

Universidad Nacional de Rosario (UNR)

Facultad de Ciencias Exactas, Ingeniería y Agrimensura (FCEIA)

Tecnicatura Universitaria en Inteligencia Artificial

Asignatura: Procesamiento de Imágenes I

Alumnos: Avecilla Tomás Valentino, Calcia Franco Nicolás

Fecha de Entrega: 21/10/2024

Informe Trabajo Práctico 1

En este trabajo práctico, se abordan dos problemas diferentes de procesamiento de imágenes. El primero consiste en la implementación de una técnica de ecualización local de histograma para resaltar detalles en zonas específicas de una imagen. El segundo, en la automatización de la corrección de exámenes de múltiple choice a partir de imágenes, detectando respuestas correctas e incorrectas, así como validando los campos de datos personales.

Índice

Problema 1: Ecualización Local de Histograma	3
Desarrollo	3
Implementación	3
Resultados	3
Problema 2: Corrección Automática de Examen de Múltiple Choice	4
Desarrollo	4
Implementación	4
1. Cargar y mostrar la imagen original	4
2. Umbralización de la imagen	4
3. Obtener contornos	5
4. Recortar el encabezado	5
5. Cortar y procesar las líneas del encabezado	5
6. Obtener y validar letras y palabras	5
7. Recorte y validación de preguntas	6
8. Evaluación del resultado	6
9. Combinar las imágenes de los nombres y resultados	6
10. Mostrar o guardar la imagen	6
Resultados	7
Conclusión	8
Bibliografía	8

Problema 1: Ecualización Local de Histograma

Desarrollo

La función desarrollada recibe como parámetros de entrada la imagen y el tamaño de la ventana. Para cada píxel, se calcula el histograma dentro de la ventana correspondiente y se realiza la ecualización en esa región. Se recorre la imagen aplicando esta técnica de forma que se mejora el contraste localmente.

Implementación

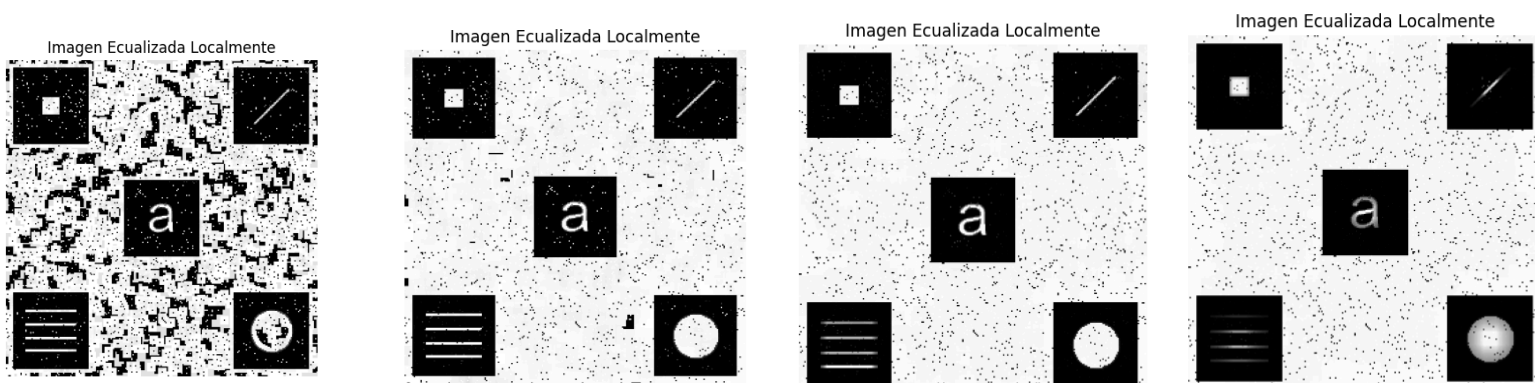
- Se utilizó la función 'cv2.copyMakeBorder' para agregar bordes a la imagen, permitiendo realizar el análisis de la ventana sin problemas en los extremos de la misma.
- El histograma local se calculó para cada ventana, y se aplicó la ecualización correspondiente para mejorar el contraste en áreas específicas.

Resultados

Probamos la ecualización local en la imagen y los resultados fueron claros: los detalles ocultos, quedan resaltados.

Tamaño de ventana:

A continuación, se incluye un análisis visual de los resultados con diferentes tamaños de ventana. Vemos que a medida que aumenta el tamaño de la ventana se puede perder un poco de detalles pequeños, el mejor tamaño estaría entre 20 y 30.



