene 14 puntos claves, y al someterlos a comparación, es posible obtener toda la información necesaria para su posterior clasificación e implementación de una red neuronal tipo *Perceptron Multicapa* (MLP).

Estos puntos claves son; izquierda, derecha y centro de la muñeca, izquierda y derecha de los cinco dedos y el centro de la palma.

Para identificar estos puntos, solo es necesario obtener la segmentación de la mano y su transformada de distancia, tal y como se explica en la primera parte del segundo método, explicado en la Sección 2.2. Partiendo del centro de la mano y al trazar el radio de 1.5, se obtienen tanto los puntos de izquierda y derecha de la muñeca como de los dedos, tal y como se observa en la Figura 28, el punto correspondiente al centro de la muñeca equivale a la mitad entre izquierda y derecha.

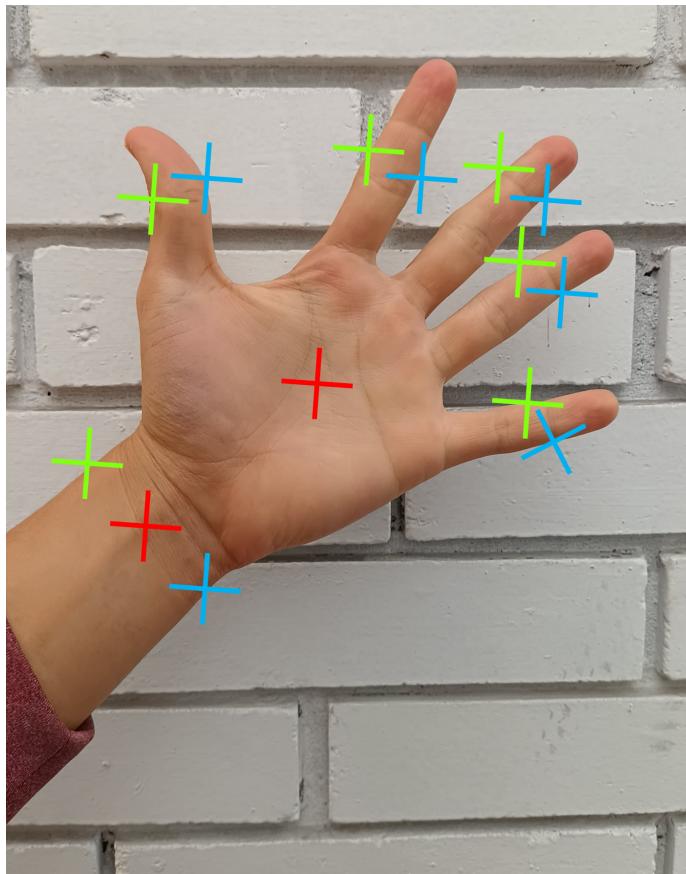


Figura 27. Puntos claves de la mano.

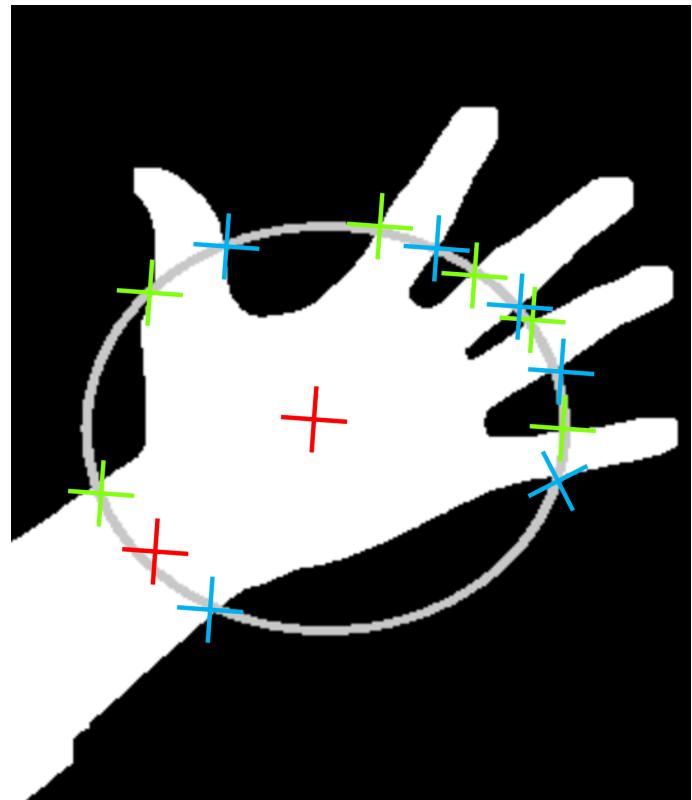


Figura 28. Circulo de radio 1.5, para la obtención de los puntos claves.

