



Trabajo Práctico N° 1

Procesamiento de Imágenes I

Tecnicatura Universitaria en Inteligencia Artificial

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

Universidad Nacional de Rosario

2023

Integrantes:

Augusto Farias,

Guido Lorenzetti,

Micaela Pozzo,

Patricio Vercesi

Problema 1

En este problema, debemos detectar las figuras camufladas en la imagen. Esto lo llevaremos a cabo realizando una verificación en las intensidades de píxeles.

Decidimos utilizar la función ecualización local del histograma y no una global ya que en utilizando todos los píxeles de la imagen no daría buenos resultados ya que pierde localidad de análisis. Esta función divide la imagen en ventanas pequeñas y ecualiza el histograma de cada ventana por separado. Luego, recompone la imagen a partir de las ventanas ecualizadas para obtener una imagen con mejoras locales de contraste.

Procedimiento:

- Se carga la imagen **“imagen_con_detalle_escondidos.tif”** en escala de grises usando la librería OpenCv.
- Se define un tamaño de ventana y se agrega un borde replicado a la imagen para que la función de actualización local pueda operar sin problemas en los bordes de la imagen.
- Se llama a la función **‘local_histogram_equalization’** para aplicar la ecualización del histograma local a la imagen.
- Luego, se aplica una binarización a la imagen resultante para convertirla en una imagen en blanco y negro. Los píxeles con intensidad mayor que 60 se establecen en blanco(255), mientras que los píxeles con intensidad menor o igual a 60 se establecen en negro(0).
- Se aplica un filtro de mediana para eliminar el ruido de la imagen binarizada.
- La imagen procesada (después de la binarización y filtrado de mediana) se guarda en el archivo **‘salida_ejercicio_1.png’**.

Resultado:

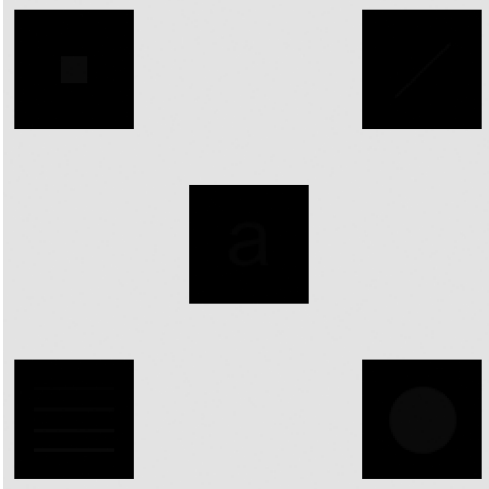
- En el cuadrado superior izquierdo se encuentra escondido un cuadrado de menor tamaño que éste.
- En el cuadrado superior derecho se encuentra una línea recta en dirección diagonal.
- En el cuadrado del centro de la imagen se encuentra escondida una letra a minúscula.
- En el cuadrado inferior izquierdo hay cuatro líneas rectas en dirección horizontal.
- En el cuadrado inferior derecho se esconde un círculo.

Conclusión:

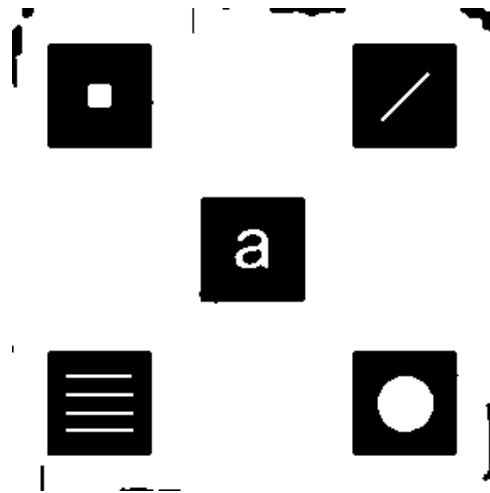
Luego de experimentar en diferentes ocasiones con tamaños variados de ventana, podemos concluir que esto influye directamente en la visualización de la imagen. En caso que se achique en gran medida, se observa mayor ruido en la imagen y si se expande en gran medida no se diferencian de buena manera las figuras ocultas. Por estas razones, decidimos setear el tamaño de la ventana en 30, ya que disminuye el área del ruido y nos permite identificar que es sólo ruido y podemos ver de buena manera las figuras ocultas.

La elección del tamaño de la ventana es importante para no confundir el ruido de la imagen con la figura oculta, que es nuestro objetivo en este caso.

Input:



Output:



Problema 2

En el segundo problema el objetivo es chequear si los campos de una serie de formularios con datos personales en formato de .png se encuentran completos de manera correcta. Procedemos a analizar el formulario verificando la validez del mismo.

Procedimiento:

- Como primer paso decidimos binarizar la imagen de manera que los bordes queden de 1 pixel de ancho
- Luego sumamos los pixeles negros por columna y por fila para tener un registro donde los valores altos indiquen las posiciones de los bordes verticales y horizontales del formulario.
- Como sabemos la cantidad de bordes verticales y horizontales presentes, nos quedamos con la cantidad especificada de los mayores valores de la sumatoria (ordenada de mayor a menor).
- Con la posición de inicio y fin de cada borde, y conociendo las celdas de interés, extraímos las sub-imágenes de cada campo. Para evitar tomar los bordes, extremos desde 2 pixeles internos.
- Usando la función `contar_caracteres_y_palabras` separamos los caracteres y las palabras en función de los espacios habidos entre ellos.
- Se devuelve el recuento de caracteres y palabras encontrados en la imagen
- Chequeamos si el número de caracteres y palabras es correcto para la celda que se está estudiando.
- Repetimos lo mismo para cada celda necesaria y para cada uno de los formularios existentes.

Resultado:

El resultado debe ser el siguiente:

Se coloca OK si los datos ingresados son válidos o MAL si hay errores en el ingreso de los mismos

Nombre y apellido	OK / MAL
Edad	OK / MAL
Mail	OK / MAL
Legajo	OK / MAL
Pregunta 1	OK / MAL
Pregunta 2	OK / MAL
Pregunta 3	OK / MAL
Comentarios	OK / MAL

Input:

FORMULARIO A		
Nombre y apellido	JUAN PEREZ	
Edad	45	
Mail	JUAN_PEREZ@GMAIL.COM	
Legajo	P-3205/1	
	Si	No
Pregunta 1	X	
Pregunta 2		X
Pregunta 3	X	
Comentarios	ESTE ES MI COMENTARIO.	

Output:

```
Nombre y Apellido:  OK
Edad:              OK
Mail:              OK
Legajo:            OK
Pregunta 1:        OK
Pregunta 2:        OK
Pregunta 3:        OK
Comentarios:       OK
```

FORMULARIO A		
Nombre y apellido	JORGE	
Edad	4500	
Mail	JORGE @GMAIL.COM	
Legajo	X45ASLAB W45	
	Si	No
Pregunta 1		
Pregunta 2	X	x
Pregunta 3		xx
Comentarios	ESTE ES UN COMENTARIO MUY MUY LARGO.	

```
Nombre y Apellido:  MAL
Edad:              MAL
Mail:              MAL
Legajo:            MAL
Pregunta 1:        MAL
Pregunta 2:        MAL
Pregunta 3:        MAL
Comentarios:       MAL
```