



FACULTAD DE
CIENCIAS EXACTAS,
INGENIERIA Y AGRIMENSURA

Universidad Nacional de Rosario

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

Trabajo práctico N° 01

Tecnicatura Universitaria en Inteligencia Artificial

Procesamiento de imágenes (IA4.4)

Integrantes	Legajo	Mail
Zahradnicek, Ezequiel	Z-1214/9	ezeonico@gmail.com
Martinez Dufour, Caterina	M-7169/2	caterinaxmd7@gmail.com
Grimaldi, Damián Daniel	G-5977/3	grimaldidamian001@gmail.com

Fecha: 21/10/2024

AÑO 2024



Índice:

1. Introducción.
2. Instrucciones generales.
3. Problema 1 - Ecualización local de histograma.
4. Problema 2 - Corrección de múltiple choice.
5. Conclusión.

Introducción:

En este informe se encontrará la resolución del primer trabajo práctico de la asignatura “Procesamiento de imágenes” de la Tecnicatura en Inteligencia Artificial. Se tiene como propósito aplicar los conceptos fundamentales de procesamiento de imágenes para resolver dos ejercicios prácticos donde se trabajará con imágenes, utilizando herramientas y técnicas dadas en clase.

Instrucciones generales:

A continuación se detallan las instrucciones para acceder al repositorio de GitHub y configurar el entorno de trabajo.

Acceso al Repositorio: El repositorio se encuentra en el enlace a continuación:
https://github.com/CaterinaMartinezD/TUIA_PDI_TP1.git

Clonar el Repositorio: Para clonar el repositorio en tu equipo local, ejecuta el siguiente comando en tu terminal o consola:

```
git clone --depth 1 https://github.com/CaterinaMartinezD/TUIA\_PDI\_TP1.git
```

Crear un Entorno Virtual (venv): Es obligatorio utilizar un entorno virtual para evitar conflictos con otras instalaciones de Python en tu sistema. En caso de utilizar un sistema operativo como linux, no será necesario este paso.

```
python -m venv venv (crea un entorno virtual)
```

```
venv\Scripts\activate (activa el entorno virtual)
```

Módulos necesarios: Para poder ejecutar correctamente todo el código será necesario instalar 3 módulos. Una vez que el entorno virtual esté activado, se deberá ejecutar los siguientes comandos en tu terminal:

```
pip install numpy
```

```
pip install matplotlib
```

```
pip install opencv-contrib-Python
```

Problema 1 - Ecualización local de histograma:

Este problema nos pide informar cuales son los detalles escondidos en las diferentes zonas de la imagen llamada "Imagen_con_detalle_escondidos.tif". Para resolver este ejercicio se utiliza la técnica de ecualización local del histograma, que mejora el contraste de una imagen de manera específica al analizar pequeñas áreas o "ventanas" de tamaño $M \times N$ que se desplazan píxel a píxel por toda la imagen.

En cada posición de la ventana, se genera el histograma de los píxeles que contiene, el cual se emplea para ajustar el valor de intensidad del píxel central, generando así una transformación local. Finalmente, se repite el proceso hasta cubrir toda la imagen, lo que permite resaltar detalles que podrían quedar ocultos con una ecualización global. Además, debemos analizar la influencia del tamaño de la ventana en los resultados obtenidos.

Resolución:

Para resolver este enunciado planteamos una función que calcula el histograma local en base a la imagen y tamaño de ventana recibidos como parámetros. Además agregamos como adicional un parámetro llamado truncate que ajusta los valores de intensidad de los píxeles, cambiando las intensidades más altas y más bajas según un umbral aplicado. Esto permite el ruido que en este caso es el del "salt and pepper".

