INFORME TRABAJO PRÁCTICO 1

PROCESAMIENTO DE IMÁGENES

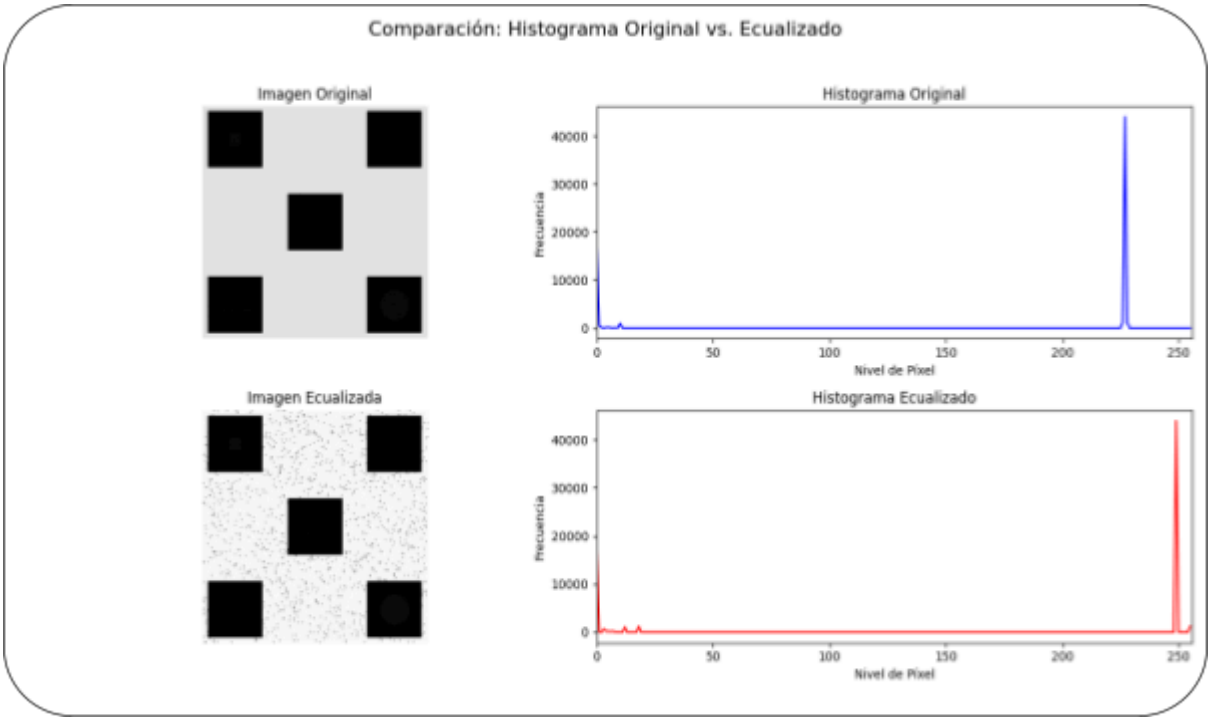
ESTEVA MATIAS
PRIETO TOBIAS

TECNICATURA UNIVERSITARIA EN INTELIGENCIA ARTIFICIAL
2025

Problema 1 | Ecualización local del histograma

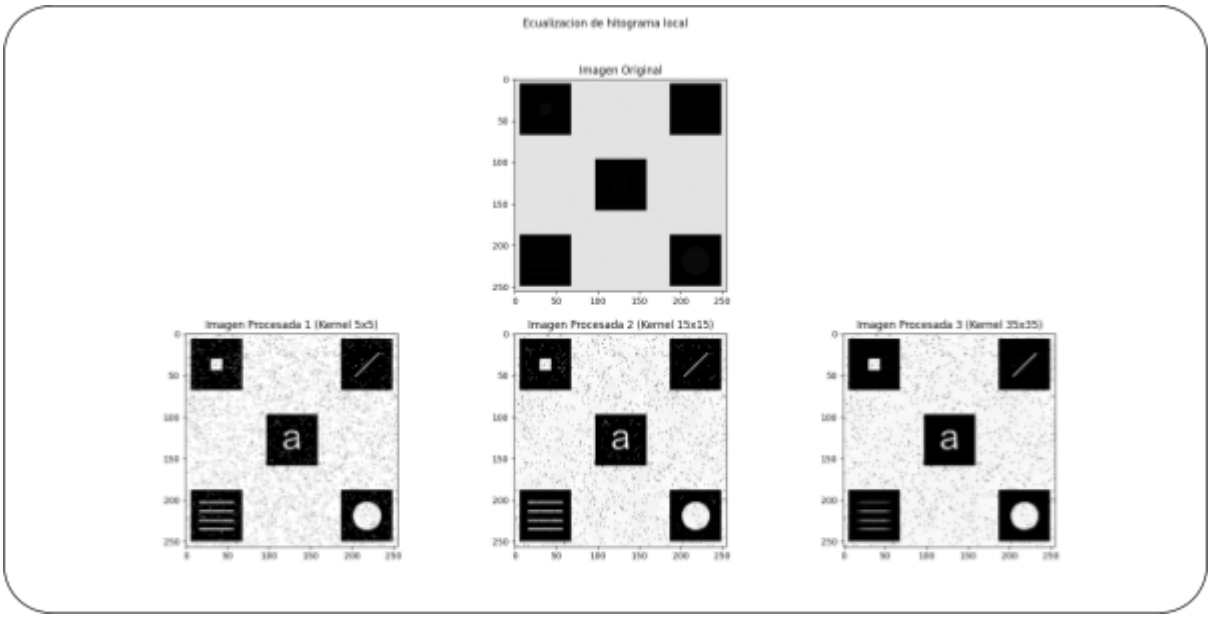
Análisis

Para la primera parte de este trabajo práctico se nos planteó implementar la técnica de ecualización del histograma, la cual se basa en aplicar una transformación a los niveles de intensidad de una imagen para distribuirlos de forma uniforme y de esa manera mejorar el contraste de la misma, cubriendo el rango completo de intensidades. Pero en particular para este caso, el problema nos plantea ejecutar este mismo algoritmo pero de forma local para de esa forma poder revelar cuales son las figuras que se encuentran por debajo de los cuadrados negros de la imagen, ya que de esa manera el ajuste del contraste se realizara a lo largo de toda la imagen pero de forma iterativa con un tamaño de kernel o ventana de actuacion especificado al momento de ejecutar la funcion. Podemos observar en la siguiente figura la distribución original de los píxeles de la imagen en comparación con su posterior ecualización, de esta forma comprobamos que el método global no es oportuno en este caso.



Entonces definimos la función **ecualización local** la cual recibe como parámetros una imagen y dos números correspondientes al tamaño de la ventana de actuación del ecualizador. Inicialmente la imagen es expandida según los píxeles situados en los bordes de la misma para de esa forma evitar problemas cuando se ejecute la función, dicha expansión está dada por la mitad del tamaño de la ventana utilizada.
Por último una vez realizada las ecualizaciones devuelve la imagen de salida, en nuestro caso decidimos representar tres figuras en un gráfico para poder

comparar el accionar de la función con distintos parámetros de dimensión de la ventana.



