



Procesamiento de Imágenes I

Informe del TP1

Integrantes:

Mateo Mansilla

Nicolas Duclos

Abril 2025

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

Universidad Nacional de Rosario

Av. Pellegrini 250 - 2000 Rosario – Argentina

Problema 1: Ecualización local del histograma

Descripción:

En este ejercicio implementamos desde cero una función para aplicar **ecualización local del histograma** sobre una imagen. El objetivo era resaltar detalles que pudieran estar ocultos por falta de contraste en distintas regiones de la imagen.

A diferencia de la ecualización global, la versión local toma pequeñas ventanas centradas en cada píxel y realiza la transformación de histograma únicamente dentro de esa región. De este modo, mejora el contraste adaptativamente en cada zona de la imagen.

Se trabajó con la imagen proporcionada *Imagen_con_detalles_escondidos.tif*, la cual contenía detalles poco visibles a simple vista.

Inconvenientes a resolver:

- Carga de la imagen: En un primer momento, OpenCV no podía abrir correctamente la imagen *.tif* debido a un error en la ruta o a un problema de compatibilidad. Resolvimos esto verificando extensiones, rutas absolutas y usando correctamente `cv2.IMREAD_GRAYSCALE`.
- Error al aplicar operaciones sobre *None*: En una ocasión, el programa arrojaba un error al intentar obtener `.shape` de una imagen no cargada correctamente. Añadimos validaciones para asegurarnos de que la imagen se cargara antes de procesarla.

Técnicas utilizadas:

- **OpenCV:**
 - Lectura y visualización de imágenes
 - Aplicación de bordes replicados (`cv2.copyMakeBorder`) para permitir el desplazamiento de ventanas
 - Cálculo de histogramas (`cv2.calcHist`) y normalización
- **Numpy:**
 - Manejo de arrays para construir la imagen salida
 - Cálculo de funciones de distribución acumuladas (CDF)
- **Matplotlib:**
 - Visualización de los resultados comparativos antes y después de la ecualización

Conclusiones

Pudimos implementar correctamente un algoritmo de ecualización local del histograma, que demostró ser útil para resaltar detalles que estaban invisibles a simple vista.

Este ejercicio nos ayudó a reforzar los conceptos de procesamiento por regiones, mejora del contraste y manipulación de píxeles

Problema 2: - Corrección de Examen Multiple Choice

Descripción:

El objetivo del trabajo fue desarrollar un sistema automático para la corrección de exámenes tipo multiple choice, a partir de imágenes escaneadas. El sistema debía reconocer automáticamente las respuestas seleccionadas por los estudiantes y determinar si aprobaban o no, en base a un criterio establecido. Además, debía generar un resumen visual con el nombre del estudiante y su estado de aprobación.

Inconvenientes a resolver:

- **Reconocimiento de palabras en el encabezado:** Uno de los principales problemas fue que no logramos identificar correctamente los contornos de palabras completas en el encabezado. Debido a esto, no pudimos implementar una validación de nombres a través de OCR, y nos vimos obligados a trabajar con caracteres individuales o bien con recortes fijos de celdas.
- **Segmentación del encabezado:** Para solucionar el problema anterior, decidimos dividir el encabezado en celdas (boxes) utilizando operaciones morfológicas que nos permitieran detectar las líneas de la tabla. De esta manera, pudimos acceder directamente a la casilla que contenía el nombre del alumno (por ejemplo, el box 2), sin necesidad de detectar palabras completas.
- **Procesamiento de líneas:** Fue necesario ajustar cuidadosamente los tamaños de los kernels morfológicos para detectar correctamente las líneas horizontales y verticales del encabezado, ya que pequeñas variaciones en el escaneo de las hojas afectaban la detección.
- **Manejo de nombres y resultados de forma automática:** Al principio se trabajaba con rutas manuales para acceder a los nombres de los alumnos y sus respectivos estados de aprobación. Luego se modificó el código para que funcione de forma automática, recorriendo la estructura de datos generada durante el procesamiento.

Técnicas utilizadas:

- OpenCV: Lo utilizamos para todas las operaciones de procesamiento de imágenes:
 - Lectura de imágenes (cv2.imread)
 - Conversión a escala de grises (cv2.cvtColor)
 - Binarización (cv2.threshold)
 - Operaciones morfológicas (cv2.morphologyEx) para detectar líneas verticales y horizontales
 - Detección de contornos (cv2.findContours) para segmentar las celdas
 - Recorte de subimágenes para cada casilla de respuesta y el encabezado
 - Escritura y visualización de resultados (cv2.imwrite, cv2.imshow)
- Reconocimiento de respuestas: Comparamos las posiciones de los círculos marcados en cada pregunta contra una clave de respuestas esperada para determinar cuántas eran correctas.
- Recorte del encabezado: Extrajimos la casilla que contenía el nombre del alumno a partir de coordenadas calculadas desde las líneas detectadas en el encabezado.
- Generación automática del resumen: Creamos una imagen que concatena todas las filas correspondientes a los estudiantes, con su nombre y estado ("APROBADO"/"DESAPROBADO").

Conclusiones:

Pudimos poner en práctica diversas herramientas de procesamiento de imágenes y manejo de estructuras de datos. A lo largo del desarrollo nos enfrentamos a varios desafíos, principalmente relacionados con la segmentación del encabezado y el reconocimiento de marcas en las casillas de respuestas.

A pesar de estas dificultades iniciales, logramos construir un sistema funcional que automatiza tanto la corrección de exámenes tipo multiple choice como la generación de un resumen visual con el estado de cada alumno.