



Universidad Nacional de Rosario

Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Tecnatura

Universitaria en Inteligencia Artificial

Procesamiento de Imágenes I - IA 4.4

TRABAJO PRÁCTICO N°2 - Año 2024 - 2° Semestre

Alumnos:

Antuña, Franco A-4637/1

Gallardo, Jonatan G-5970/6

Orazi, Roberto O-1815/5

Problema 1 - Deteccion y clasificacion de monedas y datos.....	3
Descripción del problema.....	4
Implementación.....	4
Conclusión final.....	9
Problema 2 - Detección de patentes.....	10
Descripción del problema.....	10
Implementación.....	10
Problemas.....	13
Conclusión Final.....	14

Problema 1 - Deteccion y clasificacion de monedas y dados

En la figura 1 se muestra la imagen del archivo *monedas.jpg*, la cual consiste en un fondo de intensidad no uniforme sobre el cual se hallan dados y monedas de distinto valor y tamaño.

Se debe elaborar un algoritmo capaz de procesar dicha imagen y resolver los siguientes puntos sobre ella:

- A. Procesar la imagen de manera de segmentar las monedas y los dados de manera automática.
- B. Clasificar los distintos tipos de monedas y realizar un conteo automático
- C. Determinar el número del valor que representa la cara superior de cada dado y realizar un conteo automático.

Aviso: En cada punto, el script elaborado debe informar y mostrar los resultados en cada una de las etapas de procesamiento



Figura 1: Imagen con monedas y dados del archivo *monedas.jpg*.

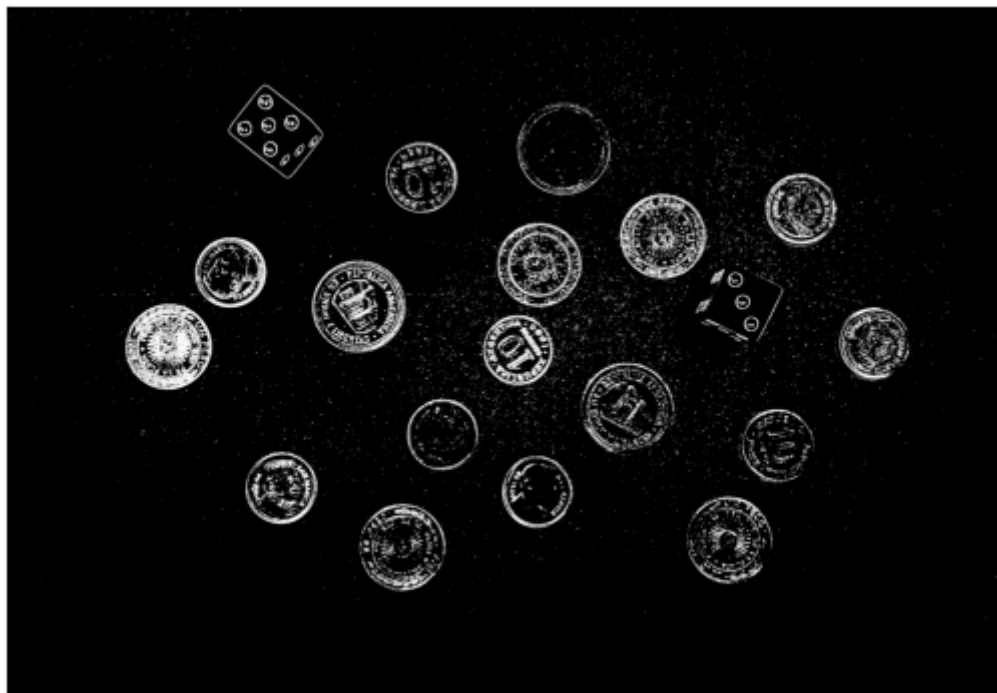
Descripción del problema

El objetivo principal del proyecto es desarrollar un algoritmo capaz de procesar una imagen denominada *monedas.jpg*, la cual contiene monedas de distintos valores y tamaños, así como dados con caras visibles en un fondo de intensidad no uniforme. Se busca segmentar automáticamente estos objetos, clasificar las monedas según su valor nominal, determinar el número visible en la cara superior de cada dado y realizar un conteo total. Para cada paso, el script debe generar resultados visuales y numéricos.