Problema 2: Corrección Automática de Examen de Múltiple Choice

Desarrollo

En este trabajo, se desarrolló un algoritmo en Python para corregir automáticamente un examen de opción múltiple.

El objetivo principal es identificar las respuestas correctas e incorrectas y validar los datos personales en el encabezado del formulario.

El examen consta de 10 preguntas con cuatro opciones (A, B, C, D) y un encabezado que incluye campos para el nombre del alumno, la fecha y la clase. Las respuestas correctas están definidas previamente, y el sistema evalúa las respuestas marcadas para determinar su validez.

Implementación

1. Cargar y mostrar la imagen original

- Se desarrolló la función ``cargar_imagen`` para leer la imagen en escala de grises utilizando OpenCV(``cv2``).

Esto facilita el procesamiento posterior, eliminando la complejidad de trabajar con imágenes en color.

2. Umbralización de la imagen

- La función ``umbralizar_imagen`` convierte la imagen a blanco y negro mediante un umbral binario inverso. Esto facilita identificar los contornos y detectar elementos como letras o líneas horizontales.

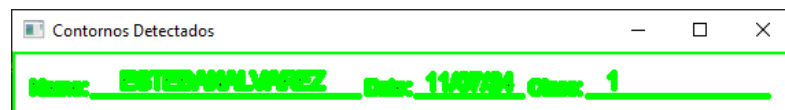
3. Obtener contornos

- La función 'obtener_contornos' se encarga de encontrar todos los contornos de la imagen. Los contornos detectan los límites de los elementos (letras, líneas, etc.), y luego se ordenan según su área para poder procesar los elementos más grandes primero, como por ejemplo las columnas que cuentan con 4 preguntas.



4. Recortar el encabezado

- La función 'recortar_encabezado' se utiliza para recortar la parte superior de la imagen, que corresponde al encabezado del examen, donde se encuentra la información del estudiante (Nombre, Clase, Fecha). Para identificar el encabezado, se utiliza uno de los contornos previamente calculados que detecta la primera columna de preguntas; descartamos esa parte y nos quedamos con lo que está por encima, que corresponde al encabezado.



5. Cortar y procesar las líneas del encabezado

- La función 'recortar_lineas_encabezado' detecta las líneas horizontales dentro del encabezado (que corresponden al Nombre, Clase y Fecha). Luego, con la función 'recortar_hacia_arriba', se recorta cada línea individualmente desde el inicio hasta el final de la línea detectada.

6. Obtener y validar letras y palabras

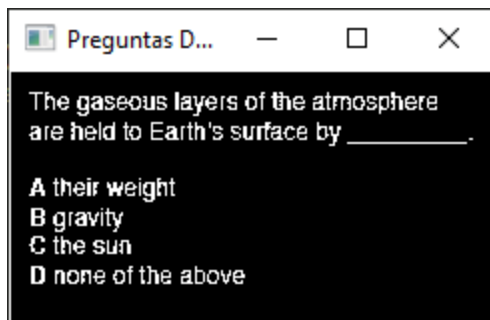
- Se utiliza ``obtener_indices`` para detectar los índices de las letras (los bordes donde empiezan y terminan) en cada línea. Luego, funciones como ``contar_espacios``, ``validar_nombre``, ``validar_clase`` y

``validar_fecha`` validan si los campos del encabezado cumplen las reglas:

- **Nombre:** Debe tener al menos dos palabras y un máximo de 25 caracteres.
- **Clase:** Debe ser una sola letra.
- **Fecha:** Debe tener exactamente 8 caracteres.

7. Recorte y validación de preguntas y respuestas

- Después de procesar el encabezado, se usa ``recortar_preguntas`` para identificar y recortar las preguntas del examen.



- Para cada pregunta, con ``recortar_respuesta_desde_linea``, detectamos la respuesta marcada (si es que hay una) mediante la detección de contornos, y luego utilizamos ``validar_pregunta`` para verificar si la respuesta es válida, es decir, que solo una opción esté marcada.



Luego usamos ``validar_pregunta`` para asegurar que la pregunta:

- Contenga una respuesta
- que la respuesta sea una sola

Esto lo hicimos de la misma manera que validamos el encabezado, usando índices.

