



Trabajo Práctico N.º 2

Materia: IA4.4 Procesamiento De Imágenes

Carrera: Tecnicatura Universitaria en Inteligencia Artificial

Primer integrante: Nicolás Noir: N-1273/4

Segundo integrante: Ignacio Eloy González: G-5933/1

Tercer integrante: Gonzalo Calvo: C-7345/8

Profesores: Gonzalo Sad, Juan Manuel Calle, Julián Alvarez

Fecha: 25/11/2024

ÍNDICE

PORTADA.....	1
ÍNDICE.....	2
PROBLEMA 1:.....	3
INTRODUCCIÓN.....	3
Punto A:.....	3
Punto B:.....	3
Punto C:.....	4
METODOLOGÍA RESOLUCIÓN.....	5
RESULTADOS.....	16
PROBLEMA 2:.....	17
INTRODUCCIÓN.....	17
Punto A:.....	17
Punto B:.....	17
METODOLOGÍA RESOLUCIÓN.....	18
RESULTADOS.....	27
CONCLUSIÓN.....	29

PROBLEMA 1:

INTRODUCCIÓN

En este trabajo práctico, se tiene como objetivo identificar y diferenciar distintos tipos de monedas y datos

Punto A:

Para este punto, nuestro procedimiento fue aplicar un filtro promedio sobre la imagen original, lo siguiente fue utilizar el canal S del mapa de colores HLS, es decir, la saturación. Una vez obtenido este canal, le aplicamos otro filtro promedio, pero esta vez con un kernel un poco más grande, dejando la imagen un poco más suavizada. Luego se umbralizo con un umbral seleccionado manualmente, el cual nos facilitó diferenciar las monedas y datos del fondo, pero con algunos de los mismos teniendo partes faltantes, esto lo arreglamos aplicando una operación morfológica de clausura con un kernel de elipse y después una apertura con un kernel de elipse más grande. A continuación, al tener la imagen preparada para poder segmentar las monedas y datos, decidimos utilizar contornos y mediante el factor de forma pudimos diferenciar entre las formas circulares de las monedas y las formas rectangulares/cuadradas de los datos.

Punto B:

Para este punto, utilizamos dos enfoques diferentes, uno más rápido para el caso de tener 2 o 3 tipos de monedas en la imagen y uno más robusto para el caso de tener un solo tipo de moneda, el cual tiene un cómputo mayor. En el primer método ordenamos las áreas de las monedas y al dividir el área más pequeña sobre el área más grande podemos saber cuál es la moneda más pequeña de área y la más grande, entonces iterando sobre todos los contornos y dividiendo el área sobre la más grande, contamos las diferentes monedas. El segundo método es un poco más complejo porque en el caso de no tener 2 o 3 tipos de monedas, la división del área más pequeña sobre la más grande da 1 y es necesario trabajar sobre el recorte rectangular del contorno de la moneda. Este recorte lo hacemos sobre una imagen sin procesar, la cual es obtenida en el mapa de colores CIELAB para utilizar el canal B, el cual es la posición entre amarillo y azul, esto nos permite umbralizar la imagen en un valor seleccionado manualmente, siendo donde comienzan los valores amarillos, y mediante una sucesión de clausura y apertura encontramos las partes amarillas de las monedas. Luego es utilizamos contornos y en caso de no encontrar ningún contorno, podemos afirmar que es una moneda de 10 centavos, la cual no tiene color amarillo, en caso de encontrar un contorno comparamos el área del recorte rectangular de la moneda con el contorno de la parte amarilla de la moneda, al ser la parte amarilla de la moneda de un peso la parte interna de la misma y tener un borde exterior plateado, la proporción de área entre el área del rectángulo y área de la misma es diferente a la de 50 centavos que es toda amarilla.

Punto C:

Para este punto, diferenciamos las formas que no eran circulares con el factor de forma anteriormente, utilizamos su contorno en una imagen de ceros nueva, rellenándolos, luego se realiza una clausura grande y una apertura pequeña para poder obtener una forma más acorde a la de un dado. A continuación se buscan los contornos nuevamente para poder recortar cada dado por separado y procesarlo.

Una vez obtenido el dado, se lo recorta y se le da un margen de error, para así aplicarle un filtro de la mediana y una umbralización, luego se le aplica un umbral, ya que sabemos que el dado es blanco y los puntos dentro de él son negros, después se realiza una operación lógica para cambiar los colores negros a blancos y blancos a negros y así poder utilizar contornos para poder detectar los diferentes puntos de cada dado. Los cuales detectamos mediante el factor de forma, ya que son circulares. Obviando la posibilidad de detectar los puntos de las caras laterales de los dados, ya que tienen una forma ovalada.

METODOLOGÍA RESOLUCIÓN

Para este problema primero se generó un desenfoque para así poder diferenciar mejor los objetos luego se pasaría la imagen introducida en RGB a HLS, para así poder realizarle un desenfoque en la componente S (saturación), ya que esto ayuda a suavizar aún más las variaciones de saturación en los objetos facilitando así, la binarización y detección de objetos.



Figura 1.0

Primer desenfoque

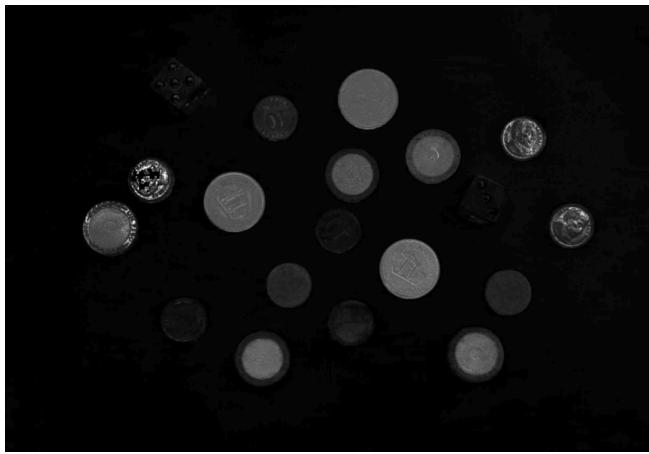


Figura 1.1
Canal S

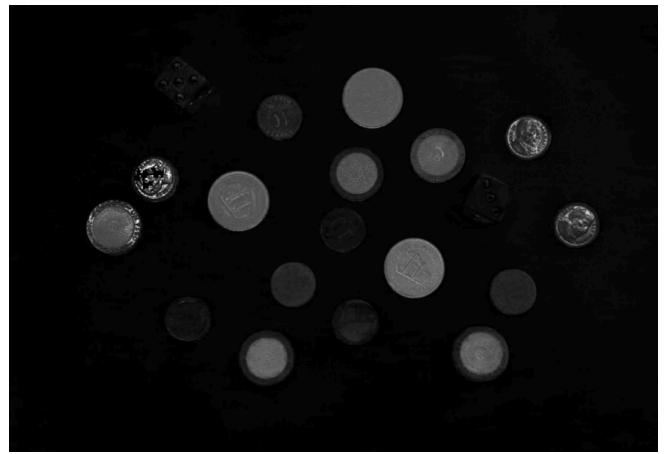


Figura 1.2
Desenfoque sobre el canal S

Trabajo Práctico N.^o 2

Luego, se le realiza un umbralizado para binarizar la imagen y así poder diferenciar los objetos del fondo obteniendo de esta manera sus diferentes características .