Ahora bien, lo siguiente es tomar como eje X, el eje que pasa por el centro de la muñeca y la palma, y como eje Y, el eje que pasa por la muñeca, y así poder comparar por altura o distancia en X, cada uno de los puntos claves. Para ello, se obtiene el ángulo de la recta que pasa por los centros de la muñeca y la palma, y haciendo uso de la matriz de rotación en Z, para desplazar todos los puntos, se obtiene el resultado de la Figura 29.

Seguidamente, se comparan los puntos claves de los dedos con los puntos de la muñeca, es claro que los puntos clave del pulgar siempre son superiores al punto izquierdo de la muñeca, como los puntos del meñique al punto derecho de la muñeca.

Esto facilita enormemente la identificación del gesto en cuestión, pero aparecen casos especiales, en primer lugar, el caso de la mano derecha, para ello, antes de proceder con la comparativa de los puntos claves, se hace uso de la geometría de rectas para encontrar el pulgar, las rectas encontradas se someten a la operación AND, y dependiendo de cual halla cruzado con el pulgar, se sabrá de que mano se trata. Ver Figura 30.

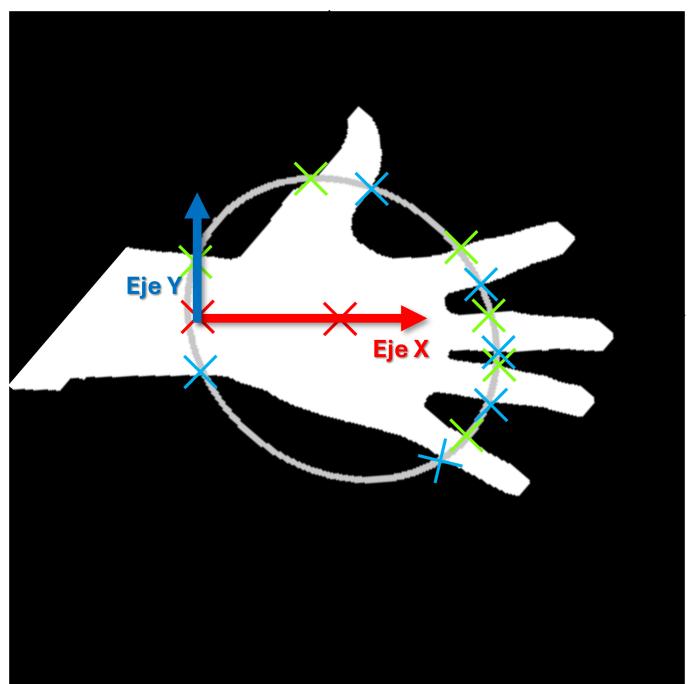


Figura 29. Puntos desplazados.

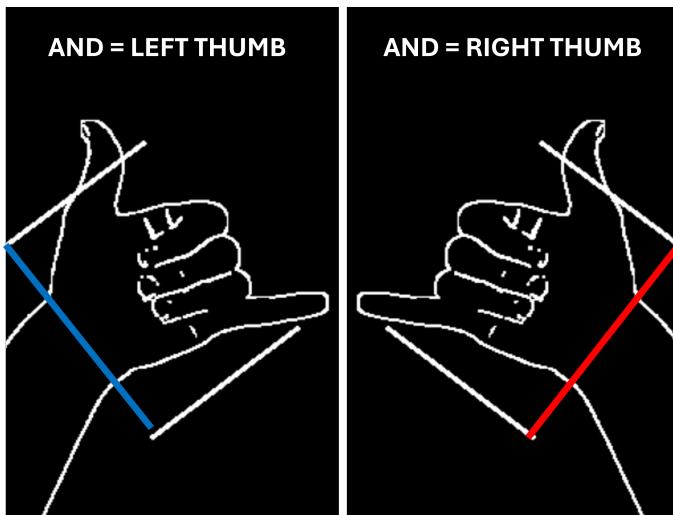


Figura 30. Rectas para detectar de que mano se trata.