A continuación paso a explicar paso a paso que realiza el código:

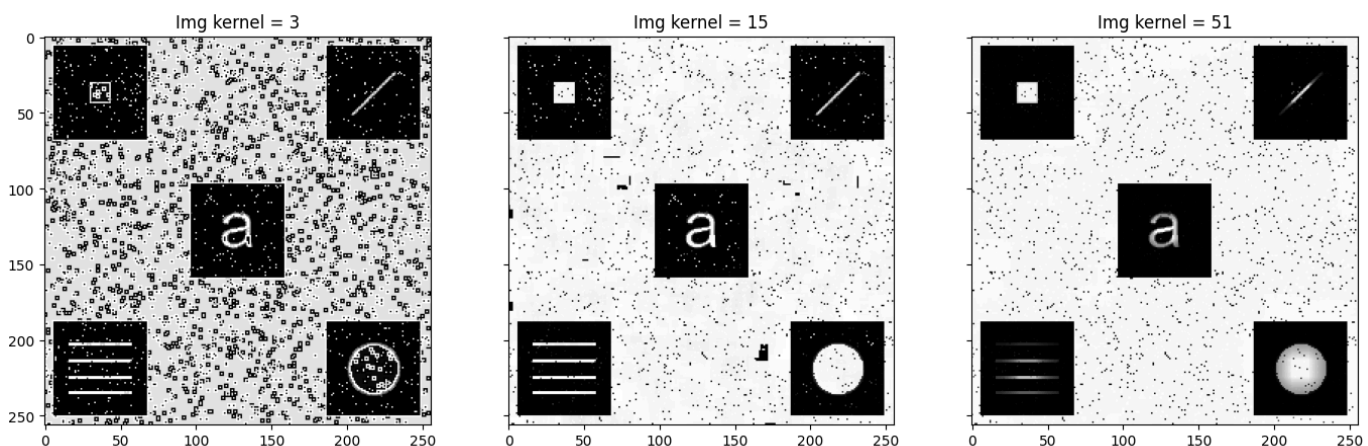
Se importan los módulos comentados anteriormente, se almacena en el parámetro "img" la lectura de "Imagen_con_detalle_escondidos.tif" y se utiliza una función llamada "equalizacion_local".

Esta función toma el tamaño de la ventana y genera bordes utilizando cv2.copyMakeBorder. Se agregan estos bordes a la imagen porque, al trabajar con una ventana deslizante de tamaño $m \times n$, el píxel central de cada ventana se actualiza según la información de los píxeles que lo rodean.

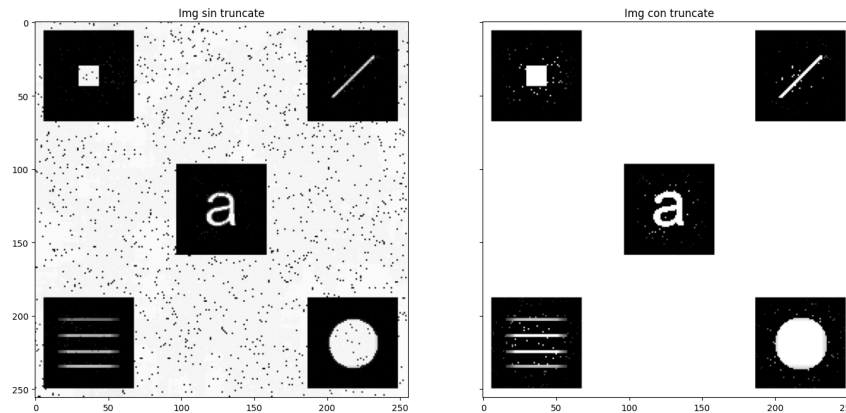
El problema surge cuando la ventana se aproxima a los bordes de la imagen, ya que no hay suficientes píxeles disponibles alrededor para completar la ventana. Para solucionar esto, “cv2.copyMakeBorder” agrega un borde reflejado, lo que asegura que siempre haya suficientes píxeles alrededor de cada píxel.

Luego, se obtienen las dimensiones de la imagen (filas y columnas) y, con esta información, se recorre la imagen píxel por píxel, tanto por fila como por columna. Para cada píxel, se extrae una ventana local de tamaño definido, y a esta ventana se le aplica la ecualización del histograma. Por último, el valor del píxel central de la ventana ecualizada se asigna a la posición (fila, columna) correspondiente en la imagen de resultado.

A continuación se agregan unas imágenes del funcionamiento del código:



Podemos observar que al usar ventanas de diferentes tamaños varía en cómo se muestran las figuras ocultas. Si usamos una ventana muy chica, el detalle es mayor, pero también aparece más ruido porque los píxeles se ajustan usando pocos vecinos. En cambio, con una ventana muy grande, el ruido se reduce bastante, pero se pierden muchos detalles pequeños.