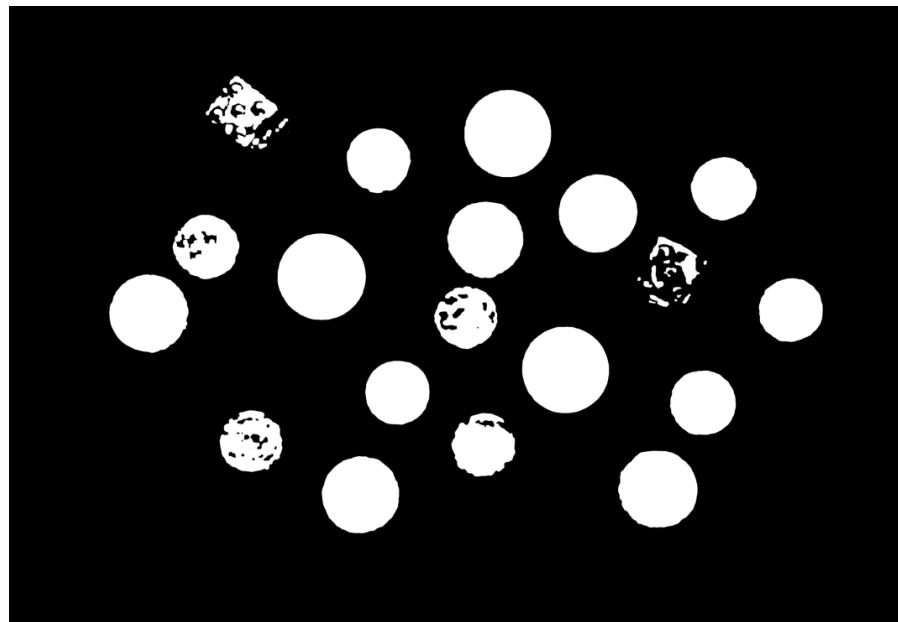


Figura 1.3.0

Se obtienen las formas de los objetos, pero con cierto ruido o inconsistencia en sus formas, por lo que se plantea implementar una clausura y apertura para cerrar y suavizar los contornos de los objetos

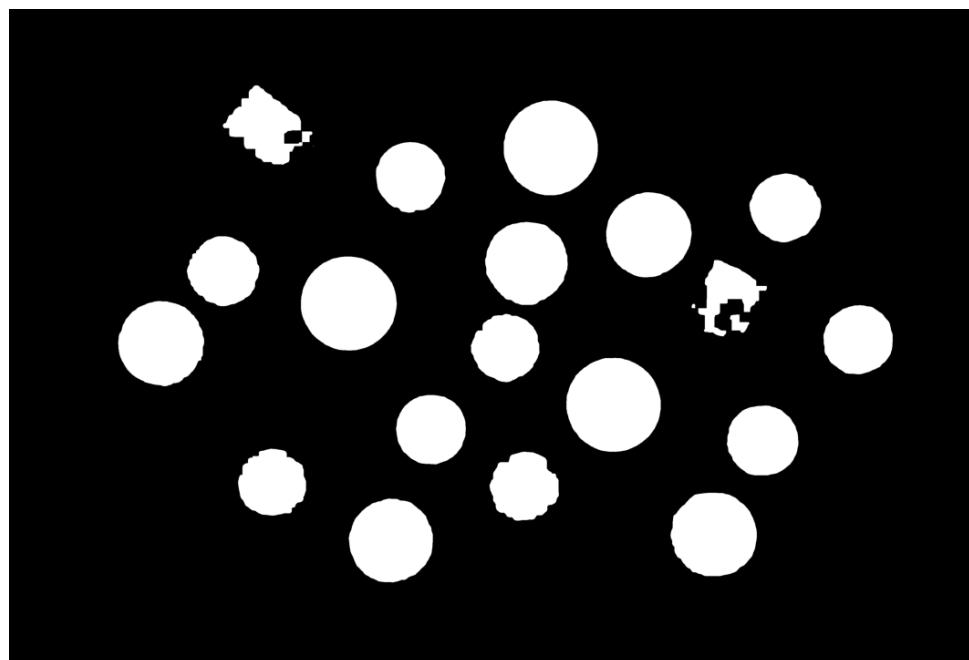


Figura 1.3.1

Trabajo Práctico N.^o 2

Se puede visualizar que la mayoría de las monedas ya tienen una forma circular óptima, pero otras aún tienen picos, además de tener ciertas formas o ruido distribuido en la imagen, como los siguientes ejemplos:



Figura 1.3.1.1

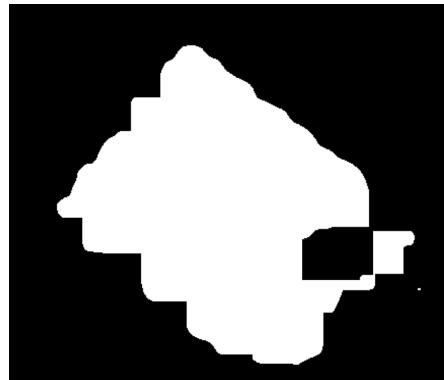


Figura 1.3.1.2

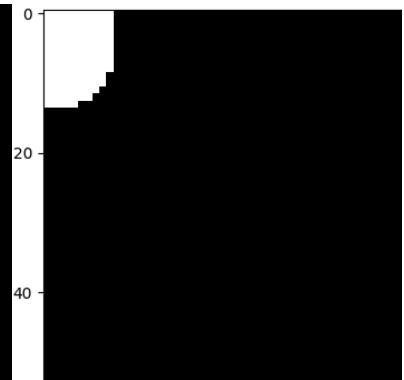


Figura 1.3.1.3

por consiguiente, además de los bordes con ciertos picos en las monedas, se tienen formas sobrantes que luego van a molestar en la identificación de los círculos, por lo tanto, se procede en aplicarle una apertura, para remover las regiones sobrantes de los elementos y suavizar los contornos