Implementación

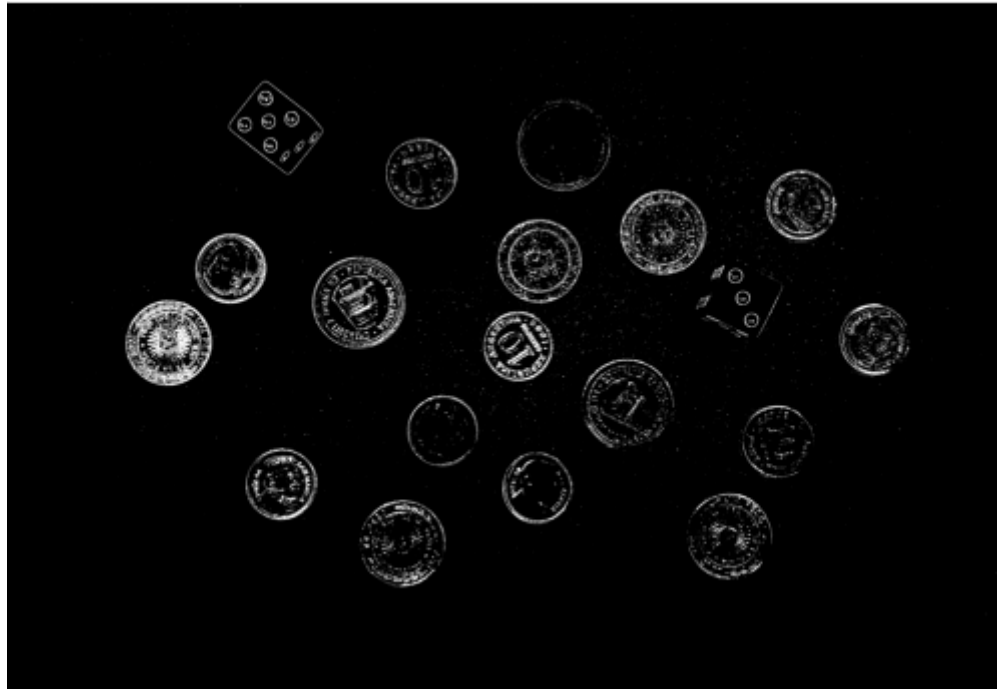
Para abordar este problema, se siguieron varios pasos en el procesamiento de la imagen, empleando herramientas de visión por computadora y transformaciones morfológicas.

En primer lugar, se carga la imagen y se convierte a escala de grises para simplificar las operaciones posteriores. Se aplica un gradiente morfológico que resalta los bordes de los objetos mediante transformaciones que calculan diferencias entre dilataciones y erosiones. Este procedimiento es clave para separar los objetos del fondo, particularmente en imágenes con gradientes de intensidad no uniformes. Posteriormente, se utiliza el método de Otsu para realizar una umbralización automática, convirtiendo la imagen en binaria y facilitando la segmentación de las regiones de interés.