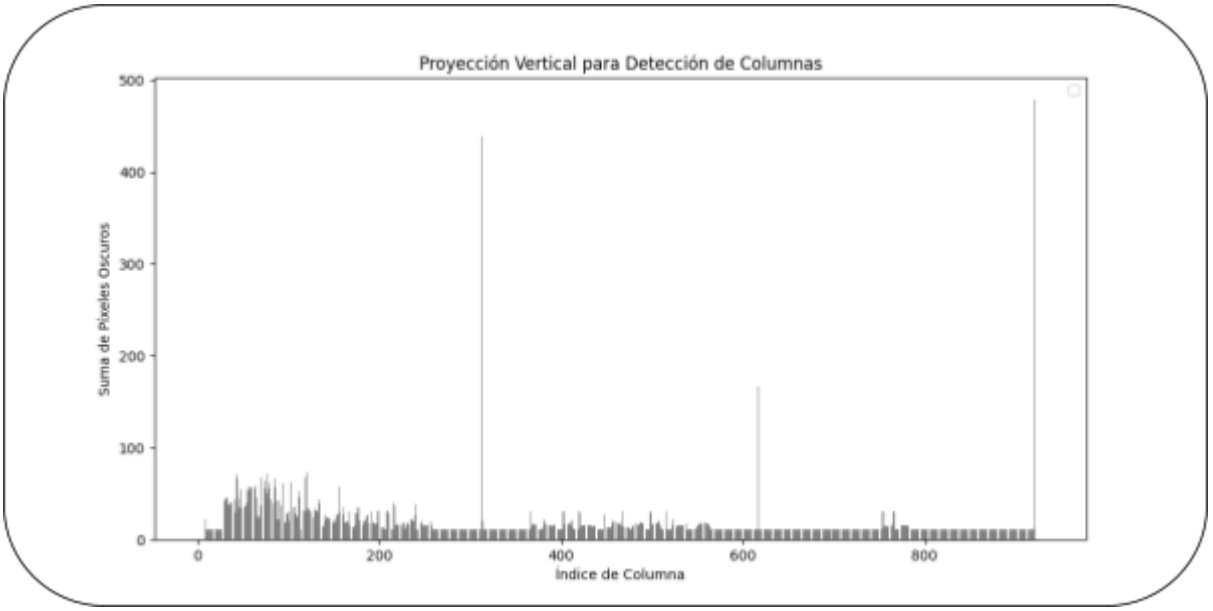
A partir de la representación podemos ver los caracteres ocultos debajo de los cuadrados, los cuales son una letra “a”, un círculo, una línea diagonal, un cuadrado y cuatro líneas horizontales. El valor óptimo para los parámetros dependerá de que tanta nitidez en las letras necesitemos a costa de sacrificar ruido en el resto de la imagen, en tamaños de ventana pequeños existe un gran contraste entre los caracteres y el cuadrado, pero se genera mucho ruido en el resto de la imagen, en cambio cuando tomamos valores más grandes el ruido del fondo y la nitidez de las formas disminuyen.