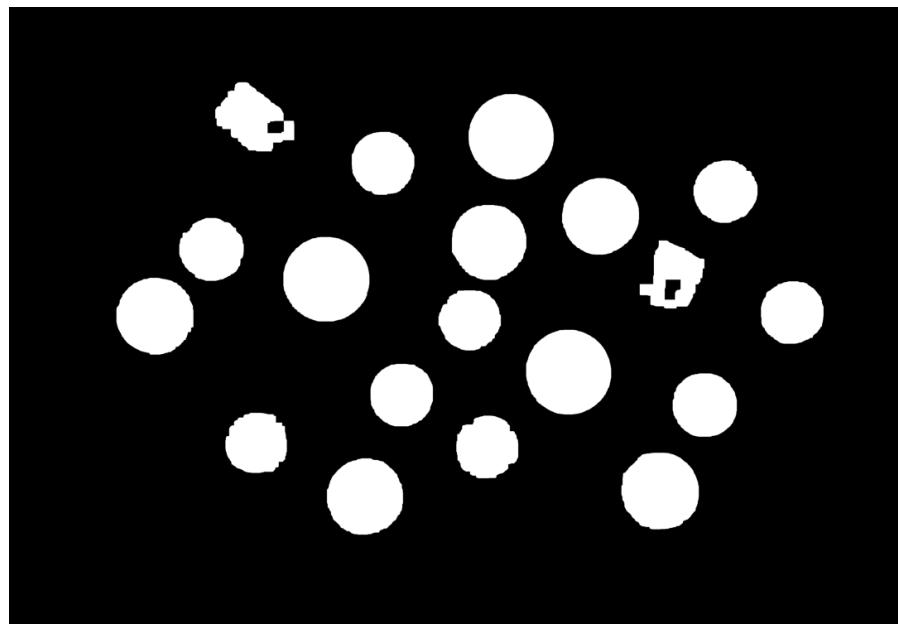


Figura 1.3.2

Trabajo Práctico N.^o 2

Posteriormente, se decide en buscar los contornos, para así poder identificar las formas que sean círculos (las monedas), de las que no (los dados), iterando sobre cada contorno de cada objeto, calculando su área y su perímetro, para buscar el factor de forma para identificar los círculos. Luego el resultado de identificar y dibujar los contornos de los círculos es:



Figura 1.4

Al tener ya encontradas todas las monedas con sus respectivos contornos y áreas, asumiendo que el área y los contornos pueden variar dependiendo de la distancia que se encuentra la persona que toma la foto, la relación que no cambia con respecto a la distancia entre la persona y las monedas, es la relación entre el área de las diferentes monedas, formalizando esto como:

Sea A_x , B_x , y C_x áreas de monedas, p y k constantes escalares.

Relaciones:

1. Primera relación:

$$A_x \cdot p = B_x$$

Donde B_x es el producto de A_x y el escalar p .

2. Segunda relación:

$$B_x \cdot k = C_x$$

Donde C_x es el resultado de multiplicar B_x por el escalar k .

3. Combinando las ecuaciones:

Sustituyendo la primera relación en la segunda, obtenemos:

$$C_x = (A_x \cdot p) \cdot k = A_x \cdot p \cdot k$$

Por lo tanto, se cumple que:

$$A_x \cdot p \cdot k = C_x$$

Este planteamiento muestra la relación entre A_x , B_x , y C_x en función de los escalares p y k .

Gracias a estas relaciones, se miden las relaciones entre la diferencia del área de la moneda más grande (moneda de 50 centavos) con la moneda mediana (moneda de 1 peso), la diferencia entre la moneda mediana (moneda de 1 peso) con la moneda chica (moneda de 10 centavos) y por último la diferencia entre la moneda grande (moneda de 50 centavos) con la moneda chica (moneda de 10 centavos).

Estas relaciones tienen cierto umbral de error, asumiendo que el contorno que se puede encontrar de las monedas no es perfecto. Gracias a esto, podemos tomar la moneda más grandes y compararlas teniendo en cuenta estas relaciones, pudiendo clasificarlas de manera exitosa, en los casos donde hay 3 o 2 tipos distintos de monedas.

En el caso donde solo existe un tipo de moneda, se compara el área máxima con el área mínima de las monedas que se encuentren, determinando que si su relación de aspecto es cercana a 1, es decir, que son de la misma categoría, se busca diferenciarlas según sus colores, recortándolas primero por sus contornos.

Luego, como ya sabemos que las monedas que tenemos, son “todas iguales”, tomamos la que tenga un mayor tamaño, ya que la mayor suele ser la moneda con menos errores. Siguiendo con el problema que teníamos al principio del ejercicio, que tuvimos que utilizar otro espacio de color para poder diferenciar los colores de las monedas, por lo que pensamos en utilizar CIELab, ya que su canal b se mueve en los valores amarillos y azules, por lo que podíamos utilizar los valores amarillos o dorados en el caso de las monedas para diferenciarlas, ya que la moneda de 10 se puede diferenciar del resto, ya que no tiene colores dorados o amarillos, por lo que podríamos descartarla por esta característica, siguiendo con el mismo planteamiento, la moneda de un peso tiene el área exterior plateado y el centro dorado. Entonces, recortamos la moneda, para poder trabajar con CIELab, específicamente con el canal b.



Figura 1.5.1.0



Figura 1.5.2.0



Figura 1.5.3.0

Luego le aplicamos un umbralizado, para poder trabajar con los distintas áreas de las monedas y le realizamos una operación lógica para poder identificar los contornos de las monedas

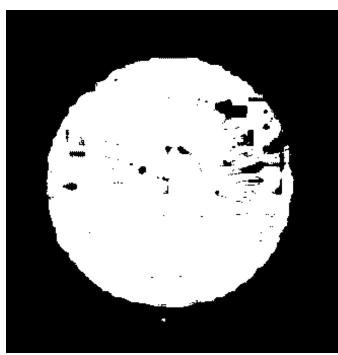


Figura 1.5.1.1

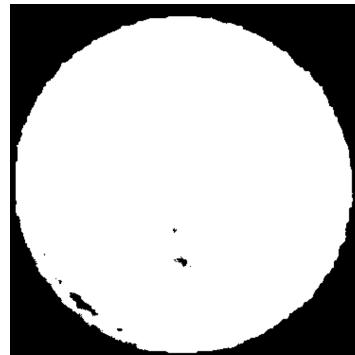


Figura 1.5.2.1



Figura 1.5.3.1

Al obtener el área de las monedas con cierto ruido o huecos en ellas, procedemos a realizarle una clausura seguida de una apertura, al igual que al principio del ejercicio, así podemos identificar mejor las monedas, sin ningún defecto.