En caso de que ambas operaciones den como resultado vacío, es decir que la pose en cuestión no tiene pulgar, se halla con el índice, como se observa en la Figura 29, cuando la mano es izquierda el punto izquierdo del índice es superior al punto izquierdo de la muñeca, caso invertido en caso de la mano derecha, de igual manera si la pose no contuviera el índice, se podría hallar comparando los puntos del dedo medio, etc.

Si el lector es observador, puede concluir fácilmente que existen casos arbitrarios muy sencillos de clasificar, si el círculo trazado solo encuentra dos puntos, quiere decir que se trata de un puño y no es necesario identificar de qué mano se trata, si detecta 12 puntos, quiere decir que es la mano abierta, si no se detecta pulgar y los puntos detectados son 10, quiere decir que es el gesto de 4 dedos. Y así, se pueden ir identificando todos los gestos, sin necesidad de operaciones que compliquen aún más el procedimiento.

Así las características obtenidas, resultan ser invariantes a la escala, rotación y traslación. Ver Tabla 2, se incluyeron otras 11 características de la misma naturaleza, los 7 momentos de Hu y 4 características geométricas, normalizadas con el perímetro y área del círculo resultando de la transformada de distancia.

Frente a las pruebas de clasificación el método presenta un desempeño excelente, incluso con 7 clases al tiempo. La clasificación por distancias encadenadas, encuentra las 7 clases rápidamente y estás se mantienen constantes, ver Figura 31. Por otra parte el clasificador bayesiano, presenta una precisión del 100%, ver Figura 32.

N	Característica	Valor	Valor Número
1	Pulgar Extendido	FALSE	0
2	Indice Extendido	TRUE	1
3	Medio Extendido	FALSE	0
4	Anular Extendido	FALSE	0
5	Meñique Extendido	FALSE	0
6	Pulgar Contraido	TRUE	1
7	Indice Contraido	FALSE	0
8	Medio Contraido	TRUE	1
9	Anular Contraido	TRUE	1
10	Meñique Contraido	TRUE	1

Cuadro 2. Características obtenidas

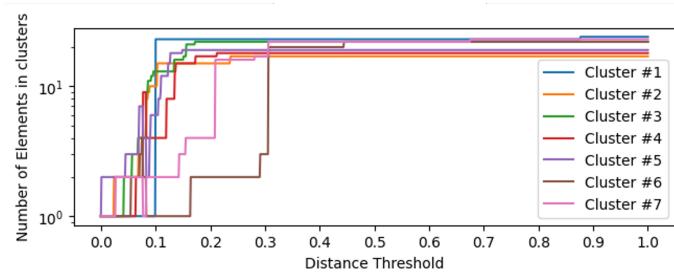


Figura 31. Clasificación mediante distancias encadenadas.

	precision	recall	f1-score	support
1	1.00	1.00	1.00	11
2	1.00	1.00	1.00	8
3	1.00	1.00	1.00	11
4	1.00	1.00	1.00	8
5	1.00	1.00	1.00	7
6	1.00	1.00	1.00	12
7	1.00	1.00	1.00	6

Figura 32. Clasificador Bayesiano.

Se procede con el entrenamiento de la red neuronal, en total de utilizaron 210 imágenes, 30 por cada gesto. Se establecieron 2 capas de neuronas, 70 y 7 neuronas respectivamente y como función de activación, la función ReLU (Rectified Linear Unit).

Conclusiones

- Ante el planteamiento de cualquier problema de caracterización de objetos no siempre es necesario hacer uso de una gran cantidad de estrategias y técnicas para lograr los resultados idóneos, si se realiza un análisis correcto, el problema se puede ver altamente simplificado.