A su vez, esta misma función usa los stats de la función `cv2.connectedComponentsWithStats` para obtener la imagen de la letra encontrada y pasárselo a `'identificador_letra'`

7.1 `'identificador_letra()'`

Esta función identifica letras (A, B, C o D) basándose en la cantidad y las características de los contornos encontrados en una imagen binarizada de la letra. El flujo general es:

- **Inversión de la imagen:** Se invierte la imagen binaria para que las letras sean blancas sobre un fondo negro.
- **Detección de contornos:** Se detectan los contornos en la imagen binaria.
- **Clasificación de la letra:**
 - Si hay un solo contorno, la letra es 'C'.
 - Si hay dos contornos, se verifica si uno de ellos es una línea vertical para diferenciar entre 'A' y 'D'. Si se encuentra una línea vertical, se identifica como 'D', de lo contrario, como 'A'.
 - Si hay tres contornos, la letra es 'B'.
 - Si no coincide con ninguno de los casos, se considera como una respuesta inválida.



7.2 Lógica

Usamos una lógica sencilla para donde mapeamos las respuestas correctas desde un diccionario con las preguntas corregidas, cabe aclarar que el diccionario tiene las respuestas ingresadas de modo que corresponden a cómo se van ingresando las preguntas ya que nos dimos cuenta que el código recorta las preguntas en la secuencia (6 - 2 - 4 - 3 - 1 - 10 - 9 - 8 - 7 - 5).

Esto fue un problema a resolver y debe pasar tal vez por la cantidad de contenido que detecta dentro de la imagen.

8. Combinar las imágenes de los nombres y resultados

- Finalmente, con la función `combinar_imagenes`, se crea una imagen final que combina el recorte del nombre del estudiante con un borde verde si aprobó o rojo si no aprobó. Además, se agrega el nombre del estudiante en el centro de la imagen, con un tamaño de fuente ajustado automáticamente.

9. Mostrar o guardar la imagen

- La imagen generada se muestra con `matplotlib` y también puede guardarse en un archivo de salida usando OpenCV.

Resultados

Resultados de los Exámenes

```
Imagen 1
Validación Name: MAL
Validación Class: OK
Validación Date: OK
Pregunta 1: Incorrecto - No hay índices
Pregunta 2: Incorrecto - No hay índices
Pregunta 3: Incorrecto - No hay índices
Pregunta 4: Incorrecto - No hay índices
Pregunta 5: Incorrecto - No hay índices
Pregunta 6: Incorrecto - No hay índices
Pregunta 7: Incorrecto - No hay índices
Pregunta 8: Incorrecto - No hay índices
Pregunta 9: Incorrecto - No hay índices
Pregunta 10: Incorrecto - No hay índices
```



```
Imagen 2
Validación Name: MAL
Validación Class: OK
Validación Date: OK
Pregunta 1: Bien
Pregunta 2: Incorrecto - Más de 2 índices
Pregunta 3: Bien
Pregunta 6: Bien
Pregunta 7: Incorrecto - No hay índices
Pregunta 8: Incorrecto - No hay índices
Pregunta 9: Bien
Pregunta 10: Incorrecto - No hay índices
```

```
Imagen 3
Validación Name: OK
Validación Class: OK
Validación Date: OK
Pregunta 1: Bien
Pregunta 2: Bien
Pregunta 3: Bien
Pregunta 4: Bien
Pregunta 5: Bien
Pregunta 6: Bien
Pregunta 7: Bien
Pregunta 8: Bien
Pregunta 9: Bien
Pregunta 10: Bien
```

```
Imagen 5
Validación Name: OK
Validación Class: OK
Validación Date: OK
Pregunta 1: Bien
Pregunta 2: Bien
Pregunta 3: Bien
Pregunta 4: Bien
Pregunta 5: Bien
Pregunta 6: Bien
Pregunta 7: Bien
Pregunta 8: Bien
Pregunta 9: Bien
Pregunta 10: Bien
```

Conclusión

En este trabajo se implementaron dos técnicas de procesamiento de imágenes que permitieron:

1. Mejorar el contraste local de una imagen.
2. Automatizar la corrección de exámenes.

Ambas aplicaciones demostraron ser efectivas en sus respectivos campos de aplicación. Notamos la importancia de seleccionar adecuadamente los parámetros (como el tamaño de la ventana en la ecualización o la conectividad) para obtener los mejores resultados.

Bibliografía

- Documentación de OpenCV.
- Material provisto en clase.