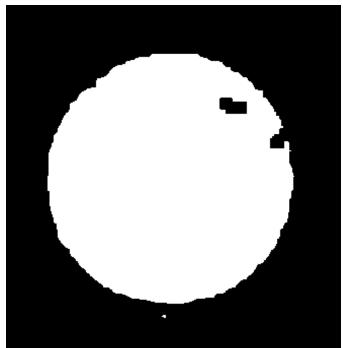


Figura 1.5.1.2

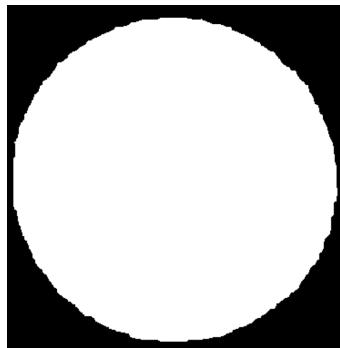


Figura 1.5.2.2

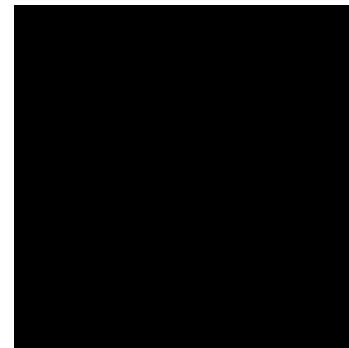


Figura 1.5.3.2

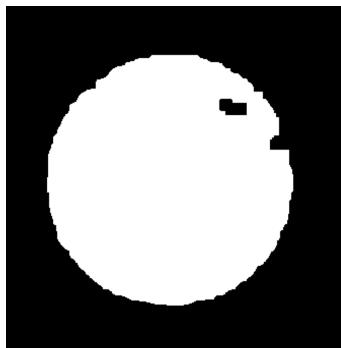


Figura 1.5.1.3

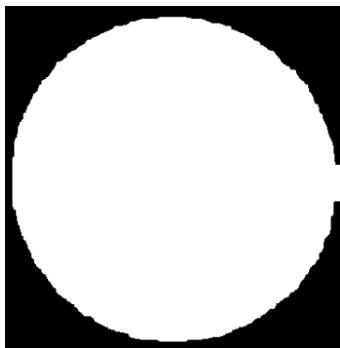


Figura 1.5.2.3

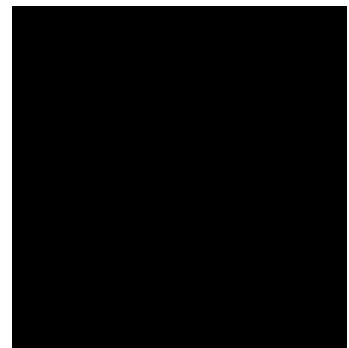


Figura 1.5.3.3

Al final decidimos realizar una clausura final con un kernel pequeño, con el único propósito de terminar de limpiar el área de las monedas, aunque esto realmente no modifica el resultado, pero para quitar cualquier tipo de inconveniente seguro, tomamos la decisión de aplicarlo igual para los casos excepcionales

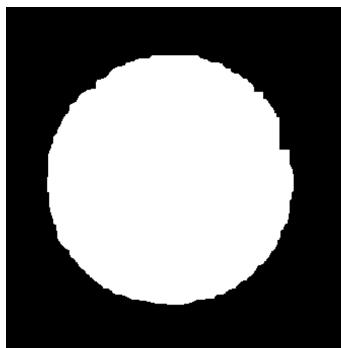


Figura 1.5.1.4

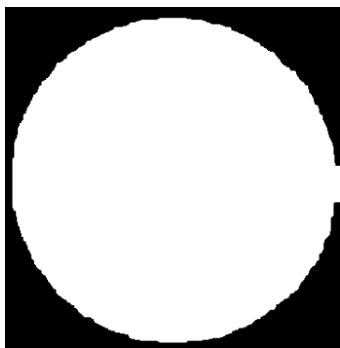


Figura 1.5.2.4

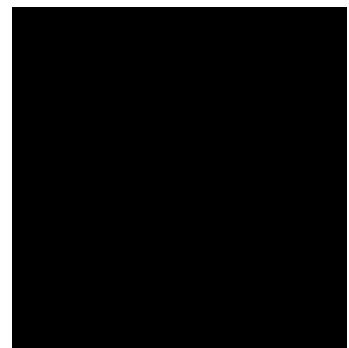


Figura 1.5.3.4

Al momento de diferenciar las monedas, se descarta la moneda de 10, ya que no contiene ningún área, al ser una moneda de color plateado. El problema es al momento de diferenciar la moneda de 1 peso y la moneda de 50 centavos, se planteó, de igual manera que para identificar las 3 monedas, buscar una relación entre el recuadro y la moneda, ya que en el caso de la moneda de 1 peso, esta siempre se verá reducida su área, generando así, que sea mayor el fondo que la moneda de 50 centavos, que su forma queda igual, esto teniendo en cuenta que las medidas de sus recuadros no son iguales, planteándose formalmente de la siguiente manera:

Relación entre Recuadros y Monedas

Supongamos que tenemos dos tipos de recuadros y dos tipos de monedas. Denotemos las áreas de los recuadros como $A_{\text{recuadro1}}$ y $A_{\text{recuadro2}}$, y las áreas de las monedas como A_{moneda1} y A_{moneda2} .

1. Relaciones promedio de área

Podemos definir una relación promedio entre el área de cada tipo de moneda y el área de su respectivo recuadro:

$$k_1 = \frac{A_{\text{moneda1}}}{A_{\text{recuadro1}}} \quad \text{y} \quad k_2 = \frac{A_{\text{moneda2}}}{A_{\text{recuadro2}}}$$

Donde k_1 y k_2 representan las proporciones promedio que el área de cada moneda ocupa respecto al área de su recuadro.

2. Expresión de las áreas de las monedas en función de los recuadros

Si conocemos los valores de $A_{\text{recuadro1}}$ y $A_{\text{recuadro2}}$, podemos expresar las áreas de las monedas como:

$$A_{\text{moneda1}} = k_1 \cdot A_{\text{recuadro1}} \quad \text{y} \quad A_{\text{moneda2}} = k_2 \cdot A_{\text{recuadro2}}$$

3. Comparación de proporciones

Para comparar las proporciones promedio entre ambos tipos de monedas y recuadros, podemos dividir una relación por la otra:

$$\frac{k_1}{k_2} = \frac{\frac{A_{\text{moneda1}}}{A_{\text{recuadro1}}}}{\frac{A_{\text{moneda2}}}{A_{\text{recuadro2}}}}$$

4. Proporción general

En general, si las áreas de los recuadros y las monedas pueden variar, pero queremos mantener una proporción promedio constante, podemos definir una proporción general k de manera que:

$$A_{\text{moneda}} \approx k \cdot A_{\text{recuadro}}$$

Donde k es la proporción promedio de área entre una moneda y su recuadro correspondiente.

Dando por finalizada la diferenciación, categorización y conteo de las diferentes monedas.

2.^a parte del ejercicio, encontrar los dados y contar el valor de su cara superior. Se parte desde la diferenciación en los contornos donde se encontraban monedas y dados.