- En un principio, la teoría expuesta en [27] acerca del reconocimiento de la piel en imágenes puede prestarse para interpretaciones erróneas, si bien el desempeño de la reducción de bins es sobresaliente, en casos de estudio donde no es relevante obtener toda la información de la piel, como en hand tracking donde solo se necesita la silueta del objeto, este desempeño es innecesario. Pasando por alto las ventajas que tienen los espacios de color, YCrCb tiene un desempeño excelente en ambientes con mucho ruido y HSV de igual manera en interiores.
- Los kernels son una parte esencial del procesamiento de imágenes y visión por computadora, ya que permiten realizar filtros y transformaciones en las imágenes, para resaltar características específicas, mediante la operación de convolución. Operación por la cual es posible la identificación de los dedos en gestos con manos.
- Este tipo de redes neuronales, se encuentran limitadas a casos discretos como estos, ya que presentan un bajo rendimiento frente a cambios bruscos en la textura del fondo, iluminación, tonalidad de la piel, entre otros. Corroborando así, lo expuesto en la teoría de un perceptron multicapa, los cuales presentan un buen desempeño con ruido mínimo y pocas muestras.

Referencias

- M. Kakizaki, A.S.M. Miah, K. Hirooka, and J. Shin. Dynamic Japanese sign language recognition through hand pose estimation using effective feature extraction and classification approach. *Sensors*, 24, 2024. doi: 10.3390/s24030826.
- H. Alsolai, L. Alsolai, F.N. Al-Wesabi, M. Othman, M. Rizwanullah, and A.A. Abdelmageed. Automated sign language detection and classification using reptile search algorithm with hybrid deep learning. *Helijon*, 10, 2024. doi: 10.1016/j.helijon.2023.e23252.
- R. Sutjiadi. Android-based application for real-time Indonesian sign language recognition using convolutional neural network. *TEM Journal*, 12:1541–1549, 2023. doi: 10.18421/TEM123-35.
- B.A. Dabwan, M.E. Jadhav, H.A. Abosaq, F.A. Olayah, M. Al Yami, and Y.A. Abdelrahman. Real-time system for translating American sign language to text using robust techniques. 2023. ISBN 9798350306927. doi: 10.1109/ICRASET59632.2023.10420110.
- T. Watanabe, M. Maniruzzaman, M.A.M. Hasan, H.-S. Lee, S.-W. Jang, and J. Shin. 2d camera-based air-writing recognition using hand pose estimation and hybrid deep learning model. *Electronics (Switzerland)*, 12, 2023. doi: 10.3390/electronics12040995.
- P. Shukla and P. Das. Enhancing human-computer interaction: Hand detection for air writing utilizing numpy and opencv. pages 517–521, 2023. ISBN 9798350342338. doi: 10.1109/ICTACS59847.2023.10390179.
- S.K. Baruah, U. Konwar, R. Mahanta, A. Boruah, and D. Sarma. A contactless control mechanism for computerized systems using hand gestures. pages 246–250, 2023. ISBN 9798350325881. doi: 10.1109/ASPCON59071.2023.10396173.
- G. Csonka, M. Khalid, H. Rafiq, and Y. Ali. Ai-based hand gesture recognition through camera on robot. pages 256–261, 2023. ISBN 9798350395785. doi: 10.1109/FIT60620.2023.00054.
- S.S. Sugantha Mallika, M. Priyadharsini, S. Samritha, C. Sowmiya, and B. Nikitha. Hand gesture recognition using convolutional neural networks. pages 249–255, 2023. ISBN 9798350340235. doi: 10.1109/ICACRS58579.2023.10404885.
- Q. Wang and Z. Xie. Arias: An ar-based interactive advertising system. *PLoS ONE*, 18, 2023. doi: 10.1371/journal.pone.0285838.
- E. Aksoy, A.D. Çakir, B.A. Erol, and A. Gümüş. Real time computer vision based robotic arm controller with ros and gazebo simulation environment. 2023. ISBN 9798350360493. doi: 10.1109/ELECO60389.2023.10416078.
- N. Kumar, H. Dalal, A. Ojha, A. Verma, and M. Kaur. Real-time hand gesture recognition for device control: An opencv-based approach to shape-based element identification and interaction. pages 1537–1541, 2023. ISBN 9798350342338. doi: 10.1109/ICTACS59847.2023.10390298.
- D. Wu, J. Huang, M. Zheng, and Y. Li. Virtual model interaction based on single rgb camera. pages 298–301, 2023. ISBN 9798350380859. doi: 10.1109/NTCI60157.2023.10403685.
- V.L. Adluri, P. Kadiyala, S. Gopu, and S. Jangiti. Virtual mouse with hand gestures using machine learning. pages 1564–1568, 2023. ISBN 9798350313987. doi: 10.1109/ICSCNA58489.2023.10370381.
- N. Zerrouki, F. Harrou, A. Houacine, R. Bouarroudj, M.Y. Chérif, A.A. Zouina, and Y. Sun. Deep learning for hand gesture recognition in virtual museum using wearable vision sensors*. *IEEE Sensors Journal*, 2024. doi: 10.1109/JSEN.2024.3354784.
- A. Das, K. Maitra, S. Roy, B. Ganguly, M. Sengupta, and S. Biswas. Development of a real time vision-based hand gesture recognition system for human-computer interaction. pages 294–299, 2023. ISBN 9798350325881. doi: 10.1109/ASPCON59071.2023.10396583.
- A. Yıldız, N.G. Adar, and A. Mert. Convolutional neural network based hand gesture recognition in sophisticated background for humanoid robot control. *International Arab Journal of Information Technology*, 20:368–375, 2023. doi: 10.34028/iajat/20/3/9.

