



Universidad Nacional de Rosario
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Tecnicatura Universitaria en Inteligencia Artificial
Procesamiento de Imágenes I - IA 4.4

TRABAJO PRÁCTICO N°1

GRUPO 6

INTEGRANTES:

Cima, Nancy Lucía
Longo, Gonzalo
Sumiacher, Julia



Universidad Nacional de Rosario
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Tecnicatura Universitaria en Inteligencia Artificial
Procesamiento de Imágenes I - IA 4.4

Índice

| | |
|-------------------------|----------|
| Índice | 1 |
| Problema 1 | 2 |
| Descripción y Enunciado | 2 |
| Desarrollo | 3 |
| Conclusión | 7 |
| Problema 2 | 8 |
| Descripción y Enunciado | 8 |
| Desarrollo | 9 |
| Conclusión | 17 |



Problema 1

Descripción y Enunciado

La ecualización del histograma de una imagen es una transformación que pretende obtener para una imagen un histograma con una distribución uniforme. Es decir, que exista el mismo número de píxeles para cada nivel de gris del histograma de una imagen monocroma.

Así, la transformación genera una imagen cuyos niveles de intensidad son igualmente probables, y cubren el rango completo $[0, 1]$. El resultado es una imagen con mayor contraste.

La técnica de ecualización del histograma se puede extender para un análisis local, es decir, se puede realizar una ecualización local del histograma. El procedimiento sería definir una ventana cuadrada o rectangular (vecindario) y mover el centro de la ventana de pixel a pixel. En cada ubicación, se calcula el histograma de los puntos dentro de la ventana y se obtiene de esta manera, una transformación local de ecualización del histograma. Esta transformación se utiliza finalmente para mapear el nivel de intensidad del píxel centrado en la ventana bajo análisis, obteniendo así el valor del píxel correspondiente a la imagen procesada. Luego, se desplaza la ventana un píxel hacia el costado y se repite el procedimiento hasta recorrer toda la imagen.

Esta técnica resulta útil cuando existen diferentes zonas de una imagen que poseen detalles, los cuales se quiere resaltar, y los mismos poseen valores de intensidad muy parecidos al valor del fondo local de la misma. En estos casos, una ecualización global del histograma no daría buenos resultados, ya que se pierde la localidad del análisis al calcular el histograma utilizando todos los píxeles de la imagen.

Desarrolle una función en python, para implementar la ecualización local del histograma, que reciba como parámetros de entrada la imagen a procesar, y el tamaño de la ventana de procesamiento ($M \times N$). Utilice dicha función para analizar la imagen que se muestra en la figura 1 e informe cuáles son los detalles escondidos en las diferentes zonas de la misma. Analice la influencia del tamaño de la ventana en los resultados obtenidos.

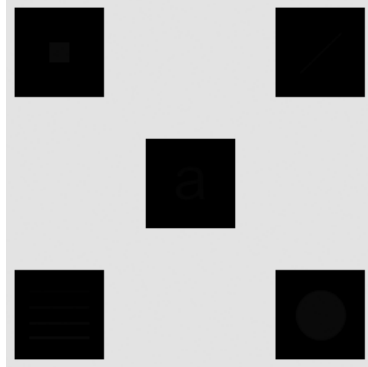


Figura 1: Imagen de entrada (con detalles en diferentes zonas)

Desarrollo

El objetivo de este problema es mejorar el contraste de la imagen en pequeñas secciones de la imagen, lo que puede revelar detalles ocultos. Para esto, desarrollamos una función que realiza una ecualización local del histograma en una imagen en escala de grises, aplicando diferentes tamaños de ventana, y luego muestra los resultados con sus respectivos histogramas. Esta función toma una imagen y un tamaño de ventana para realizar la ecualización local del histograma.

Dentro de la función, primero, se crea una copia de la imagen original para no modificarla directamente. Después, se añade un borde alrededor de la imagen, lo que permite procesar correctamente los píxeles cercanos a los bordes. A continuación, se genera una imagen vacía del mismo tamaño que la original, en la cual se almacenarán los resultados de la ecualización.

El siguiente paso es desplazar una ventana de tamaño fijo a través de toda la imagen. En cada posición de la ventana, se extrae una pequeña región de la imagen y se aplica la ecualización del histograma de forma local. Para este paso, utilizamos el método 'equalizeHist' del módulo cv2.