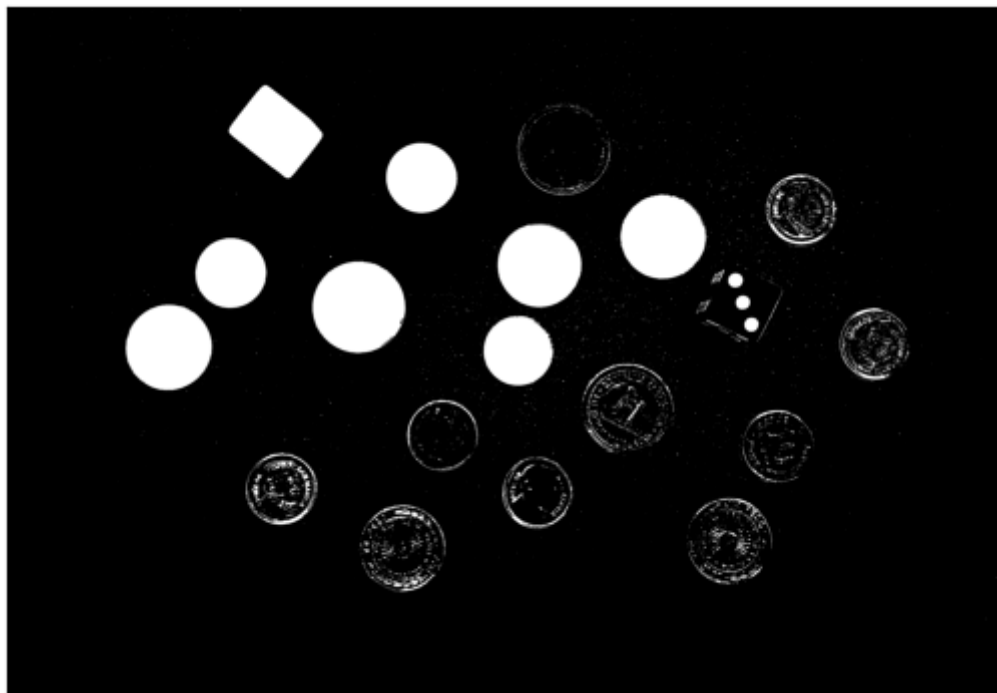


Para reducir el ruido y preservar las características esenciales de los objetos, se implementa un filtro contraharmónico. Este filtro es eficaz para eliminar ruido de tipo sal sin afectar los bordes, lo cual resulta fundamental en imágenes donde los objetos tienen bordes suaves o finos. Una vez limpiada la imagen, se procede a rellenar huecos en las regiones segmentadas mediante un procedimiento de reconstrucción morfológica iterativa. Este paso

garantiza que las monedas y dados queden completamente definidos, incluso si inicialmente presentan interrupciones en sus bordes.

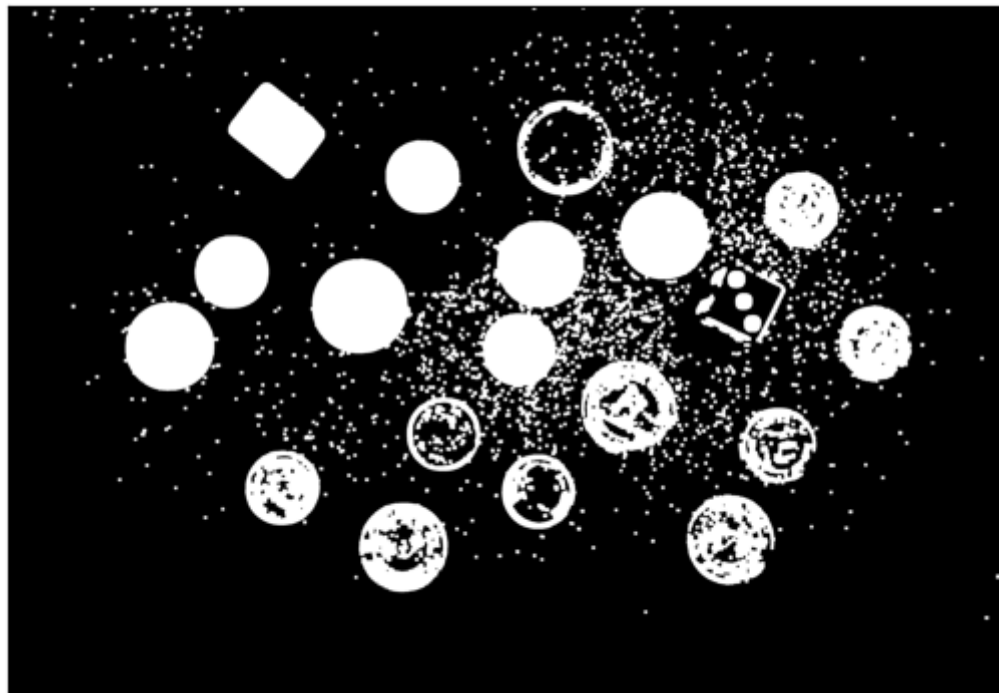


Se utilizó la función `imfillhole` sobre la imagen binarizada filtrada (`imgn_ch_filt8`) para rellenar los huecos presentes dentro de las regiones segmentadas. Esto permite que las áreas correspondientes a las monedas y dados se completen correctamente, eliminando pequeñas discontinuidades internas que podrían afectar el análisis posterior. El resultado es una imagen donde los objetos están completamente definidos, facilitando su detección en etapas posteriores.