Con las formas (dados) que encontrábamos, se dibujaban en una imagen blanca nueva, para su posterior procesamiento de forma más cómoda y sencilla

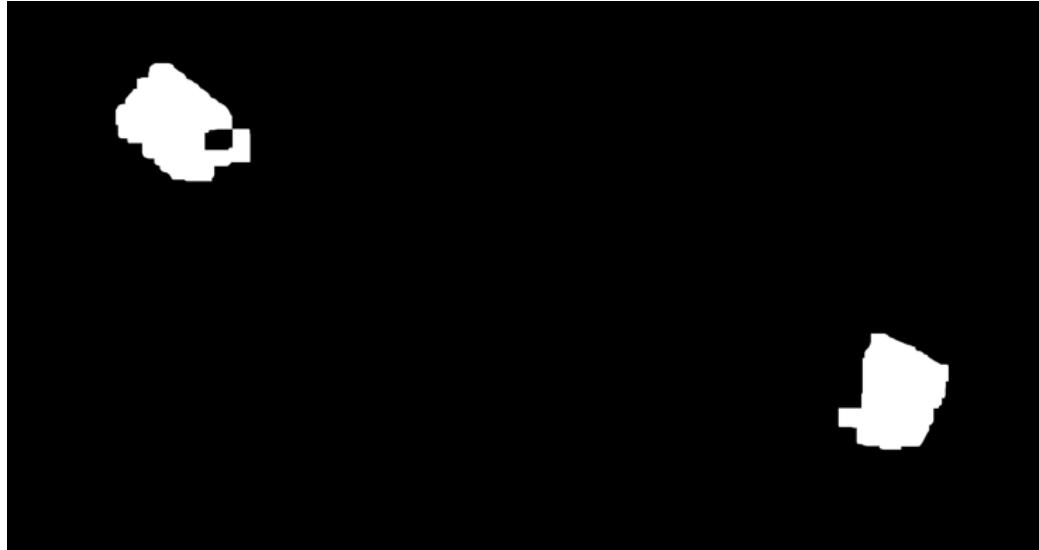


Figura 1.6.0

Trabajo Práctico N.^o 2

Posteriormente, se le aplica clausura y apertura, para, obtener forma geométricas más claras y similares a los dados de la imagen original

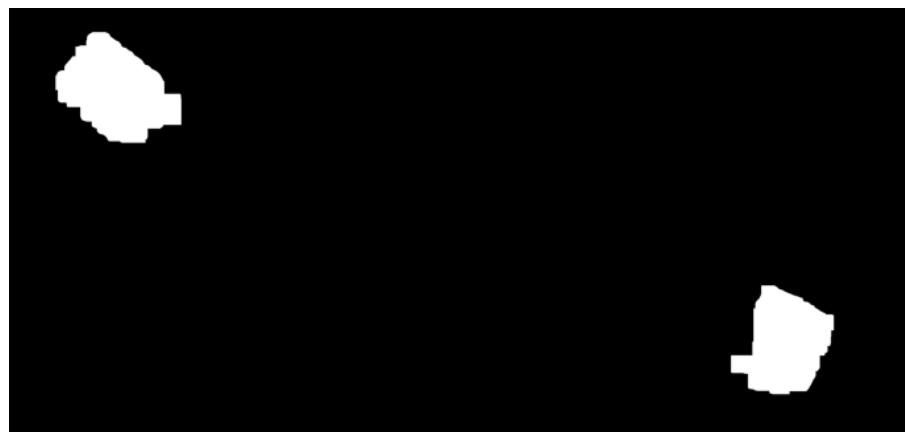


Figura 1.6.1

Luego, de una copia de la imagen original de las monedas, realizamos el recorte del área que teníamos anteriormente haciendo uso del bounding Rect.



Figura 1.7.1.0



Figura 1.7.2.0

Seguidamente, se pasa la imagen a gris, se le realiza un difuminado para quitar posible ruido.



Figura 1.7.1.1

Figura 1.7.2.1

Luego, se binariza para poder obtener los diferentes puntos del dado

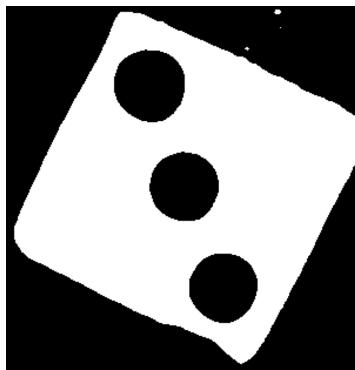


Figura 1.7.1.2



Figura 1.7.2.2

También se realiza un umbral, ya que sabemos que el dado es blanco y los puntos dentro de él, son negros, por consiguiente, se realiza una operación lógica, para cambiar los colores negros a blancos y blancos a negros, entonces, ahora se es capaz de utilizar la función para encontrar los contornos de los puntos del dado.

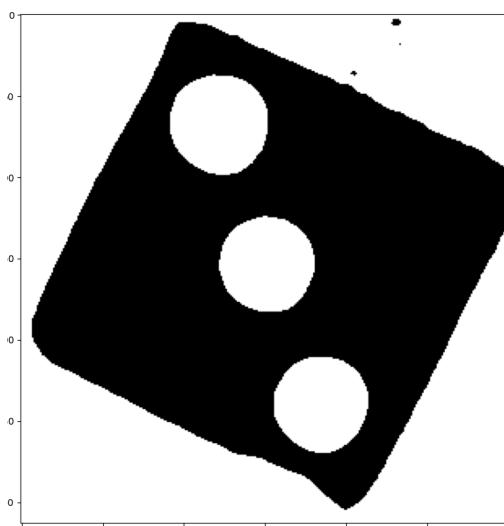


Figura 1.7.1.3

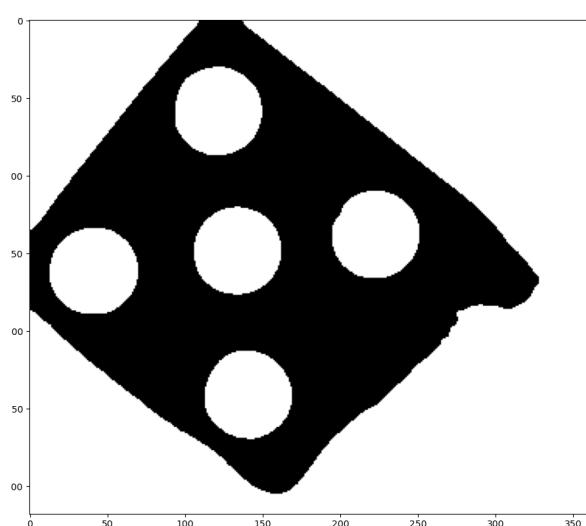


Figura 1.7.2.3

Se obtiene los contornos del fondo y de los puntos, por lo que podemos utilizar el mismo método anteriormente utilizado para diferenciar las monedas de los dados, ya que los puntos son círculos casi perfectos, y el fondo o son triángulos o son formas muy alejadas de los círculos, por lo tanto, es fácil poder diferenciarlos. Se cuenta la cantidad de círculos en

cada imagen, habiendo iterado en cada dado, es fácil asignar un dado a su cantidad de puntos, dado por finalizado el ejercicio número uno.

RESULTADOS

En este problema decidimos que será mostrada la imagen con los dados y monedas segmentados y en la terminal la cantidad de tipos de monedas y el número del valor de la cara superior de cada dado.