El valor central de la región ecualizada se asigna a la imagen de salida, de manera que el contraste mejora en cada sección de la imagen. El resultado es una nueva imagen con un contraste mejorado a nivel local.

Para evaluar el efecto de la ecualización, se aplica la función con tres tamaños de ventana diferentes, pequeño (15x15), mediano (30x30) y grande (45x45).



Universidad Nacional de Rosario
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Tecnatura Universitaria en Inteligencia Artificial

Nos pareció suficiente probar con tres tamaños de ventana para comprobar la diferencia de los resultados según el tamaño de la ventana pero también podría probarse el código con más variedad de tamaños.

Finalmente, con la ayuda de la biblioteca matplotlib, se muestra la imagen original junto con su histograma, que refleja la distribución de los tonos de gris. También se visualizan las imágenes resultantes tras aplicar la ecualización con las dos ventanas, cada una acompañada de su propio histograma. Estos histogramas permiten observar cómo se redistribuyen los niveles de brillo después de aplicar la ecualización, demostrando el ajuste del contraste en cada caso.

La visualización obtenida fue:

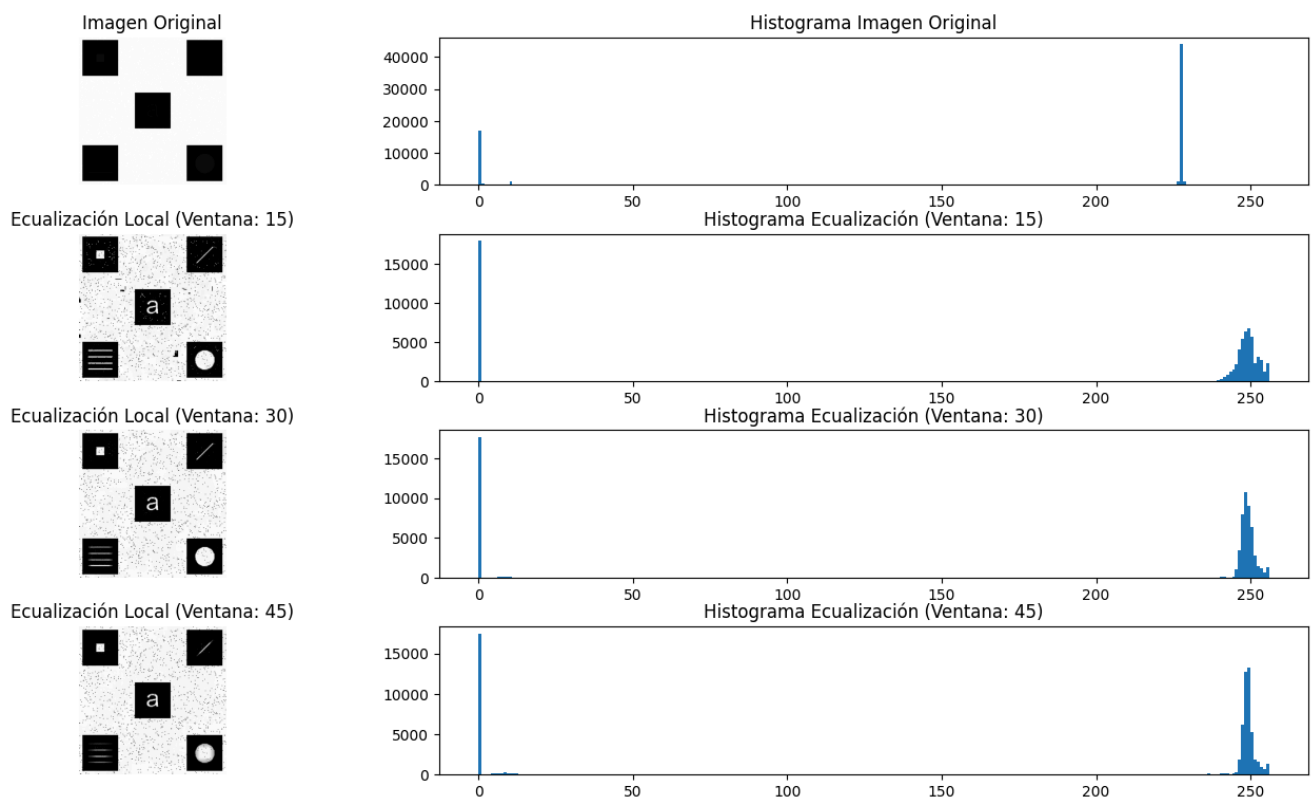


Figura 2: Imagen de salida

A continuación, también adjuntamos las imágenes por separado para ver en más detalles los resultados de las ecualizaciones según el tamaño de la ventana:



