



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



---

**Universidad Nacional de Rosario -**

**Facultad de Ciencias Exactas, Ingeniería y Agrimensura**

**Tecnicatura Universitaria en Inteligencia Artificial**

**Asignatura:** Procesamiento de Imágenes

# TRABAJO PRÁCTICO N° 1

**Alumnos:** Accurso, Agustín

Barbarroja, Federico

Cena, Lautaro

**Equipo Docente:** Gonzalo Sad

Juan Manuel Calle

Joaquín Allione

El presente Trabajo Práctico N° 1 consta de dos problemas, que fueron resueltos en Python, utilizando Visual Studio Code para realizar los scripts y librerías vistas en clase, como OPENCV, NUMPY y MATPLOTLIB. Durante este informe se analizarán los resultados obtenidos y se elaborarán conclusiones.

### Problema 1 - Ecualización local de histograma

En este problema se solicita utilizar la Ecualización Local del Histograma, técnica que consiste en definir una ventana cuadrada o rectangular (vecindario) y mover el centro de la misma de píxel en píxel, calculando el histograma de cada punto para lograr dicha transformación. Para ello, se debe definir una función en Python, utilizando herramientas propias del lenguaje, sumadas a otras de las librerías OPENCV y NUMPY.

En síntesis, dicha función recorre la imagen píxel por píxel, tomando en cada posición un vecindario definido por el tamaño de la ventana. Para cada vecindario, calcula el histograma local y su función de distribución acumulada (CDF), que luego se usa para recalcular el valor del píxel central, ajustando su intensidad según la distribución de su entorno. De este modo, se obtiene una nueva imagen con contraste mejorado de forma local, resaltando detalles que no serían visibles con una ecualización global. (Otros detalles están aclarados en el código).

Luego de analizar la imagen con distintos tamaños de ventana, concluyendo con 7x7 como un tamaño acorde para una mejor visualización. Se utilizó un filtro de mediana para eliminar ruido salt-pepper pero no resultó en una imagen con notoria diferencia en cuanto a la visualización de dichos objetos. Concluyendo, podemos decir que los objetos detectados dentro de los ‘cuadrados’ negros son:

- Un **cuadrado** pequeño, en la esquina superior izquierda.
- Una **línea diagonal**, en la esquina superior derecha.
- Una **letra ‘a’** minúscula, en el centro.
- Cuatro **líneas horizontales**, en la esquina inferior izquierda.
- Un **círculo**, en la esquina inferior derecha.

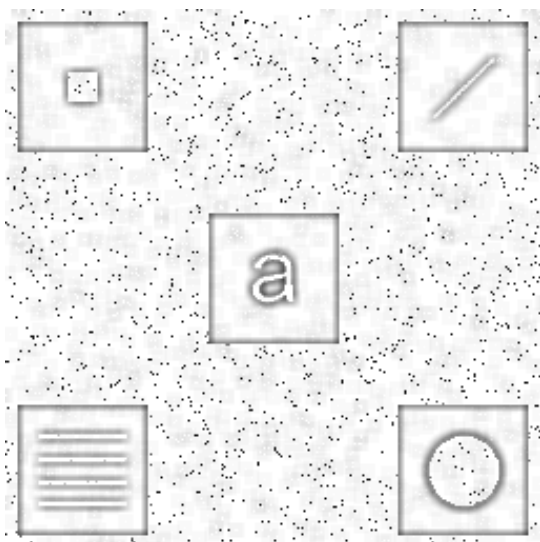


Imagen 1: Resultado de aplicar Ecualización Local del Histograma con una ventana de 7x7.



## Problema 2 - Validación de formulario

Este problema consiste en desarrollar un algoritmo para validar automáticamente un conjunto de formularios, siendo concretamente el objetivo principal determinar si los campos de cada formulario están correctos (OK) o incorrectos (MAL), siguiendo un conjunto de restricciones.

Se puede abreviar el desafío en los siguientes pasos:

1. Tomar como entrada imágenes de formularios y mostrar por pantalla los resultados de la validación de cada campo.
2. Aplicar esta validación de forma cíclica sobre un conjunto de imágenes.
3. Generar una imagen de salida de resumen, que muestre los *crops* del campo "Nombre y apellido" de cada formulario junto con un indicador visual que diferencie a los formularios correctos de los incorrectos (aquellos con un campo incorrecto como mín.).
4. Generar un archivo CSV que registre los resultados de la validación para cada campo y cada formulario, incluyendo un ID único.