Figura 1.8.0

Se encontraron: "3" monedas de 50 centavos, "5" monedas de 1 peso y "9" monedas de 10 centavos
Y tambien se encontro "2" dados y sus respectivos valores son:

- En el dado número 1, tiene 3 puntos
- En el dado número 2, tiene 5 puntos

Figura 1.8.1

PROBLEMA 2:

INTRODUCCIÓN

En esta parte del trabajo se tenía como objetivo detectar y segmentar automáticamente la placa patente de las imágenes de autos designadas junto con las letras dentro de la misma.

Punto A:

Inicialmente, procesamos la imagen convirtiéndola a escala de grises para así luego aplicar blackhat (operación morfológica que consta de aplicar una clausura y restar la imagen en escala de grises a esta clausura) y poder resaltar el fondo negro de la placa patente. Luego restamos la imagen resultante de la operación blackhat a la imagen en escala de grises, ya que nos permite tener el fondo de la placa patente uniforme, precisamente en el valor 0, lo cual permite umbralizar la imagen y obtener el fondo de la patente como un componente conectado. El siguiente paso para detectar la placa patente, es comparar la relación de aspecto entre el alto y el ancho de los puntos máximos y mínimos con respecto al eje X e Y de todos los componentes conectados de la imagen binarizada, sabiendo que al ser un rectángulo de mayor ancho que alto y dividiéndolos podremos filtrar la gran mayoría de componentes que no tengan esa forma, además al ser imágenes de un tamaño 360 x 640, podemos filtrar estos componentes ya con forma “rectangular” según el área calculada con alto por ancho y no por la cantidad de píxeles contados en el componente conectado. De esta forma obtenemos una lista pequeña de posibles patentes.

Punto B:

Una vez encontrada la lista de posibles patentes, para saber que es una patente, necesito reconocer si dentro de este componente se encuentran las letras de la misma. Por lo que mediante otra imagen procesada con una serie de thresholds y operaciones de conjuntos que me permite detectar y segmentar de mejor manera las letras, recortamos esta posible patente en esta nueva imagen y filtramos los componentes conectados que se encuentren dentro del recorte, primero con la relación de aspecto, ya que una letra generalmente es más alta que ancha y después por el área mínima calculada con alto por ancho. Una vez iterados todos los componentes dentro de la posible patente, si el componente pasa ambos filtros es contado como una letra y mediante un contador, en caso de tener 6 letras dentro, se segmentan las mismas y se afirma que es una patente el componente seleccionado primeramente.

METODOLOGÍA RESOLUCIÓN

Primero vemos la imagen en el mapa de colores RGB



Figura 2.0

Observamos que excepto un auto, todos respetan el formato de patente de la Figura 2.0 entonces nuestra idea es tratar de segmentar primero el fondo de la patente que es mayormente uniforme y no suele tener tanto ruido a los costados. Procesaremos la imagen primero pasándola a escala de grises.



Figura 2.1

Ahora es necesario resaltar la parte negra del fondo de la placa patente, para eso usaremos Blackhat (operación morfológica parecida a Tophat, pero en este caso se realiza una clausura a la imagen con un kernel con las dimensiones parecidas al objeto que necesitamos resaltar y a esa clausura le restamos la imagen en escala de grises) de este modo la imagen quedará de la siguiente manera:



Figura 2.2

Vemos en la Figura 2.2 que la placa patente queda en un valor casi uniforme, si quisieramos umbralizar la Figura 2.2 con Otsu o un valor fijado manualmente tendríamos problemas con diferentes imágenes de autos

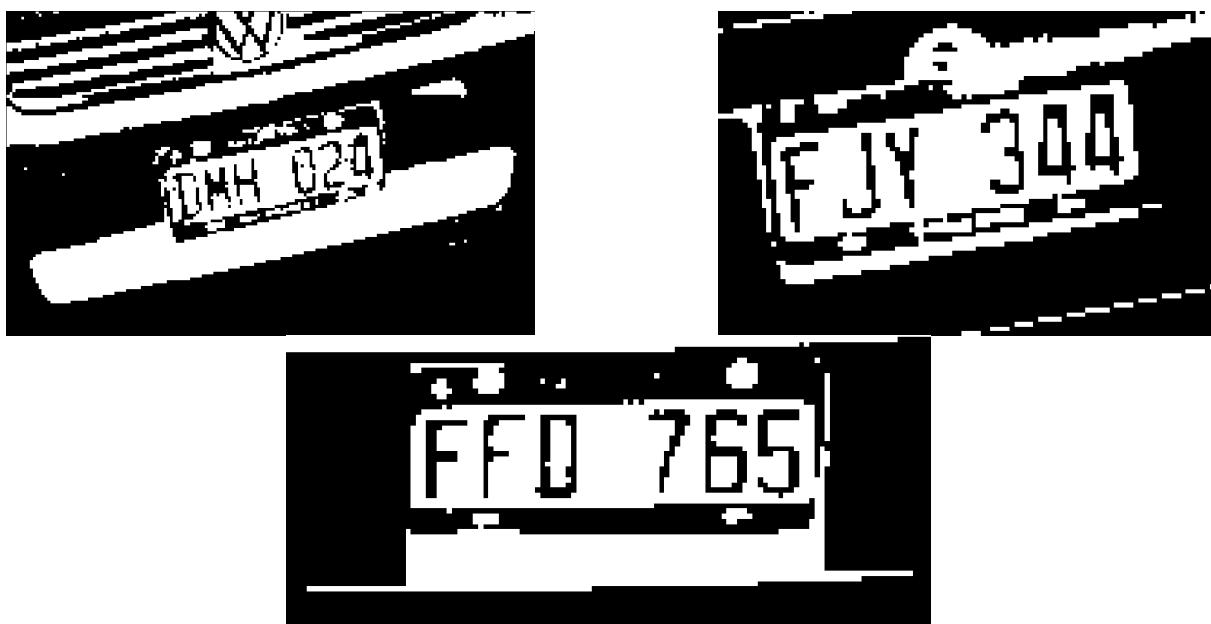


Figura 2.3

Como vemos en la Figura 2.3 la umbralización con Otsu nos traería problemas para detectar las placas patentes, por lo que pensamos que si nosotros le restamos la Figura 2.2 a la Figura 2.1, podríamos obtener el fondo de la patente en un valor uniforme y remarcado por el borde exterior de la placa patente, permitiendo que el umbralizado funcione mejor.



Figura 2.4

Observando la Figura 2.4 vemos que la zona de interés ahora tiene un valor uniforme, y procedemos a umbralizar la imagen para valores menores a 1





Figura 2.5

Observamos que en la Figura 2.5 la mayoría de las patentes están perfectamente recortadas, excepto la última patente, la cual lleva un poco de ruido, pero sigue manteniendo la relación de aspecto de la placa patente.