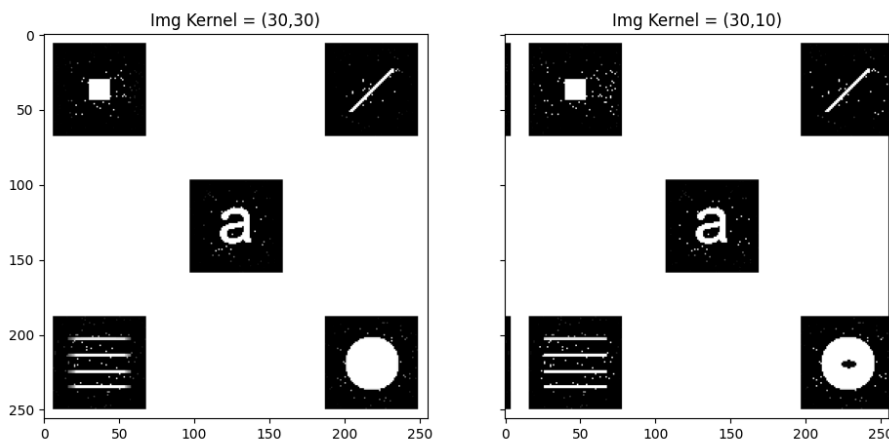
En cambio, cuando utilizamos truncate podemos observar que eliminamos el ruido que se genera en el fondo. Esto hace que se obtenga una imagen más clara de los resultados. En conclusión podemos decir que los objetos ocultos en las imágenes son un cuadrado, líneas con diferente intensidad de brillo, un círculo, una letra “a” cursiva y una línea diagonal.

Un problema que encontramos que tuvimos fue al calcular el borde de la ventana. Inicialmente, usamos $\text{borde} = \max(m, n) // 2$ para generar un borde teniendo en cuenta el peor de los casos. Esto lo que generaba es un problema cuando la ventana no es cuadrada porque estaba generando un borde más grande del necesario en uno de los lados, lo que provoca que la imagen salga recortada. Por ejemplo:

Si se usa una ventana de tamaño (15, 3):

Para las filas, necesitas un borde de tamaño $15 // 2 = 7$.

Para las columnas, necesitas un borde de tamaño $3 // 2 = 1$.



Problema 2 - Corrección de múltiple choice:

Este problema contiene una serie de exámenes resueltos en formato imagen y se debe realizar una corrección automática. Para ello se brinda una lista de las respuestas correctas que pueden ser entre A, B, C y D. su criterio de corrección será la siguiente:

- Si una respuesta contiene más de una opción seleccionada, se considera **incorrecta**.
- Si una respuesta no tiene ninguna opción marcada, también se considera **incorrecta**.

El algoritmo a desarrollar debe tener en cuenta los siguientes puntos:

- tomar únicamente como entrada la imagen de un examen y mostrar por pantalla cuáles de las respuestas son correctas y cuáles incorrectas
- validar los datos del encabezado y mostrar por pantalla si:
 - Name: debe contener al menos 2 palabras y no más de 25 caracteres
 - Date: deben ser 8 caracteres formando una sola palabra
 - Class: un único carácter
- Utilice el algoritmo desarrollado para evaluar las imágenes de exámenes resueltos (archivos examen_.png) e informe cada resultado obtenido.
- Generar una imagen de salida informando los alumnos que han aprobado el examen (con al menos 6 respuestas correctas) y aquellos alumnos que no.

Resolución:

Para resolver este ejercicio realizamos una serie de análisis con funciones las cuales explicaré a continuación:

Función “DetectarLineas()”:

Esta función detecta las líneas horizontales y verticales. Calcula la suma de los píxeles en cada fila y columna para obtener el área donde se pueden encontrar

líneas. Luego, al obtener un número muy elevado, se realiza una división por 255 para trabajar con cifras más pequeñas. Se realiza un filtrado de líneas determinando a través de un umbral cuales sumas son menores o iguales a este, indicando que hay líneas existentes.

La primera fila contiene los datos del estudiante y se almacena para usar en un ejercicio más adelante. Por último devuelve las filas y columnas detectadas además de los datos del estudiante.