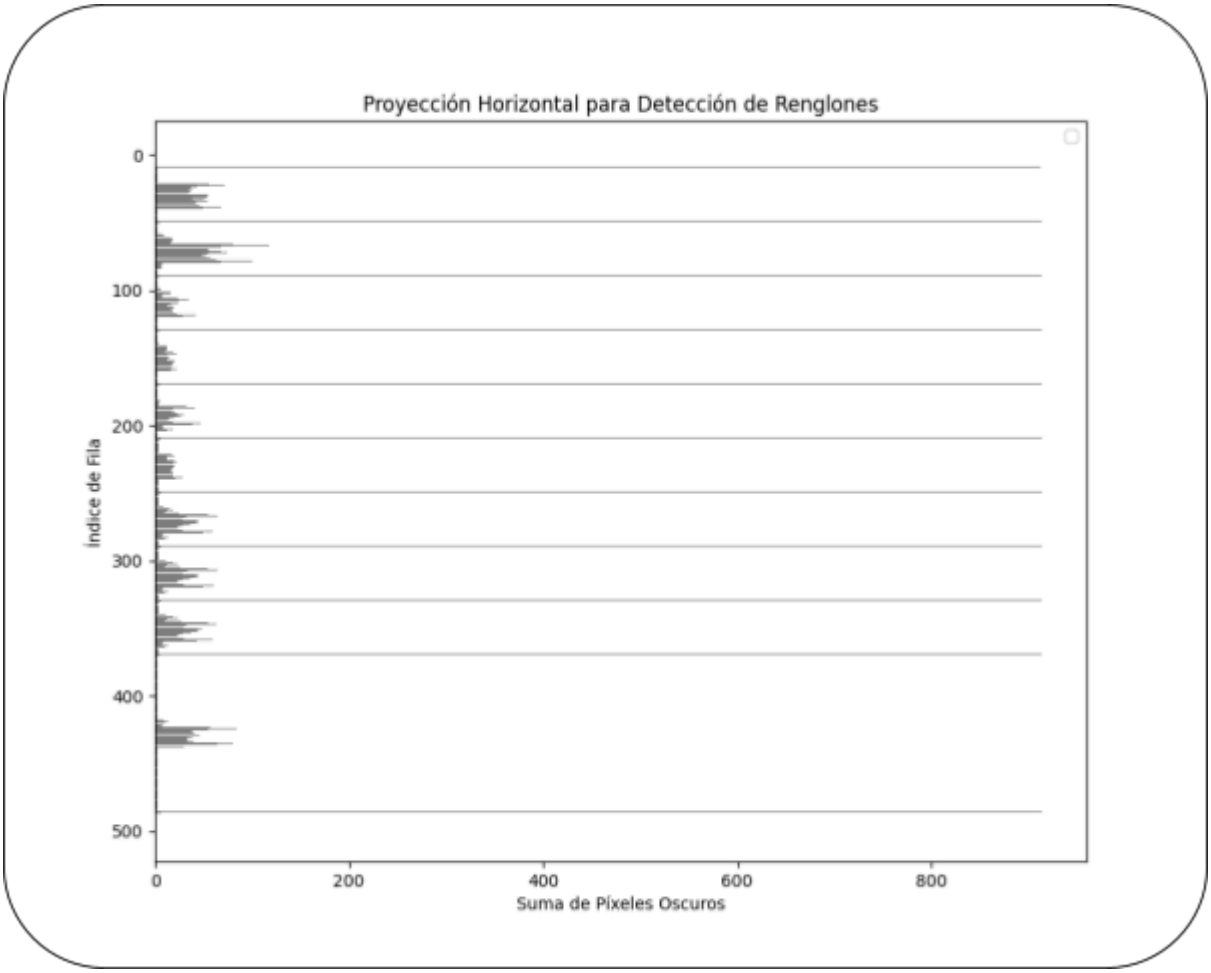
Problema 2 | Validación de formulario

Para este segundo ejercicio se nos plantea la realización de un algoritmo en Python el cual deberá recibir una serie de formularios, todos con el mismo formato de estructura, y en base a reglas establecidas en la consigna del problema, poder validar los campos determinando si cumplen o no las pautas establecidas. Además se deberá informar en pantalla el resultado de cada celda de los formularios indicando si cumplen o no la validación, también se generará una imagen con el nombre y apellido de las personas ubicandolas en un lado u otro en función de si completaron correctamente o no el formulario y por último se generará un archivo csv con la misma información previa.
Para afrontar este problema generamos muchas funciones que luego al final actuarán de forma conjunta para lograr el objetivo del ejercicio. Las dos primeras nos delimitaran donde están las columnas y los renglones para en la