Universidad Nacional de Rosario
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Tecnicatura Universitaria en Inteligencia Artificial
Procesamiento de Imágenes I - IA 4.4

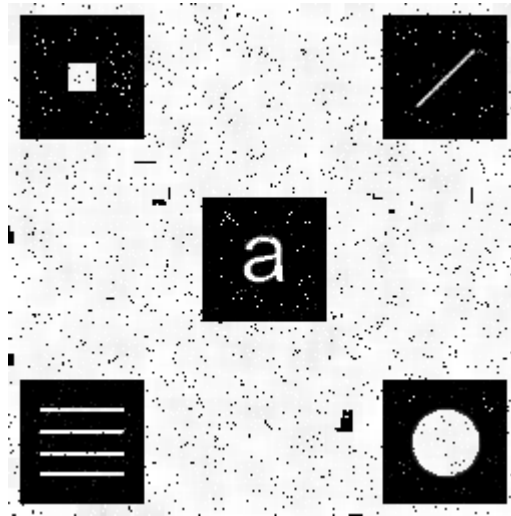


Figura 3: Imagen ecualizada localmente con ventana de tamaño 15x15

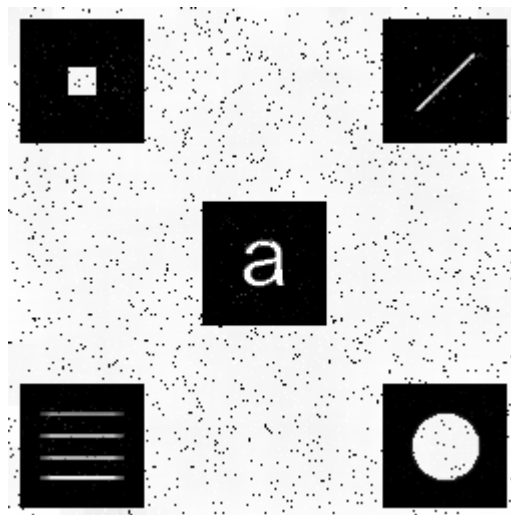


Figura 4: Imagen ecualizada localmente con ventana de tamaño 30x30

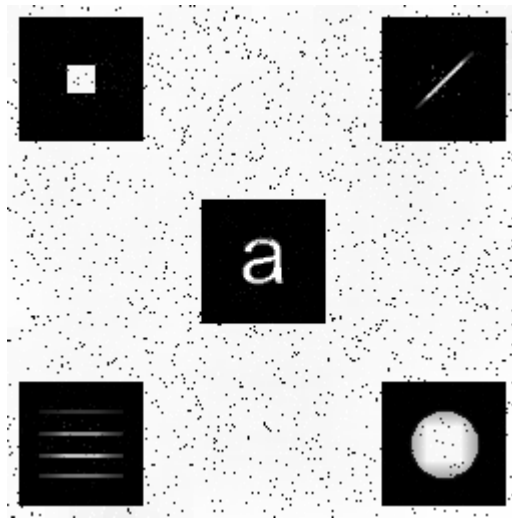


Figura 5: Imagen ecualizada localmente con ventana de tamaño 45x45

Más aún, la imagen con una ecualización “global” se ve así:

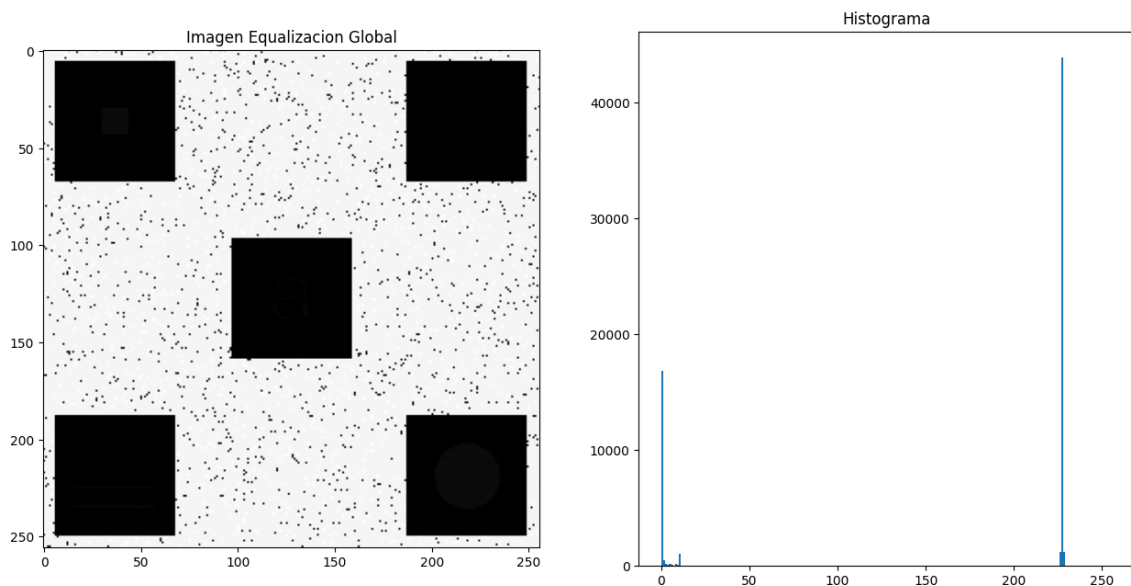


Figura 6: Imagen ecualizada globalmente con su histograma correspondiente

No se encontró ninguna dificultad en particular al resolver este problema.



Conclusión

Notamos que efectivamente ecualización del histograma aplicada localmente, permite ajustar el contraste en diferentes partes de la imagen de forma más precisa y así, detectar objetos o patrones ocultos. En nuestro caso, descubrimos que los detalles escondidos eran un cuadrado, una línea diagonal, una letra “a” minúscula, cuatro líneas horizontales y un círculo.

Además, concluimos que una ventana pequeña revela detalles más finos y permite mejorar el contraste en áreas específicas de la imagen, mientras que una ventana grande afecta áreas más amplias de la imagen y produce un efecto más suave.

Sin embargo, un tamaño de ventana más pequeño puede sobre-procesar ciertas áreas, generando ruido al resaltar detalles muy finos que tal vez no sean relevantes. Por otro lado, un tamaño de ventana demasiado grande puede suavizar la imagen y eliminar detalles importantes. Por tanto, es importante seleccionar el tamaño de ventana adecuado según el objetivo, es decir, según cuánto detalle necesitamos ver y cuánta tolerancia al ruido tenemos.



Problema 2

Descripción y Enunciado

Enunciado:

Se tiene una serie de exámenes resueltos, en formato de imagen, y se pretende corregirlos de forma automática por medio de un script en python.

Para esto, asuma que las respuestas correctas son las siguientes:

1. C 2. B 3. A 4. D 5. B 6. B 7. A 8. B 9. D 10. D

En el caso que alguna respuesta tenga marcada más de una opción, la misma se considera como incorrecta, de igual manera si no hay ninguna opción marcada.

El algoritmo a desarrollar debe resolver los siguientes puntos:

- a. Debe tomar únicamente como entrada la imagen de un examen (no usar como dato las coordenadas de las preguntas) y mostrar por pantalla cuáles de las respuestas son correctas y cuáles incorrectas.
- b. Con la misma imagen de entrada, validar los datos del encabezado y mostrar por pantalla el estado de cada campo teniendo en cuentas las siguientes restricciones:
 - i. Name: debe contener al menos dos palabras y no más de 25 caracteres.
 - ii. Date: deben ser 8 caracteres formando una sola palabra.
 - iii. Class: un único carácter.
- c. Utilice el algoritmo desarrollado para evaluar las imágenes de exámenes resueltos (archivos examen_<id>.png) e informe cada resultado obtenido.
- d. Generar una imagen de salida informando los alumnos que han aprobado el examen (con al menos 6 respuestas correctas) y aquellos alumnos que no. Esta imagen de salida debe tener los “crop” de los campos Name del encabezado de todos los exámenes del punto anterior y diferenciar de alguna manera aquellos que corresponden a un examen aprobado de uno desaprobado.