La estrategia para la resolución fue resumidamente la siguiente:

1. **Segmentación y detección de líneas:** la imagen se umbralizó y se sumó el valor de los píxeles por filas y columnas para detectar la posición de las líneas divisorias del formulario.
2. **Segmentación de preguntas:** para las preguntas, cada celda se dividió en dos sub-celdas analizando el pico de densidad de píxeles (la línea vertical central) usando `np.argmax(col_sums)`.
3. **Conteo y Limpieza de Componentes:**
  - Para validar el número de caracteres o la presencia de marcas, se utilizó el algoritmo `cv2.connectedComponentsWithStats` de `opencv`.
  - El ruido (fragmentos de líneas divisorias) se eliminó mediante un filtrado por área, descartando aquellas componentes conectadas que poseían un área muy pequeña.
  - Se desarrolló la función que, basándose en el conteo de componentes, calcula el número de palabras analizando la distancia horizontal entre componentes.
4. **Validación (OK/MAL):**
  - `validar_por_cantidad`: aplica las restricciones de número total de componentes (caracteres).
  - `validar_preguntas`: aplica la regla de una única marca por pregunta.
  - `validar_palabras`: aplica las restricciones de número de palabras.
5. **Generación del Resumen Visual:** se desarrolló la función que toma el crop del "Nombre y apellido" y le añade un margen lateral, con un OK para formularios correctos y X para incorrectos.

6. Finalmente, se generó un archivo CSV tal como se solicitaba en la consigna para guardar los resultados de cada campo escaneado por formulario.

Para finalizar, se seleccionaron algunas de las imágenes generadas durante el desarrollo del código para explicar cómo funciona el mismo. Utilizaremos como ejemplo el formulario 5:

FORMULARIO B		
Nombre y apellido	PEDRO JOSE GAUCHAT	
Edad	8	
Mail	PEDRO_JOSE@GMAIL.COM	
Legajo	G-6721/0	
	Si	No
Pregunta 1	SI	
Pregunta 2		NO
Pregunta 3	SI	
Comentarios	COMENTARIO DE PEDRO	

Imagen 2: formulario 5 en su forma original

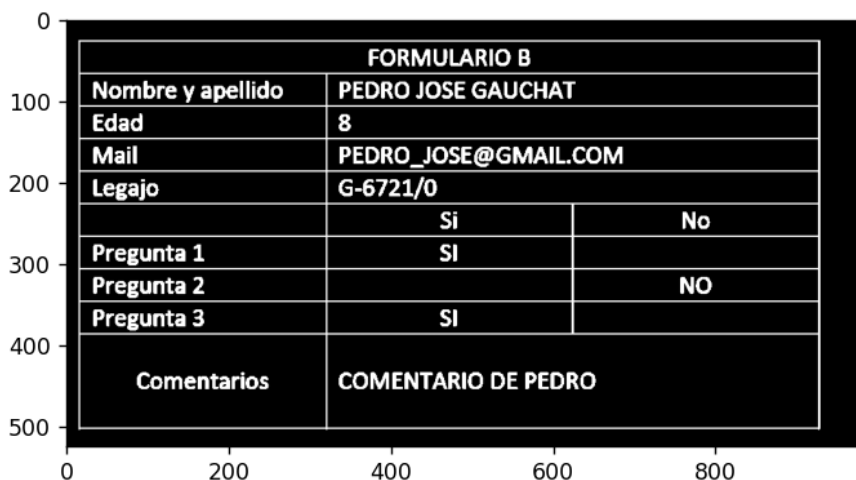


Imagen 3: Imagen binarizada.

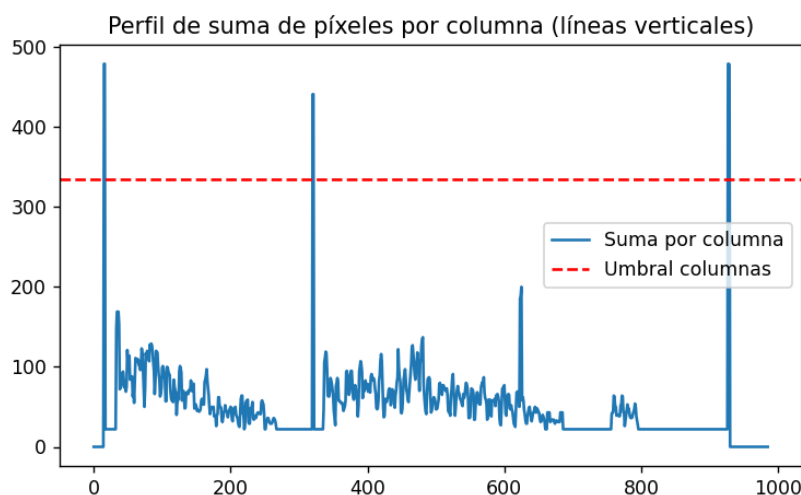


Imagen 4.