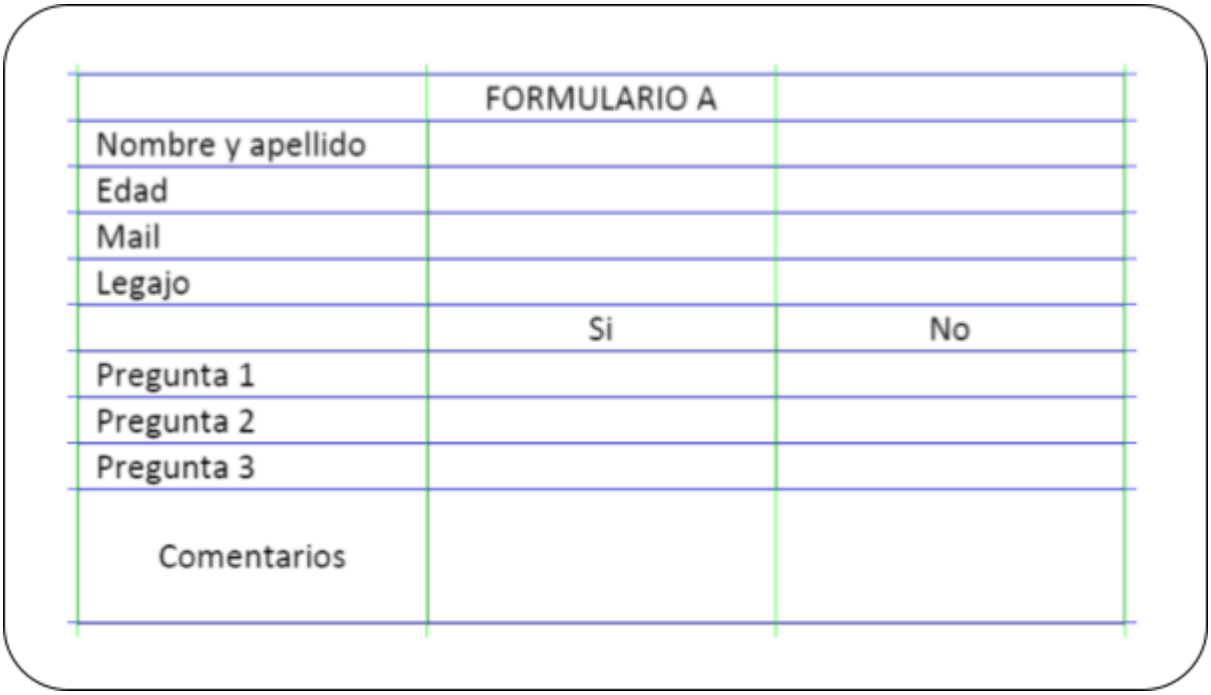
siguiente función poder efectivamente separar esos campos, para ello reciben a la imagen, la transforma a una imagen binaria por medio de un umbralado, luego detecta las filas y las columnas respectivamente donde existe una cantidad mayor de píxeles oscuros, forma pares de inicio-fin y luego almacena toda esa información en un diccionario. Todas estas acciones están contempladas dentro de la función **separar_campos** que llama a las dos previas y guarda los resultados finales de cada campo en un nuevo diccionario, es así que obtenemos todas las componentes de nuestra imagen separadas de forma automática.
Una vez que contamos con la segmentación pasamos a la etapa de análisis donde actúa la función **analizar_campo** que se encarga de indicar cuantas palabras, caracteres y si tiene espacios o no el campo en cuestión. Luego en la función **informe_formulario** es donde se realizan las comprobaciones de cada campo para conocer si cumple o no con las reglas establecidas para cada entrada, dicha información se almacena en un diccionario que será usado en la función **mostrar_formularios** para mostrar en pantalla la imagen que indicará en dos grupos cuáles son las personas que completaron correctamente o no todos los campos a rellenar, y por último la información de los resultados del formulario será guardada en un archivo csv generado al final de la ejecución del algoritmo.

En las imágenes que presentamos a continuación podemos observar cómo por medio de la detección de píxeles las funciones encargadas de detectar dichos patrones los localizan para su posterior segmentación.