Cantidad de líneas horizontales: 6, cantidad de líneas verticales: 4

Name: ESTEBANALVAREZ Date: 11/07/24 Class: 1	
1 The Earth's system that involves all our air is called the _____. A geosphere B hydrosphere C atmosphere D biosphere	6 The gaseous layers of the atmosphere are held to Earth's surface by _____. A their weight B gravity C the sun D none of the above
2 The Earth's system that involves all our water is called the _____. A geosphere B hydrosphere C atmosphere D biosphere	7 78% of the Earth's atmosphere is made up of _____. A nitrogen B oxygen C carbon dioxide D water vapor
3 The Earth's system that involves all our rock is called the _____. A geosphere B hydrosphere C atmosphere D biosphere	8 They layer of the atmosphere we live in is called the _____. A stratosphere. B troposphere. C mesosphere. D exosphere.
4 The Earth's system that involves all living things is called _____. A geosphere B hydrosphere C atmosphere D biosphere	9 Most life in the ocean is found _____. A throughout all its waters. B deep down in the depths. C far from shore. D on the surface and closer to shore.
5 97% of Earth's water is found in _____. A lakes B the ocean C our underground aquifers D the clouds	10 A biomes location on Earth depends upon: _____. A climate B amount of rainfall C temperature D all of the above

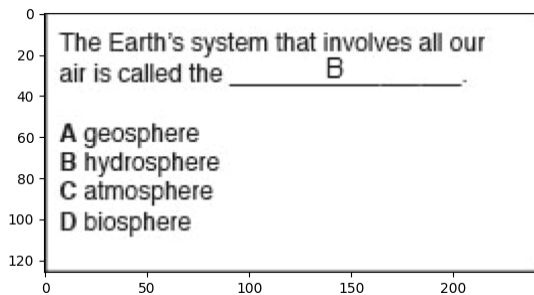
Cantidad de líneas horizontales: 6, cantidad de líneas verticales: 4

Name: LUCAS FERNANDEZ Date: Class: 1	
1 The Earth's system that involves all our air is called the _____. A geosphere B hydrosphere C atmosphere D biosphere	6 The gaseous layers of the atmosphere are held to Earth's surface by _____. A their weight B gravity C the sun D none of the above
2 The Earth's system that involves all our water is called the _____. A geosphere B hydrosphere C atmosphere D biosphere	7 78% of the Earth's atmosphere is made up of _____. A nitrogen B oxygen C carbon dioxide D water vapor
3 The Earth's system that involves all our rock is called the _____. A geosphere B hydrosphere C atmosphere D biosphere	8 They layer of the atmosphere we live in is called the _____. A stratosphere. B troposphere. C mesosphere. D exosphere.
4 The Earth's system that involves all living things is called _____. A geosphere B hydrosphere C atmosphere D biosphere	9 Most life in the ocean is found _____. A throughout all its waters. B deep down in the depths. C far from shore. D on the surface and closer to shore.
5 97% of Earth's water is found in _____. A lakes B the ocean C our underground aquifers D the clouds	10 A biomes location on Earth depends upon: _____. A climate B amount of rainfall C temperature D all of the above

Función "DetectarPreguntas()":

Esta función se encarga de extraer las preguntas del examen utilizando las filas y columnas obtenidas en la función anterior. Se recorre por fila y columna los datos de interés y se obtienen las preguntas, luego se agregan a la lista "preguntas_ord" creado anteriormente.

Cómo se obtienen las preguntas por fila, el orden queda [pregunta 1, pregunta 6, pregunta 2, pregunta 7, pregunta 3, pregunta 8...], es por eso que se deben ordenar. Para ello se toman los índices pares que contienen las 5 primeras preguntas y los impares las últimas 5 preguntas. Esta función devuelve las preguntas ordenadas.



Obtiene de un examen el rectángulo de cada pregunta (ejemplo del examen_1.png).

Función “DetectarRegiones()”:

Esta función se encarga de obtener los renglones donde se encuentra la respuesta de cada pregunta. Crea una lista vacía llamada "renglones" y umbralizar la imagen de forma que solamente quede el guión de la respuesta. La imagen al realizar esto quedará con ruido y obtendrá el renglón deseado.

Se suma los pixeles por fila y los que estén dentro del umbral serán considerados renglones y se agregaran a la lista. Devuelve los renglones obtenidos.

Por ejemplo, al ejecutar esta función en el “examen_2.png”, la función devuelve lo siguiente: [35, 31, 36, 41, 24, 36, 36, 35, 18, 39].