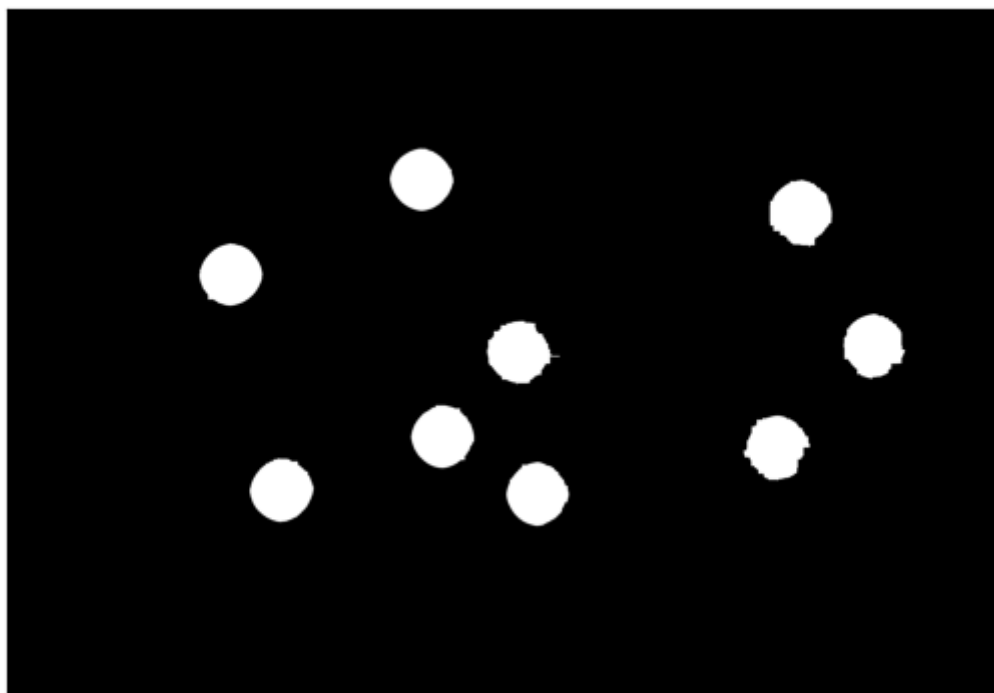
Se aplicó un cierre morfológico (morphologyEx con el tipo MORPH_CLOSE) a la imagen resultante del paso anterior (img_fh). Este proceso utiliza un elemento estructurante de 17x17 píxeles y tiene como objetivo suavizar los bordes de los objetos segmentados, además de cerrar pequeños espacios entre elementos contiguos. Esto asegura que objetos cercanos, como monedas que podrían estar ligeramente superpuestas, se consideren como una única región conectada.

Finalmente, se aplicó una dilatación morfológica sobre la imagen resultante del cierre (img_modif). Esta etapa expande las regiones segmentadas utilizando un kernel de 13x13 píxeles, lo que permite mejorar la conectividad de las áreas y resaltar más claramente los contornos de los objetos. Este paso es especialmente útil para asegurar que las monedas y los dados se representen como regiones compactas y fácilmente identificables.