En resumen, el objetivo del problema es desarrollar un script en Python que corrige automáticamente exámenes en formato de imagen. El script debe evaluar las respuestas seleccionadas en comparación con un conjunto de respuestas correctas, validar los datos del encabezado (Nombre, Fecha y Clase) según restricciones específicas y generar una imagen de salida que informe sobre la aprobación de los



alumnos (al menos 6 respuestas correctas), diferenciando visualmente entre aprobados y desaprobados.

Desarrollo

Dado que para este problema debemos tener en cuenta distintas cosas, decidimos crear funciones para modularizar nuestro código. Hablaremos un poco de cada una, pero para eso debemos tener en cuenta que usaremos la imagen examen_1.png de ejemplo.

Name: ESTEBANALVAREZ Date: 11/07/24 Class: 1

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

Figura 7: Imagen de entrada



Además, las imágenes son binarizadas en varias etapas para facilitar la detección de estructuras, celdas y respuestas.

Ahora si, las funciones son:

1. obtener_celdas:

Esta función procesa la imagen binarizándola (con un umbral), detecta las líneas horizontales y verticales del examen (para identificar la estructura de la hoja) y extrae las celdas correspondientes a las respuestas. Cada celda contiene una imagen de una pregunta.

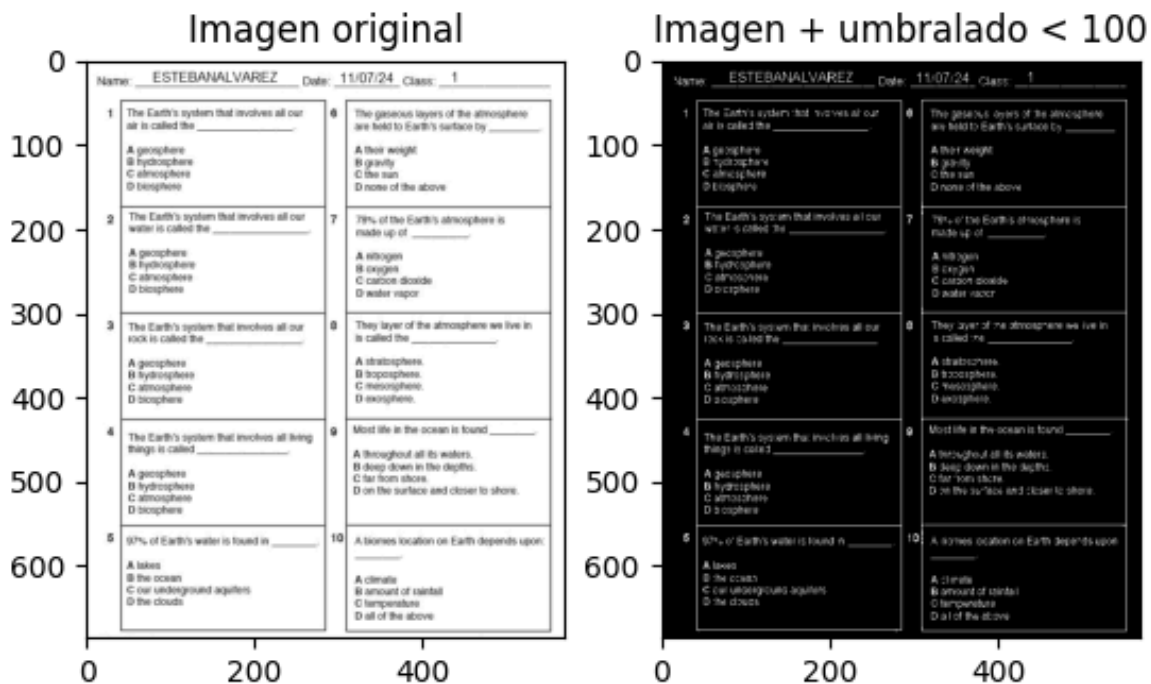


Figura 8: Imagen original e imagen binarizada con umbral menor a 100

Luego de este paso, procedemos a detectar las diferentes celdas



Preguntas del examen

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

Figura 9: celdas del examen

2. obtener_respuestas:

Procesa las celdas obtenidas y, mediante binarización (umbralado), localiza la respuesta seleccionada por el alumno en cada celda. Las respuestas se devuelven como una lista que incluye la pregunta y la región de la respuesta marcada.