Una vez que se realizó este umbralizado lo siguiente es usar componentes conectados para encontrar componentes que tengan un ancho por alto con la misma relación de aspecto de las placas patentes argentinas de modelos anteriores a 2016, es decir, 360 mm de ancho por 130 mm de alto, lo cual da una relación de 2.76. Nosotros optamos por darle un margen alto de error, ya que en muchas imágenes se ven rotaciones o diferentes ángulos de captura, este margen va desde 1.8 a 4.2 en el cual probando con diferentes imágenes no resultó tener problemas.



Figura 2.6

Una vez pasado este filtro de relación de aspecto de la Figura 2.6 va a ser necesario filtrar por el área del componente, este filtro lo realizamos aunque no sea el más robusto, ya que al ser imágenes normalizadas en 360x640, y tener siempre una distancia entre la patente y la cámara parecida, pensamos que podía ser útil y no causar tantos problemas descartar componentes menores a 300 píxeles de área y mayores a 5000 píxeles de área. El área que se utiliza no es la proporcionada por los stats de componentes conectados, sino ancho por alto



Figura 2.7

Una vez pasado el filtro anterior, observamos que aun así había componentes que no eran patentes y tenían estas mismas características, por lo que se nos ocurrió que una vez encontrados componentes con forma y área parecida a la de una placa patente, para ser una placa patente tendría que tener 6 caracteres dentro de su perímetro. Entonces decidimos utilizar las coordenadas que proporciona stats para recortar la imagen en la placa patente y poder hacer componentes conectados dentro del recorte de la placa patente. Luego de varias pruebas, nos encontramos con que la imagen que utilizábamos para encontrar los fondos de las placas patentes, al invertirla y tener las letras en valor 255 para poder ser utilizada con componentes conectados, no funcionaba bien para encontrar las letras dentro, ya que muchas letras se terminaban conectando entre sí y siendo un solo componente.



Figura 2.8

Después de muchas pruebas de diferentes filtros para encontrar las letras dentro de las patentes, llegamos a la conclusión de que BlackHat capaz no era la mejor opción para encontrar las letras. Por lo que vimos necesario utilizar otro preprocesamiento para la segmentación de las letras dentro de la patente. Lo primero fue probar diferentes thresholds por sí solos como Otsu o Adaptive, en el que encontramos que funcionaban bien en algunas patentes, pero en otras no. Después intentamos juntar Otsu y Adaptive, ya que si funcionaban bien en diferentes patentes, capaz después de juntarlos funcionarían para la mayoría, lo cual no fue el caso. Lo siguiente fue probar Otsu con una máscara que no es más que un threshold seleccionado manualmente que tenía buen funcionamiento en varias patentes, esto resultó en algo que por sí solo no funcionaba para todas las patentes, pero reducía demasiado el ruido en los píxeles cercanos a la patente. Por otro lado, probamos el Adaptive con una máscara que también era un threshold seleccionado manualmente y como vimos en el caso anterior, también reducía el ruido en los píxeles cercanos a la patente y funcionaba para casos que Otsu no. Como último decidimos juntar estos métodos que reducían el ruido en una sola imagen y observamos que funcionaban para todas las imágenes.



Figura 2.9

Ahora que tenemos las imágenes preparadas (Figura 2.9, en específico método completo) para segmentar las letras, necesitaremos utilizar las coordenadas de las posibles componentes conectadas para recortar en estas nuevas imágenes para así iterar sobre los componentes dentro del recorte y encontrar las letras. Para determinar si un componente es una letra, tendremos que aplicar los mismos filtros que aplicamos para encontrar las posibles patentes, pero cambiando los valores de aceptación, ya que las letras tienen una forma rectangular que es más grande verticalmente y su área es mucho más chica.



Figura 2.10

En la Figura 2.10 se ve un ejemplo de como iría iterando el algoritmo sobre los componentes de la patente, en donde nosotros para saber si cumple con esta condición de los 6 caracteres, si cumplen los filtros antes explicados, se añaden a una lista los índices que devuelve componentes conectados, así es posible conseguir sus coordenadas fácilmente.



Figura 2.11

En la Figura 2.11 podemos observar como se verían los caracteres juntos una vez detectados.



Figura 2.12

Luego de iterar sobre los componentes conectados, verificamos si dicho componente tiene 6 subcomponentes conectados, de ser así ya pasado los filtros previos, decimos que es una patente y comenzamos a dibujar los Bounding Box de la patente y cada uno de los caracteres. En el caso de que no encontremos un componente conectado con 6 subcomponentes conectados, retornamos un string.

Al final observamos que todo este procedimiento funcionaba para 11 imágenes de 12, entonces decidimos que para detectar la última placa patente, era necesario el mismo procedimiento para encontrar las placas patentes, pero con diferentes parámetros de preprocessamiento de las imágenes. Para esto, pensamos que un enfoque útil podría ser ecualizar la imagen en escala de grises antes de aplicar la operación morfológica de BlackHat



Figura 2.13

Observamos en la Figura 2.13 que la placa patente para este caso que no podía ser resuelto con el otro preprocessamiento podría funcionar, obviamente a continuación aplicamos BlackHat.



Figura 2.14

En la Figura 2.14 observamos que este enfoque ayudaba para la segmentación de esta placa patente. Por lo que umbralizamos con valores seleccionados manualmente, para no perder las letras dentro de la placa patente en caso de utilizar algún umbralizado automático como Otsu o Adaptive.



Figura 2.15

Observamos que en la Figura 2.15 la placa patente y las letras dentro de ella son diferenciables, por lo que utilizamos esta imagen umbralizada para la detección de la placa patente con el procedimiento descrito anteriormente y para el recorte donde se segmentarían los caracteres de la patente utilizaríamos la imagen inversa



Figura 2.16

En la Figura 2.16 observamos que todos los caracteres dentro de la placa patentes son componentes 8 conectados y en efecto funcionaría todo el procedimiento para esta imagen.

RESULTADOS

En este problema decidimos que serán devueltas las imágenes en RGB una por una con el Bounding Box de la placa patente y los de todos los caracteres dentro de ella, viéndose así.



Figura 2.17



Figura 2.18



Figura 2.19



Figura 2.20

Mostramos algunas de ellas, siendo las que pensamos que tienen más posibilidades de no ser detectadas por sus diferentes características, así como ruido en la patente, estructuras cerca de ellas, ángulo en la patente y formato de patente diferente a las otras.