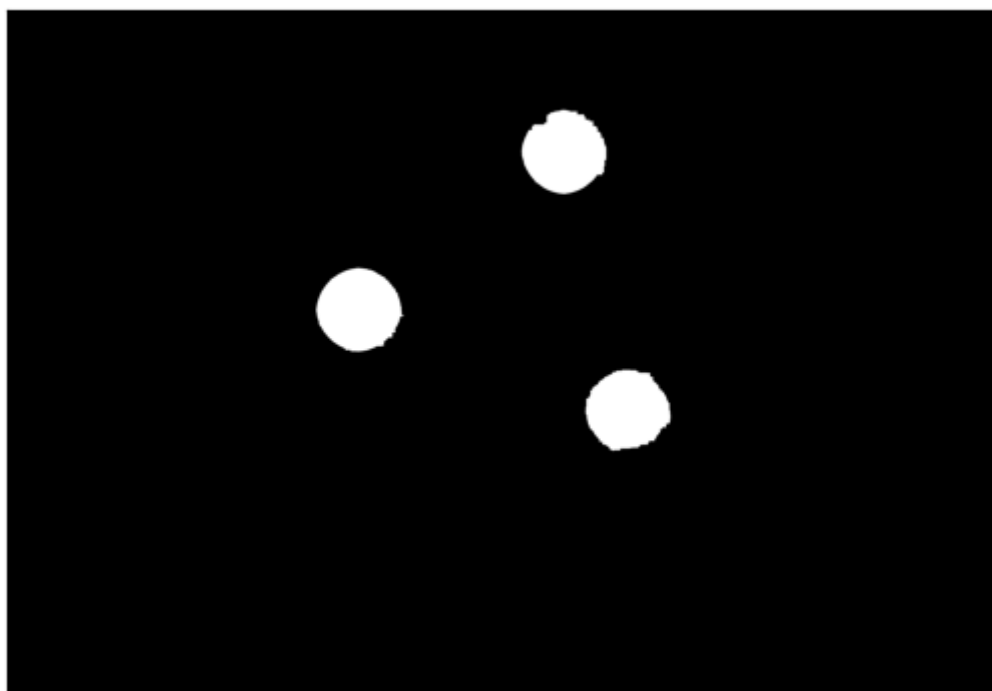


El análisis de componentes conectados permite identificar las regiones segmentadas en la imagen. Utilizando propiedades como el área de cada componente, se clasifican las monedas en diferentes categorías según su tamaño. Por ejemplo, las monedas de 10 centavos corresponden a áreas específicas que se encuentran en un rango entre 40,000 y 47,000 píxeles, mientras que las de 50 centavos y 1 peso ocupan rangos diferentes. Este criterio basado en áreas permite contar con precisión la cantidad de monedas de cada tipo y calcular su valor total.