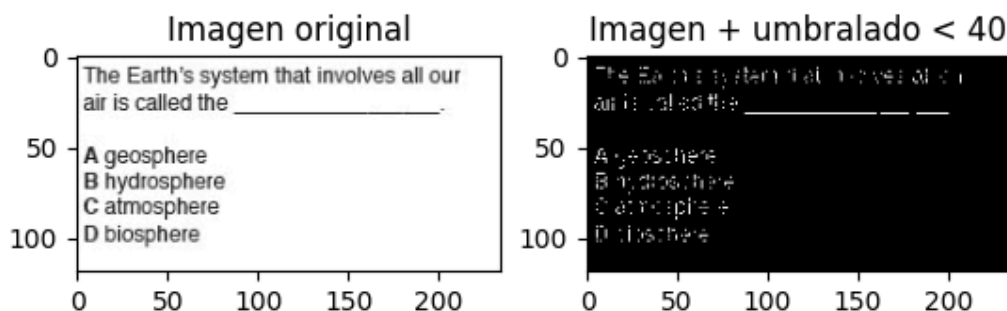


Figura 10: Imagen original e imagen binarizada con umbral menor a 40



Respuestas del examen

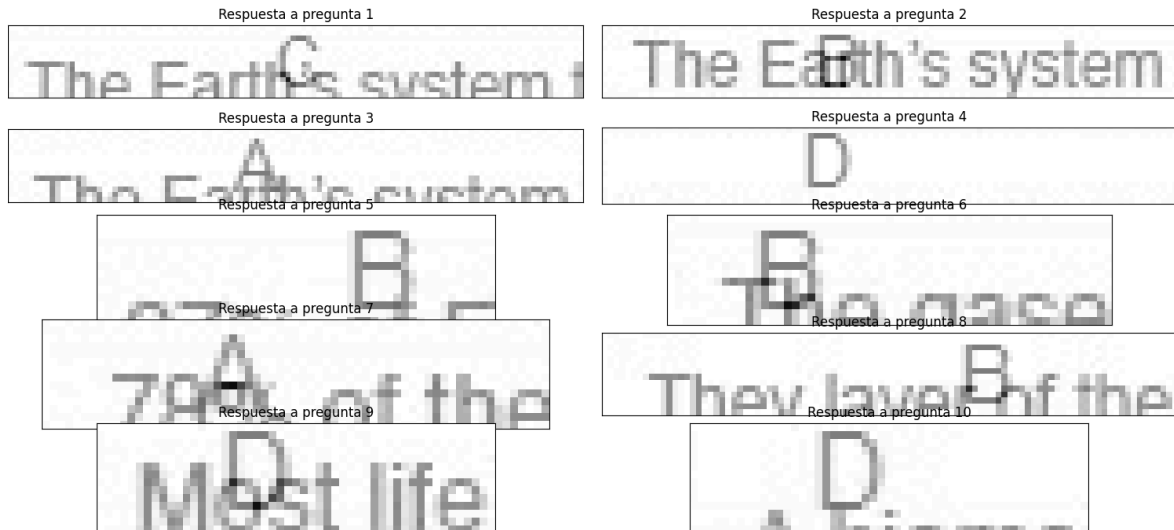


Figura 11: área de la respuesta de cada celda

3. obtener_encabezado:

Similar a las funciones anteriores, pero enfocado en extraer los campos de "Nombre", "Fecha" y "Clase" del encabezado de la imagen. Utiliza umbrales y operaciones de suma de píxeles para identificar las áreas relevantes de la imagen.

Partimos de la imagen original, la binarizamos y luego identificamos las áreas del enunciado.