CONCLUSIÓN

Para dar como finalizado este trabajo práctico nos gustaría indagar en los diferentes planteamientos y problemas que tuvimos a lo largo del desarrollo del mismo. Empezando con el primer problema, al momento de realizarle los diferentes procesos a la imagen para poder identificar los distintos objetos, tuvimos problemas al tratar de encaminar este procesamiento, ya que empezamos probando con realizarle umbralizados y filtros sobre la imagen en RGB, a lo que se nos imposibilitaba la diferenciación de los objetos del lado izquierdo de la imagen que de los de la parte derecha de la imagen por culpa de la dispar iluminación que tiene la foto. Luego probamos utilizando el canal HLS con el que se podría únicamente trabajar con la parte S de saturación, aprovechando esto para disminuir el ruido por la iluminación, gracias a esto, obtuvimos las distintas formas, pero eso sí, con formas amorfas, por lo que procedimos en utilizar clausura y apertura como se nos había explicado en clase para este tipo de problemas. Al obtener las monedas con forma redonda y los dados con una forma parecida a un cuadrado o rectángulo, pensamos en separar las monedas de los dados para que así, sea más sencillo el análisis de cada una, utilizando el factor de forma para diferenciar círculos de otras formas, guardado las monedas, por un lado, y, al resto de formas que no eran las monedas, las pasábamos a una nueva imagen de ceros, con sus contornos llenos. Estos dados, todavía contenían cierto ruido, ya que en la parte anterior, se profundizó en el procesamiento centrado en darle forma a las monedas y no tanto a los dados, ya que los valores de la apertura y clausura que utilizábamos, eran ideales para las monedas, pero insuficientes para los dados; sin embargo, si tratábamos de buscar los mejores valores para los dados, las monedas perdían su forma y se imposibilitaba la posibilidad de separarlas e identificarlas por su forma y área.

Luego, con nuestra nueva imagen, con los dados en ella, utilizamos clausura y apertura, ya que contenían cierto ruido y picos, posteriormente, decidimos con las coordenadas del área de los dados que teníamos, recortar de la imagen original utilizando las coordenadas de los dados, luego con la imagen de los dados a color, se le aplicó un umbralizado y se les invirtió los colores para poder identificar los puntos de las caras superiores, utilizando el mismo método que con las monedas de buscar el área, perímetro y factor de forma, excluyendo el fondo que puede quedar del mismo color que los puntos del dado y los puntos de las caras exteriores que no coinciden con la forma de un círculo casi perfecto como el de la cara superior del dado.

En el caso de las monedas, se plantearon varias maneras de afrontar la diferenciación de las mismas, al principio se planteó buscar diferentes por las formas dentro de las mismas, pero este método no era robusto, ya que si la iluminación diferia de gran manera, estos detalles podrían no ser diferenciados ocasionando errores, por lo que planteamos poder diferenciarlas por la diferencia entre las áreas de las mismas, ya que este método tiene en cuenta que la posición de donde se saca la fotografía, podría estar más cerca o más lejos, pero esta relación de aspectos entre las áreas en las distintas monedas, iba a seguir siendo la misma, pro lo que sacamos un promedio de áreas que encontrábamos para cada moneda, viendo el caso que cuando mayor sea el área de la moneda, menos difería del resto de las monedas de la misma categoría, al contrario, con las más pequeñas, había un margen de error mayor. Teniendo esto en cuenta, calculamos las diferentes relaciones entre monedas, pudiendo calificarlas así, tomando la mayor moneda disponible como

comparación para el resto. Quisimos ir un paso más allá y pensar que pasaría si únicamente tuviésemos un tipo de moneda, por lo que pensamos que sí, la moneda con menor área y la mayor tenían una relación de aspecto menor al ochenta porcientos, serían del mismo tipo de moneda. Para este caso pensamos en utilizar la moneda con mayor área, ya que sería la que mejor definida estuviese, siendo del tipo de moneda que sea. Buscando, encontramos que podríamos diferenciarlas por los colores dorados de las monedas, ya que como sabemos las de diez centavos no tienen colores amarillos o dorados, y podríamos utilizar esta característica para diferenciarla del resto, en el caso de la moneda de cincuenta centavos y la de un peso, pensamos en utilizar un método similar al que usamos para diferenciar a los tipos de monedas, enfocándonos en el área que cubierta cada moneda con el fondo, ya que al tener la moneda de un peso el borde dorado, sé lo retira y así, quedarías mucho más borde que del caso de la de cincuenta centavos, aunque ambas monedas tuviesen dimensiones diferentes, este método era robusto, ya que no nos fijábamos en eso. Luego de identificarlas, guardamos todas las monedas como del mismo valor, ya que no haría falta probar para todas las monedas, porque, como anteriormente mencionamos, ya sabíamos que todas las monedas eran del mismo valor, si no hubiesen pasado a esta etapa

Por otro lado, en el problema de las patentes nuestro conflicto inicial fue el método de segmentación de patentes, ya que pensamos que sí encontrábamos los 6 caracteres de la placa patente sería suficiente, pero al probarlo con otras imágenes nuevas o en los posibles caracteres que puede tener un auto en la parte trasera como su modelo o la marca, observamos que podía llegar a tener problemas con esto, entonces decidimos cambiar el enfoque a encontrar el fondo interior color negro de la placa patente y obviamente al encontrar estas posibles placas patentes no era suficiente, ya que muchos objetos y partes remarcadas luego de la umbralización tienen una forma parecida a esto, al menos con nuestros filtros, por lo que usamos como criterio que una vez encontrada esta posible placa patente, sería necesario que dentro tenga los 6 caracteres con la forma característica de rectángulo vertical. Luego otro conflicto fue todos los posibles pre procesamiento que se le puede realizar a las imágenes, y tener que llegar a uno que sea funcional a la mayoría de las mismas, aquí capaz fue donde más tiempo estuvimos probando, así como probar muchos umbralizados, ecualizar, usar diferentes métodos de detección de bordes como Canny, ciclos de clausuras y aperturas, pero esto no daba resultados sobre todas las imágenes, y ahí es donde encontramos el preprocesamiento donde juntamos los diferentes umbralizados, que para encontrar los caracteres dentro de las placas patentes era el mejor encontrado hasta el momento, pero para encontrar las placas patentes como un componente conectado por separado de cualquier parte del auto no era útil, pero bueno en este momento ya habíamos conseguido 10 de 12 autos con patentes segmentadas y aunque queríamos conseguir segmentar todas las patentes, no nos quedaba mucho tiempo. Pero ahí fue donde encontramos BlackHat y la parte de encontrar el fondo interno color negro de la patente empezó a funcionar mejor, hasta que después de pruebas y cambios en el preprocesamiento llegamos a 10 patentes y desde nuestro punto de vista de manera más robusta, después de cambiar un poco de parámetros llegamos a la 11^a patente y nos encontramos con la patente doblada que no podíamos segmentar la placa completa y, por lo tanto, tampoco encontrábamos los 6 caracteres, sino 5. Aquí, después de preguntarle a los profesores, decidimos utilizar un preprocesamiento con parámetros diferentes, así como

Trabajo Práctico N.º 2

cambiar kerneles, ecualizar antes de todo y un umbralizado con un valor seleccionado manualmente, donde encontramos la última patente que faltaba. En general, el problema 2 nos pareció el más divertido, ya que había muchas maneras de abordarlo y terminas entendiendo mucho como funcionan cada uno de los métodos para el preprocesamiento de las imágenes.