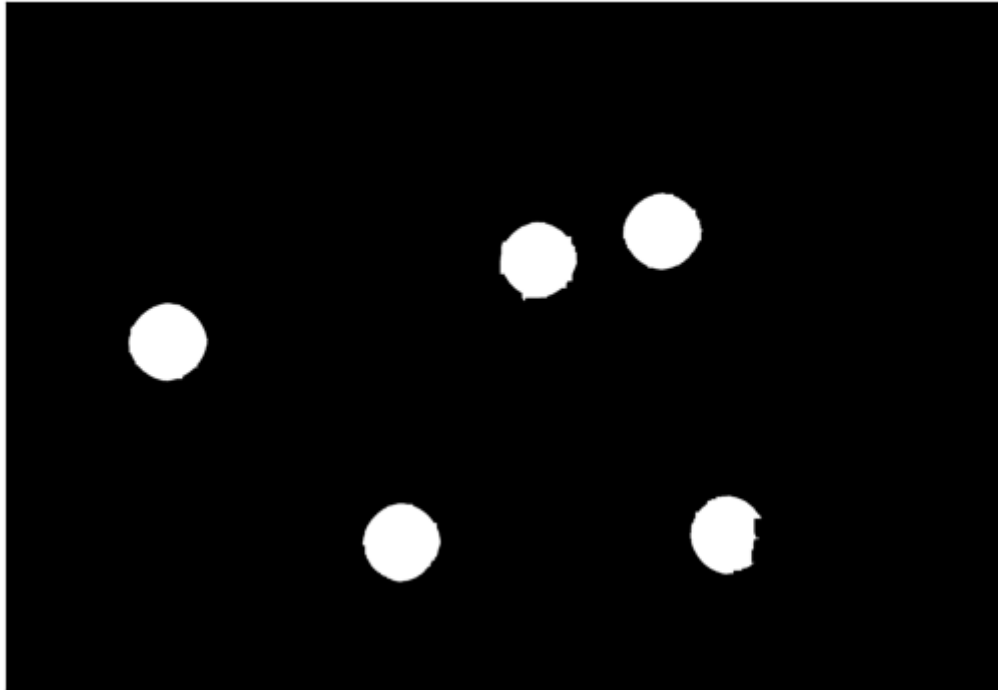
10 Centavos



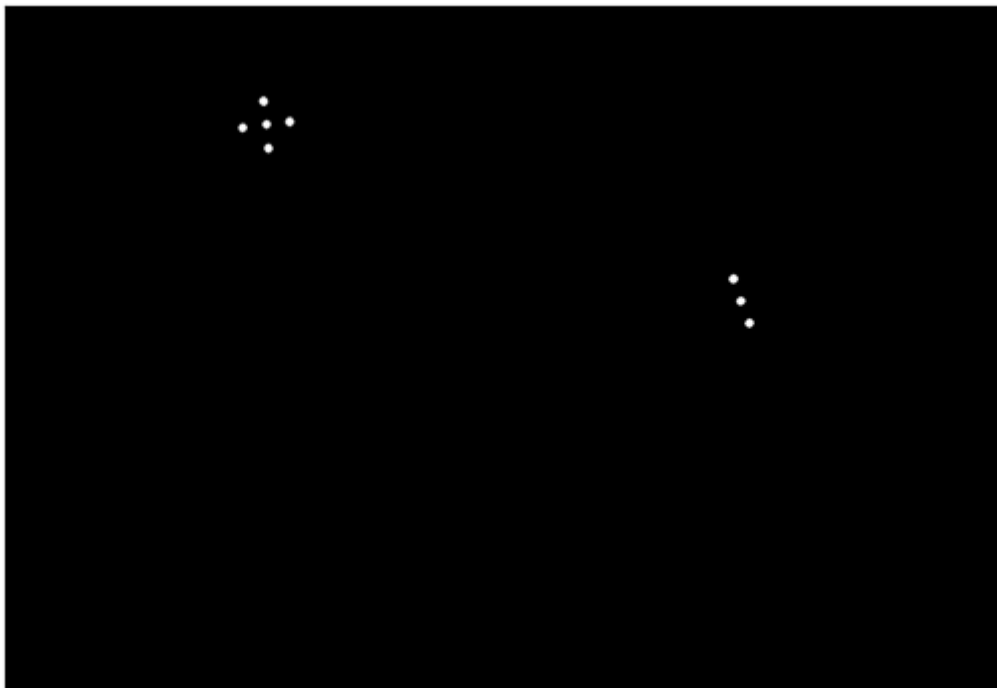
50 Centavos



1 Peso



Para detectar los dados, se aplicaron operaciones de erosión con un kernel grande para aislar las caras visibles de estos objetos. Luego, se analizaron nuevamente los componentes conectados, detectando las regiones correspondientes a los puntos en la cara superior de cada dado. Este análisis permitió calcular la suma total de los valores visibles.



Conclusión final

El procesamiento realizado en la imagen ha cumplido exitosamente con los objetivos planteados, logrando identificar y clasificar correctamente los objetos presentes. Los resultados obtenidos coinciden con las expectativas iniciales, mostrando la eficacia de las técnicas de preprocesamiento, segmentación y análisis de componentes conectados implementadas.

En total, se detectaron 9 monedas de 10 centavos, 3 monedas de 50 centavos y 5 monedas de un peso, sumando un valor total de \$7.40, que corresponde al cálculo esperado. Además, se identificaron correctamente 2 dados, con una suma visible en sus caras de 8 puntos.

Estos resultados validan la robustez de las estrategias aplicadas, incluyendo la limpieza de ruido, el relleno de huecos y la clasificación basada en áreas específicas. Esto demuestra que el flujo de procesamiento es adecuado para aplicaciones similares, donde la precisión en la identificación de objetos y la extracción de características relevantes es crucial.

El éxito del análisis también sugiere que este enfoque podría adaptarse a tareas relacionadas, como la detección y clasificación de objetos en otros contextos visuales, siempre que se ajusten los parámetros a las características particulares de las imágenes a analizar.

Problema 2 - Detección de patentes

En la figura 2 se muestra una de las doce imágenes (archivos `img<id>.png`), las cuales representan la vista anterior o posterior de diversos vehículos. En cada una de ellas se puede visualizar las correspondientes patentes.

Se debe elaborar un algoritmo capaz de procesar cada una de estas imágenes y resolver los siguientes puntos en cada uno de ellas:

- A. Detectar automáticamente la placa patente y segmentar la misma. Informar las distintas etapas de procesamiento y mostrar los resultados de cada etapa
- B. Implementar un algoritmo de procesamiento que segmenta los caracteres de la placa patente detectada en el punto anterior. Informar las distintas etapas de procesamiento y mostrar los resultados de cada etapa.