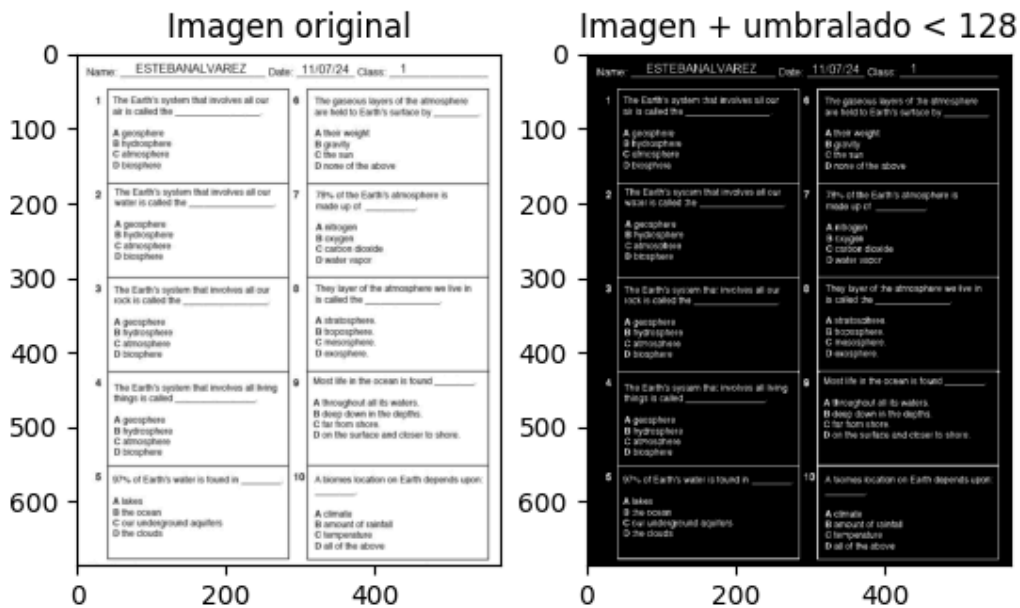


Figura 12: Imagen original e imagen binarizada con umbral menor a 128

| Campo Nombre | Campo Fecha | Campo Clase |
|----------------|-------------|-------------|
| ESTEBANALVAREZ | 11/07/24 | 1 |

Figura 13: Encabezados

4. obtiene_letra: procesa una imagen para determinar si contiene una letra, y en caso de ser así, clasifica la letra como A, B, C o D en función de sus contornos. Esta función recibe como parámetro la imagen en escala de grises que contiene la respuesta. Esta imagen se convierte a una imagen binaria, aplicando un umbral de 129. Todos los píxeles con valores menores a 129 se convierten en 255 (blanco), y los demás en 0 (negro). Luego, se utiliza cv2.connectedComponentsWithStats para detectar las áreas conectadas en la imagen binaria (en este caso, áreas que representan letras).



Se detectan los contornos de la letra utilizando `cv2.findContours` y luego, la clasificación de la letra se hace en función de la cantidad de contornos detectados y su forma:

- **1 contorno:** Se clasifica como letra C.
- **3 contornos:** Se clasifica como letra B.
- **2 contornos:** Se clasifica como letra A o D, dependiendo del número de puntos del primer contorno:
 - Si el contorno tiene menos de 15 puntos, se clasifica como A.
 - En caso contrario, se clasifica como D.

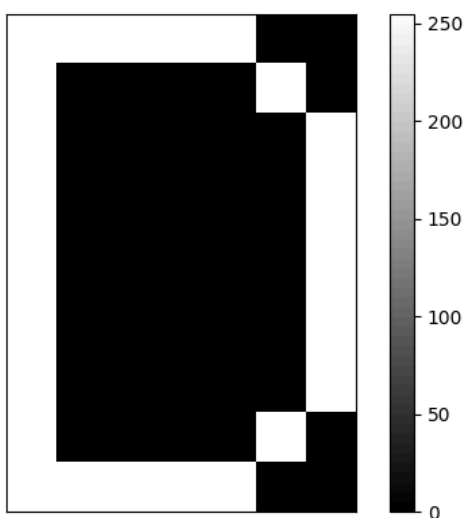


Figura 14: Ejemplo de letra (“D”)

5. `imprimir_correccion`: compara las respuestas obtenidas de imágenes con las respuestas correctas y muestra el resultado de cada pregunta. Recibe como **parámetros**

- **respuestas:** Una lista de tuplas, donde cada tupla contiene un identificador y una imagen (representando la respuesta a una pregunta).
- **correctas:** Una lista que contiene las respuestas correctas correspondientes a cada pregunta.

Se recorre cada respuesta en `respuestas` para obtener la letra correspondiente, es decir, para cada tupla `rta`, se extrae la imagen (`img`) y se llama a la función `obtiene_letra` para detectar la letra en la imagen.