Función “Detectar_Indices()”:

Se encarga de obtener los índices de las respuestas. Crea una lista para almacenar los índices y aplica un umbral a la fila correspondiente, generando una imagen binaria para detectar cambios de intensidad. Estos cambios indican el inicio y fin de cada renglón. Se elimina el valor obtenido en la posición cero

ya que este proviene del recuadro de la pregunta. Luego, se agrega a la lista índices los dos últimos cambios de intensidad debido a que el ruido producido por las letras siempre está al principio. Los índices obtenidos se almacenan y se devuelven.

Por ejemplo, al ejecutar esta función en el “examen_2.png”, la función devuelve lo siguiente: [(90, 203), (109, 222), (101, 214), (90, 197), (179, 232), (169, 229), (78, 145), (77, 178), (170, 224), (10, 64)]

Función “CorregirRespuestas()”:

Esta función corrige los resultados de las respuestas del examen y almacena las correcciones en una lista. Extrae el área de cada respuesta, crea una imagen binaria y calcula las componentes conectadas. Si hay más de una respuesta, se considera incorrecta; si no hay respuestas, se registra como en blanco. Se verifica cuál es la respuesta correcta y se almacena.

Luego las respuestas que poseen solo una letra, se verifica que letra está viendo su área y se almacena la respuesta. Por último, se inicia un contador para ingresar las respuestas obtenidas en orden. Devuelve la corrección final.

Por ejemplo, al ejecutar esta función en el “examen_3.png”, la función devuelve lo siguiente: ['Correcta', 'Correcta', 'Correcta', 'Correcta', 'Correcta', 'Correcta', 'Correcta', 'Correcta', 'Correcta', 'Correcta']

Función “DetectarIndicesDatos()”:

Esta función identifica y extrae los datos del estudiante el cual copia la imagen original y obtiene los datos de una imagen binaria con el renglón de los datos y calcula los índices de cada uno. Devuelve el índice con 3 datos (nombre, fecha y clase).

Por ejemplo, al ejecutar esta función en el “examen_3.png”, la función devuelve lo siguiente: [(58, 252), (294, 371), (418, 550)]

Función “CorregirDatos()”:

Esta función tiene como objetivo verificar la validez de los datos extraídos de la imagen del examen y si es necesario corregir o ajustar los resultados. Para cada índice en “indicesDatos”, se extraen las características de los datos (nombre, fecha, clase) de la imagen usando el método de componentes conectados. Dependiendo de la iteración, se almacenan las estadísticas de los componentes detectados en variables separadas (nombre, fecha, clase).

Para el nombre, se guarda una copia del segmento de la imagen correspondiente a ese campo para usar más tarde. Para verificar si una letra tiene espacio, se calcula la distancia entre el borde derecho de una letra y el borde izquierdo de la siguiente debe ser mayor a un umbral. Por último se devuelve el campo nombre.

Función “MostrarCorrecciones()”:

Muestra los resultados de las correcciones del examen de cada estudiante. Verifica cuántas respuestas han sido calificadas como correctas; si hay al menos seis respuestas correctas, agrega un ":)". De lo contrario, agrega un ":(".

Devuelve la imagen del nombre con el texto agregado.

ESTEBANALVAREZ :(
MARIA :(
MARIA LOPEZ :)
LUCAS FERNANDEZ :(
JUAN PEREZ :)

Luego, se utiliza un bucle para iterar en todos los exámenes y hacer un análisis con todas las funciones explicadas anteriormente. Por último, se almacenan los datos en una lista y se imprime por pantalla el ejercicio D el cual pide generar una imagen de salida informando los alumnos que han aprobado el examen (con al menos 6 respuestas correctas) y aquellos alumnos que no con una carita feliz o carita triste.

Conclusión:

El proyecto nos permitió aplicar conceptos aprendidos en clase, como el algoritmo de histograma local, la función `cv2.copyMakeBorder` para crear bordes y el análisis de imágenes. Experimentamos con el ajuste de umbrales y la función `cv2.connectedComponentsWithStats` para estudiar componentes conectados.

Utilizamos diversas estrategias para resolver problemas, lo que facilitó la manipulación y transformación de imágenes. También incorporamos texto y exploramos otros métodos de transformación, consolidando nuestra comprensión teórica a través de la práctica.