Figura 2: Ejemplo de una de las imágenes (archivo `img05.png`) de la carpeta *Patentes*.

Descripción del problema

El propósito principal del script es procesar imágenes de vehículos para identificar y extraer posibles regiones que correspondan a las patentes, utilizando técnicas de procesamiento digital de imágenes y segmentación.

Implementación

El flujo comienza con la carga y preprocesamiento de las imágenes. Cada imagen se lee en color y se convierte a escala de grises para simplificar los cálculos posteriores. A continuación, se aplica un umbral binario para segmentar las regiones de interés,

destacando las áreas claras frente a un fondo oscuro. Este paso inicial establece la base para los análisis posteriores.

imagen binaria



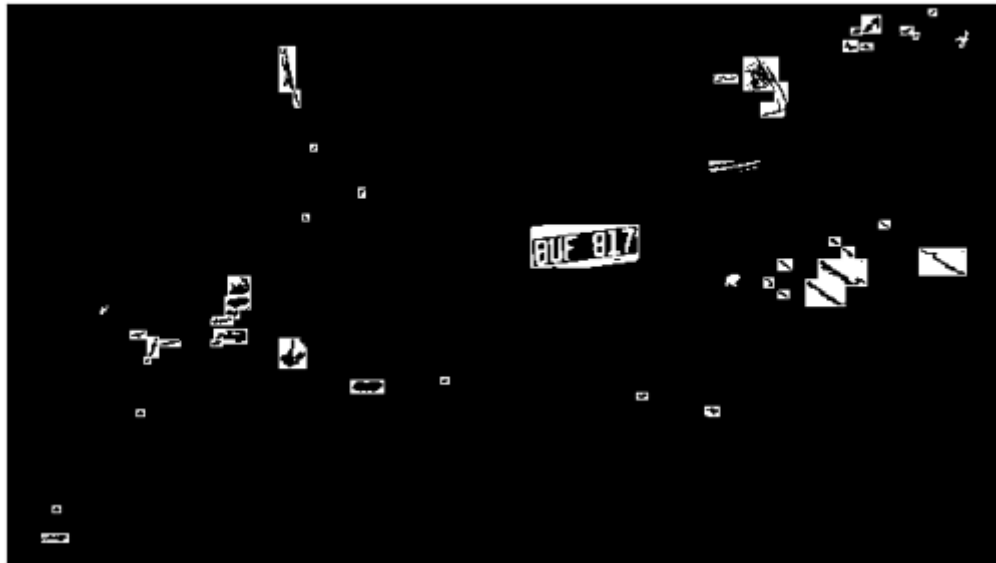
Una vez procesada la imagen binaria, se lleva a cabo un análisis de componentes conectados para identificar áreas conectadas dentro de la imagen. Estas áreas representan posibles regiones de interés, que son filtradas según su tamaño, eliminando tanto las regiones demasiado pequeñas como las demasiado grandes. Este filtrado garantiza que las regiones consideradas tengan dimensiones compatibles con las de una patente.

imagen filtrada por el area de los componentes



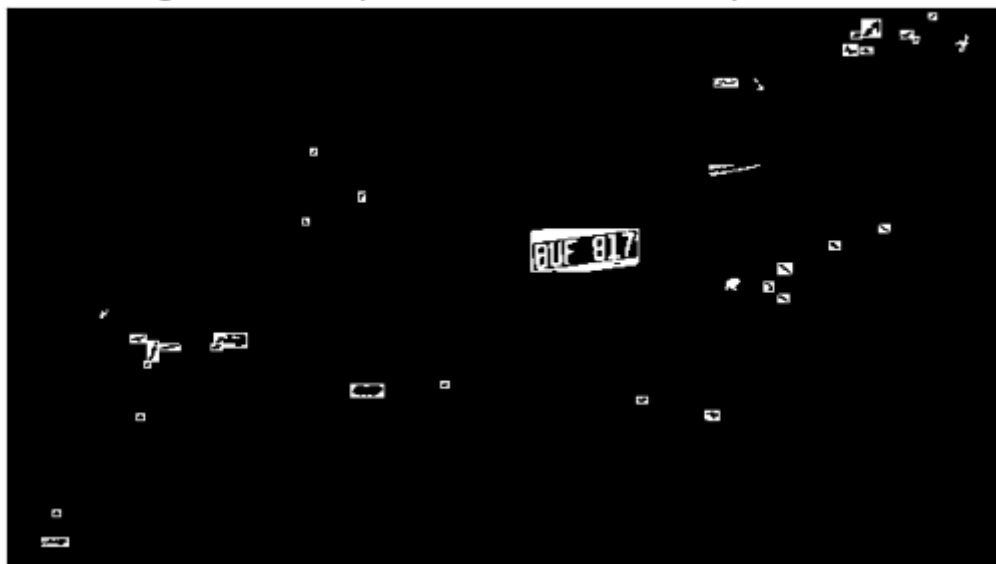
Después del filtrado inicial, se identifican los contornos de las regiones restantes. Se aplica un análisis de las relaciones de aspecto para eliminar aquellas regiones cuyas dimensiones no sean consistentes con las proporciones típicas de una patente. Este paso adicional de filtrado refina las regiones detectadas y reduce los falsos positivos.