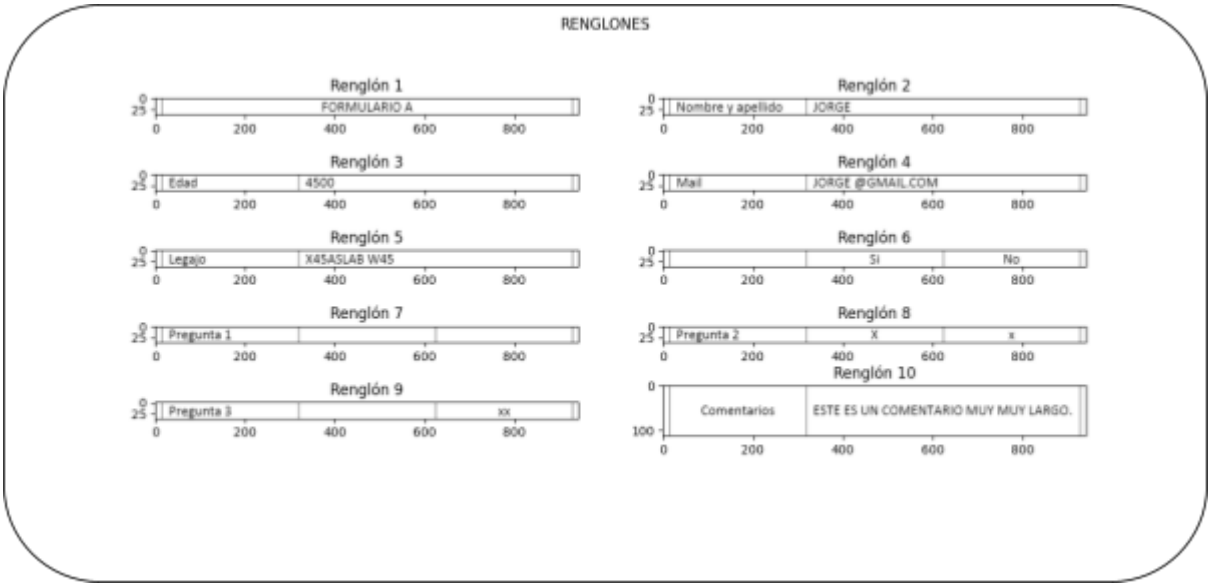


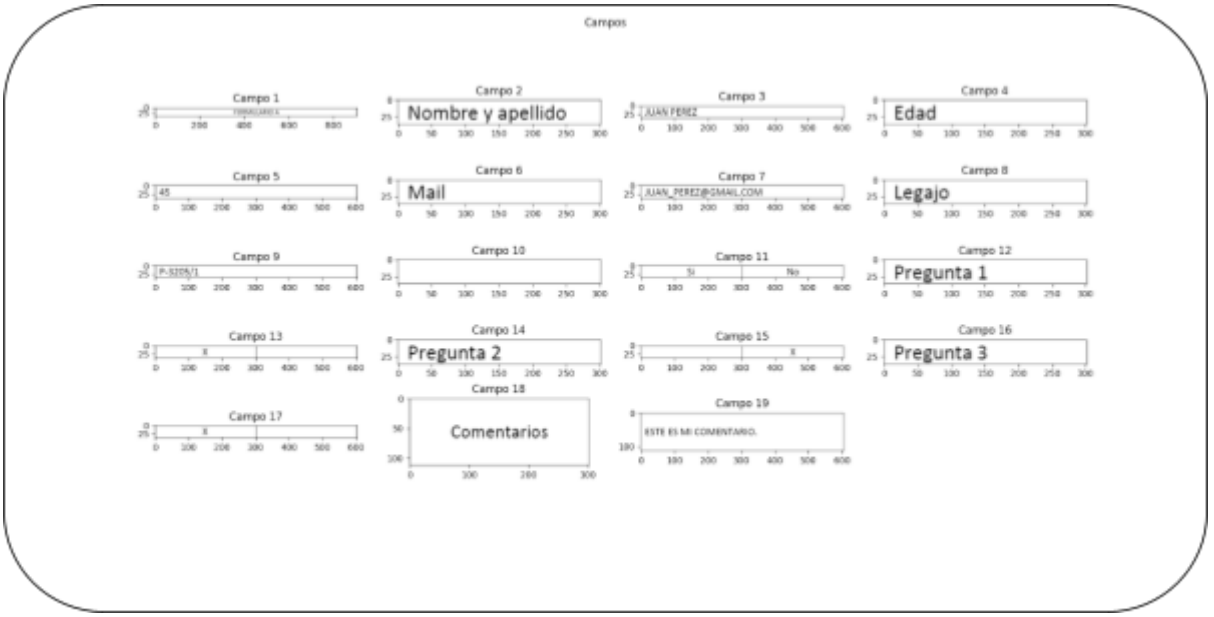
Al haber segmentado la imagen base de formulario, podemos ver como se superpone con la imagen original, indicando con verde las columnas y en azul los renglones.