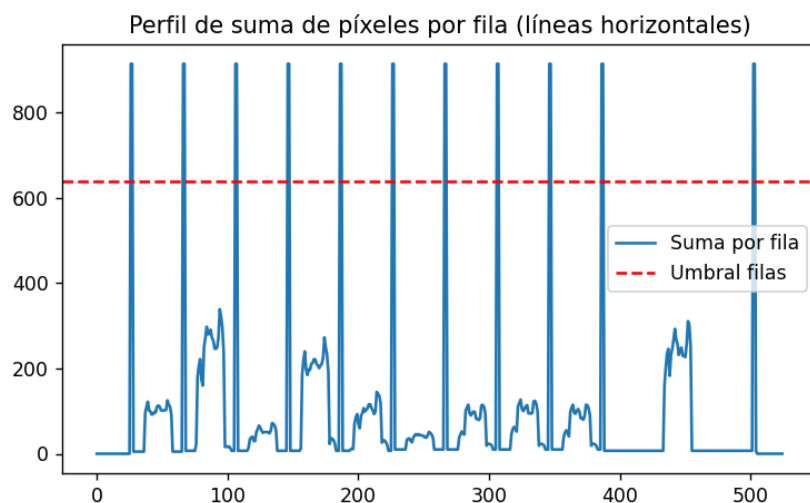


Imagen 5.

En las imágenes 4 y 5, podemos observar la suma de valores de píxeles por fila y columna, y la línea punteada roja indica el umbral que usaremos para seccionar la imagen.

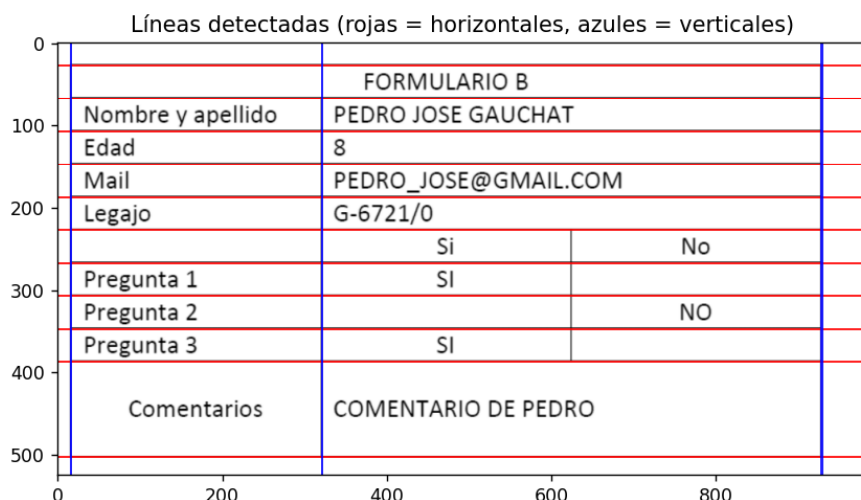


Imagen 6: División del formulario.

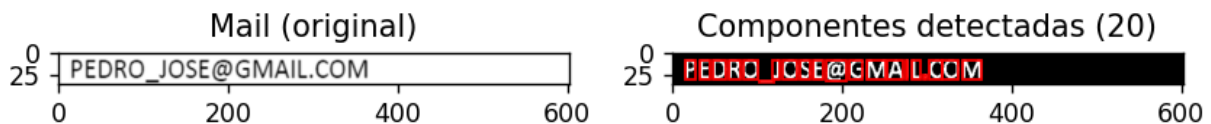
Revisión de la Segmentación del Formulario

Tipo Formulario	Nombre y apellido
FORMULARIO B	PEDRO JOSE GAUCHAT
Edad	Mail
8	PEDRO_JOSE@GMAIL.COM
Legajo	Comentarios
G-6721/0	COMENTARIO DE PEDRO
Pregunta 1_Si	Pregunta 1_NO
SI	
Pregunta 2_Si	Pregunta 2_NO
	NO
Pregunta 3_Si	Pregunta 3_NO
SI	

Imagen 7.



En la imagen 7, podemos ver las secciones de interés que tomamos del formulario. Las secciones de respuesta a las preguntas 1, 2 y 3, se ven negras por un error de visualización, pero en realidad son completamente blancas.



Por último, la imagen 8 nos ayuda a visualizar la forma en que se realizaron los recortes por carácter, para la posterior validación de los campos.

Como conclusión final del trabajo práctico, asumimos que el problema 1 buscó abordar el realce local mediante la Ecualización Local del Histograma, lo que nos sirvió para ver cómo el análisis por vecindario revela detalles ocultos en la imagen. El Problema 2, siendo un ejemplo más práctico, nos ayudó a desarrollar distintas técnicas como la suma de filas/columnas (a través de sus píxeles), analizar el contenido con Componentes Conectadas para aplicar reglas de formato (se contaron caracteres y palabras).