- [18] R. Özakar and E. Gedikli. Evaluation of hand washing procedure using vision-based frame level and spatio-temporal level data models. *Electronics (Switzerland)*, 12, 2023. doi: 10.3390/electronics12092024.
- [19] S. Karedoudra and R.K. Veerasha. Hand gestures and machine learning: A novel path to number prediction. 2023. ISBN 9798350306927. doi: 10.1109/ICRASET59632.2023.10419929.
- [20] G. Huang, S.N. Tran, Q. Bai, and J. Alty. Real-time automated detection of older adults' hand gestures in home and clinical settings. *Neural Computing and Applications*, 35:8143–8156, 2023. doi: 10.1007/s00521-022-08090-8.
- [21] F.A. Farid, N. Hashim, J.B. Abdullah, M.R. Bhuiyan, M. Kai-ranbay, Z. Yusoff, H.A. Karim, S. Mansor, M.D.T. Sarker, and G. Ramasamy. Single shot detector cnn and deep dilated masks for vision-based hand gesture recognition from video sequences. *IEEE Access*, 12:28564–28574, 2024. doi: 10.1109/ACCESS.2024.3360857.
- [22] Y. Zhou, G. Jiang, and Y. Lin. A novel finger and hand pose estimation technique for real-time hand gesture recognition. *Pattern Recognition*, 49:102–114, 2016. doi: 10.1016/j.patcog.2015.07.014.
- [23] R. Zahra, A. Shehzadi, M.I. Sharif, A. Karim, S. Azam, F. De Boer, M. Jonkman, and M. Mehmood. Camera-based interactive wall display using hand gesture recognition. *Intelligent Systems with Applications*, 19, 2023. doi: 10.1016/j.iswa.2023.200262.
- [24] H. Liang, J. Yuan, and D. Thalmann. Parsing the hand in depth images. *IEEE Transactions on Multimedia*, 16:1241–1253, 2014. doi: 10.1109/TMM.2014.2306177.
- [25] R.M. Gurav and P.K. Kadbe. Real time finger tracking and contour detection for gesture recognition using opencv. pages 974–977, 2015. ISBN 9781479971657. doi: 10.1109/IIC.2015.7150886.
- [26] Z.-H. Chen, J.-T. Kim, J. Liang, J. Zhang, and Y.-B. Yuan. Real-time hand gesture recognition using finger segmentation. *Scientific World Journal*, 2014, 2014. doi: 10.1155/2014/267872.
- [27] M.J. Jones and J.M. Rehg. Statistical color models with application to skin detection. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 1, pages 274–280 Vol. 1, 1999. doi: 10.1109/CVPR.1999.786951.