Se inicializa una lista corrección para almacenar el resultado de la comparación entre las respuestas obtenidas y las correctas. Se compara cada letra en rtas con la correspondiente en correctas y se añade True o False a la lista corrección según si la respuesta fue correcta o incorrecta.

Finalmente, se imprimen los resultados de la corrección:

Si c es True, se imprime "Pregunta X OK"; si es False, se imprime "Pregunta X MAL", donde X es el número de la pregunta (comenzando desde 1).

```
Correccion de examen:  
  
Pregunta 1 MAL  
Pregunta 2 MAL  
Pregunta 3 MAL  
Pregunta 4 OK  
Pregunta 5 MAL  
Pregunta 6 OK  
Pregunta 7 OK  
Pregunta 8 MAL  
Pregunta 9 MAL  
Pregunta 10 OK
```

Figura 15: Ejemplo de salida de 'imprimir_correccion'

6. `imprime_encabezado`: se encarga de verificar la corrección de un encabezado, que consta de tres campos: nombre, fecha y clase, extrayendo la información de las imágenes y evaluando su validez. Recibe como parámetro una lista que contiene tuplas. Cada tupla tiene un identificador (índice) y una imagen (que representa un campo del encabezado), es decir, la salida de 'obtener_encabezado'.

Primero, se inicializan las variables ``n``, ``f``, y ``c`` para indicar si el nombre, la fecha y la clase son correctos, respectivamente. Inicialmente, todos se marcan como "OK":

Luego, se extraen las imágenes correspondientes a cada campo y se convierten las imágenes a formato binario mediante un umbral.

Se utiliza ``cv2.connectedComponentsWithStats`` para contar los componentes conectados en la imagen binarizada del nombre y mediante esto:

- Se evalúa la cantidad de letras encontradas.



- Se determina si hay espacios en la palabra comprobando la distancia vertical entre las letras.
- Se marca como "MAL" si hay más de 25 letras o si no hay espacios (indicando que podría ser una sola palabra).

De forma similar al análisis del nombre, se evalúa el campo fecha:

- Se espera que el campo tenga exactamente 8 letras y no contenga espacios.
- Se marca como "MAL" si no se cumplen estas condiciones.

Se realiza un análisis pareció para el campo clase:

- Se espera que el campo clase tenga exactamente 1 letra (un solo componente).
- Se marca como "MAL" si hay más de un componente.

```
Encabezado:
-----
Nombre:  MAL
Fecha:   OK
Clase:   OK
-----
```

Figura 16: Ejemplo de salida de 'imprime_encabezado'

Luego de la definición de las funciones, se cargan todas las imágenes en escala de grises, se identifica cada pregunta en la hoja y las opciones marcadas (A, B, C o D), se extraen los datos del encabezado como nombre, fecha y clase y el script comparará las respuestas del alumno con una lista de respuestas correctas predefinidas para cada pregunta, además, de ver si los campos del encabezado cumplen con los requisitos establecidos.

Por último, se genera una imagen de salida informando los alumnos que han aprobado el examen (con al menos 6 respuestas correctas) y aquellos alumnos que no.



Conclusión

Se desarrollaron funciones para segmentar las celdas de respuestas en las imágenes de los exámenes. Esto se logró utilizando técnicas de umbralización que permitieron resaltar las áreas de interés y facilitar la identificación de las respuestas seleccionadas.

Las respuestas se analizan para asegurar que solo una opción esté marcada, lo cual es crucial para mantener la integridad del proceso de corrección.

Además de las respuestas, se implementó un análisis para verificar los campos de datos personales (Nombre, Fecha, Clase). Se utiliza un enfoque similar al de las respuestas, validando la cantidad de letras y la presencia de espacios para asegurar la correcta recolección de la información.

Nos gustó mucho poder desarrollar este programa puesto que nos dio una visión más práctica de la utilidad del procesamiento de imágenes y de las herramientas vistas en clase. No encontramos problemas en particular más allá de la complejidad misma del problema.

Creemos que nuestro programa es bueno, pero obviamente podría mejorarse. Por ejemplo, incorporando un aspecto más interactivo con una interfaz gráfica y permitiendo que el usuario cargue sus propios exámenes multiple choice a corregir o hasta la posibilidad de enviar los resultados automáticamente por mail a los alumnos.

Además, faltó terminar el último punto.