Imagen con Regiones Binarias Filtradas



En las etapas finales del procesamiento, se realiza un análisis de proximidad entre las regiones seleccionadas. Se buscan agrupaciones de regiones cercanas para identificar posibles configuraciones que correspondan a patentes completas. Estas agrupaciones se filtran utilizando criterios de cercanía y alineación, con lo que se obtienen las regiones finales que probablemente correspondan a las patentes.

Imagen filtrada por el area de los componentes 2



Finalmente, las regiones detectadas se visualizan en imágenes generadas durante cada paso del proceso, proporcionando una representación clara del progreso y los resultados del análisis.

Patentes del vehiculo



Problemas

Durante el desarrollo y ejecución del código, se identificaron los siguientes problemas y desafíos técnicos:

Cálculo de Áreas: Hubo dificultades al definir y ajustar los umbrales de área para filtrar componentes conectados. Esto implicó un proceso iterativo para determinar los valores que mejor equilibraran la detección correcta y la eliminación de falsos positivos.

Optimización de la Etapa Final (Filtrado por Bounding Boxes Cercanos): La última etapa del procesamiento, que se encarga de agrupar los bounding boxes más cercanos para formar una patente completa, resultó ser la más compleja y demandante en términos de tiempo de ejecución. Esto podría mejorarse con optimizaciones en el algoritmo de agrupamiento o implementando técnicas más eficientes.

Problemas en la Identificación Completa de las Patentes: En algunos casos, no fue posible detectar completamente las patentes. Por ejemplo, en la patente identificada como "9", no se detectaron las primeras dos letras. Esto puede deberse a configuraciones específicas de los umbrales, relaciones de aspecto, o características de la imagen binaria resultante.

Variabilidad en los Resultados Según el Color del Vehículo: El color del auto afectó significativamente la calidad de las imágenes binarias generadas. En algunas imágenes, esto permitió identificar la patente completa, mientras que en otras solo se detectaron parcialmente las letras y los números. Esto sugiere la necesidad de preprocesamientos adicionales o ajustes dinámicos según las condiciones de iluminación y contraste.

Conclusión Final

El algoritmo desarrollado para la detección y segmentación de patentes vehiculares ha demostrado ser eficaz en casi la totalidad de las imágenes procesadas. A pesar de los desafíos técnicos identificados, como la sensibilidad a ciertas características de las imágenes (color del vehículo, iluminación, y contraste) y la complejidad de la etapa final de agrupamiento, los resultados obtenidos validan el enfoque implementado.

En términos generales, el sistema logra:

- Identificar componentes relevantes de las patentes en la mayoría de las imágenes, filtrando adecuadamente elementos no deseados.
- Detectar y reconstruir correctamente las patentes completas o, en su defecto, las letras y números que las componen en escenarios más complicados.
- Funciona de manera robusta bajo diversas configuraciones, con un buen equilibrio entre precisión y eficiencia.

Sin embargo, para alcanzar un desempeño más consistente y superar los casos problemáticos, se recomienda continuar optimizando el pipeline. Esto incluye mejorar el ajuste dinámico de parámetros y explorar herramientas más avanzadas de procesamiento de imágenes o aprendizaje automático.