	FORMULARIO A	
Nombre y apellido		
Edad		
Mail		
Legajo		
	Si	No
Pregunta 1		
Pregunta 2		
Pregunta 3		
Comentarios		

Una vez que pudimos detectar de forma automática las columnas y renglones de nuestro formulario, procederemos a separar cada fragmento del mismo para su posterior análisis particular a la hora de validar los datos. Comenzaremos con los renglones, en el siguiente gráfico podemos apreciar cómo el algoritmo divide cada renglón detectado previamente, lo cual nos servirá para luego a cada uno de ellos realizarles un nuevo segmentado, esta vez por filas, dando como resultado los campos finales que deberemos validar.





Ya con los campos segmentados de esta forma, podremos realizar el análisis para conocer cuales son aquellos que fueron ingresados de forma correcta o errónea, para ello deberemos detectar cantidad de caracteres y cantidad de

palabras, lo cual se realiza en la función **analizar_campo** y posteriormente informando los resultados de dicho análisis en la función **informe_formulario**. Finalmente llamando a la función **mostrar_formularios** agregándole como argumento en nuestro caso una lista con los nombres de los formularios mostrará en pantalla la validación respectiva para cada uno y finalmente el siguiente gráfico indicando quienes fueron los usuarios y cómo completaron sus formularios (en el formulario 4 observamos que no hay nombre porque el campo no fue completado).

Conclusiones

A modo de cierre de este trabajo podemos decir que en el caso del primer ejercicio la implementación de la **ecualización de histograma local** demostró ser una técnica superior a su contraparte global para el realce de detalles específicos. Al operar sobre vecindarios de píxeles definidos por una ventana móvil, logramos revelar información oculta en distintas zonas de la imagen, algo que habría sido imposible con un análisis global que promedia las características de toda la imagen. Este método podría extrapolarse a aplicaciones críticas como el análisis de imágenes médicas, donde realzar el contraste en pequeñas regiones de interés puede ser crucial para detectar anomalías o patrones sutiles que de otro modo pasarían desapercibidos. Además, el análisis sobre la influencia del tamaño de la ventana nos permitió comprender la importancia de ajustar este parámetro para obtener resultados óptimos según las características de la imagen.

En cuanto al segundo ejercicio, el desarrollo de un sistema de **validación automática de formularios** nos enfrentó a un desafío que, aunque inicialmente parecía una tarea de automatización precisa y detallada, reveló su enorme potencial al considerarlo a gran escala. La implementación requirió segmentar la imagen para aislar cada campo, utilizando técnicas como la detección de líneas horizontales y verticales, para luego aplicar reglas de validación específicas sobre el contenido extraído. Si pensamos en organizaciones que procesan cientos o miles de formularios diariamente, un sistema de este tipo no solo minimiza drásticamente el error humano, sino que también optimiza los tiempos y recursos, permitiendo delegar una tarea manual y repetitiva a un proceso automatizado y eficiente.

En resumen, ambos problemas resueltos nos han brindado una valiosa perspectiva sobre cómo las técnicas de procesamiento de imágenes pueden solucionar problemas del mundo real, desde el análisis detallado para la extracción de información hasta la automatización de procesos a gran escala.