



Universidad Nacional de Rosario

Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Tecnatura

Universitaria en Inteligencia Artificial

Procesamiento de Imágenes I - IA 4.4

TRABAJO PRÁCTICO N°1 - Año 2024 - 2° Semestre

Alumnos:

Antuña, Franco A-4637/1

Gallardo, Jonatan G-5970/6

Orazi, Roberto O-1815/5

Problema 1 - Ecualización local de histograma.....	2
Descripción del problema.....	3
Implementación.....	3
Ventana (3x3) – Contraste Máximo, Ruido Aumentado.....	4
Ventana (15x15) – Balance entre Detalles y Ruido.....	4
Ventana (3x50) y (50x3) – Realce Direccional del Contraste.....	5
Ventana (25x25) – Buen Detalle con Reducción Moderada de Ruido.....	6
Ventana (250x250) – Suavizado Global, Pérdida de Detalles Finos.....	7
Conclusión final.....	7
Problema 2 - Corrección de múltiple choice.....	8
Inciso A:.....	9
Carga de Imágenes y Preprocesamiento.....	9
Segmentación de Imágenes.....	9
Detección de Líneas.....	10
Recorte de Preguntas.....	10
Problemas Encontrados.....	11
Problema de Detección Duplicada:.....	11
Preguntas Vacías:.....	11
Soluciones Implementadas.....	12
Evaluación de Respuestas.....	12
Resultados.....	13
Conclusiones.....	14
Problema 2 b - Validación de datos del encabezado.....	15
Descripción del problema.....	15
Descripción del problema.....	16
Implementación.....	16
Conclusión final.....	17

Problema 1 - Ecualización local de histograma

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.

Descripción del problema

El proceso de ecualización local consiste en definir una ventana cuadrada o rectangular ($M \times N$) que se centra en un píxel y, dentro de esa ventana, calcular el histograma local. Luego, utilizando la transformación obtenida, ajustamos el nivel de intensidad del píxel central. Posteriormente, la ventana mueve un píxel a la vez en todas las direcciones, repitiendo el proceso hasta recorrer toda la imagen. Esta técnica tiene la ventaja de conservar la información de cada zona específica, algo que la ecualización global no logra al considerar únicamente la imagen en su conjunto.

El ejercicio propuesto requiere el desarrollo de una función en Python que implemente la ecualización local del histograma. La función debe recibir como parámetros la imagen en escala de grises y el tamaño de la ventana ($M \times N$), que define el área de influencia para cada transformación. La tarea principal consiste en procesar una imagen dada utilizando diferentes tamaños de ventana y observar los efectos que cada uno tiene sobre los detalles ocultos en distintas zonas de la imagen.

Implementación

En este problema empleamos un algoritmo de ecualización local que toma como entrada una imagen en escala de grises y un tamaño de ventana variable. El propósito es evaluar cómo afecta el tamaño de la ventana a los resultados visuales, permitiendo identificar detalles ocultos y el impacto del ruido.

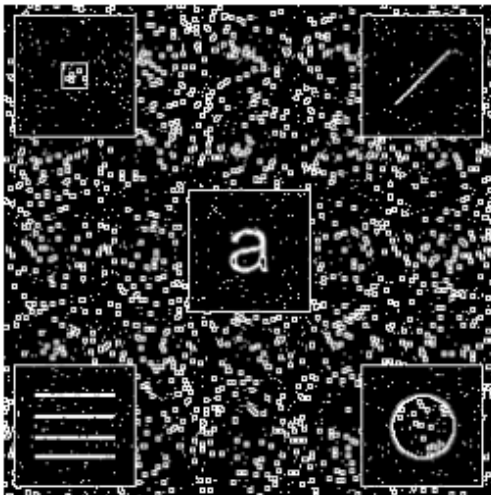
El código que diseñamos implementa una función personalizada de ecualización local del histograma que, como parámetros, recibe una imagen y un tamaño de ventana de procesamiento. La función recorre la imagen utilizando ventanas móviles y realiza el ajuste de contraste mediante la transformación del histograma en cada ventana. Se aplicaron varios tamaños de ventana para observar el impacto en los resultados.

Las ventanas que decidimos utilizar en el análisis fueron:

- (3, 3): Ventana pequeña y cuadrada.
- (3, 50): Ventana alargada horizontalmente.
- (50, 3): Ventana alargada verticalmente.
- (15, 15): Ventana mediana y cuadrada.
- (25, 25): Ventana cuadrada más grande.
- (250, 250): Ventana muy grande, cubriendo casi toda la imagen.

Ventana (3x3) – Contraste Máximo, Ruido Aumentado

Ventana (3, 3)



Vemos una gran cantidad de ruido en toda la imagen, especialmente en zonas uniformes. Este ruido se ve amplificado por el pequeño tamaño de la ventana.

Objetos ocultos (como patrones internos) comienzan a mostrarse, pero la gran cantidad de ruido visual dificulta su identificación y, sumado a esto, los objetos ocultos presentan ruido internamente.

Se resaltan bordes muy pequeños y diferencias de intensidad local. Sin embargo, los detalles finos tienden a mezclarse con el ruido, lo que puede saturar la percepción.

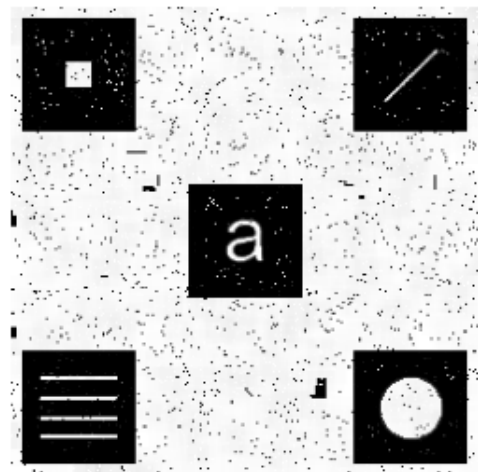
Conclusión: Este tamaño es útil para revelar microdetalles o bordes ocultos, pero también amplifica el ruido excesivamente, lo que perjudica la calidad visual en este caso.

Ventana (15x15) – Balance entre Detalles y Ruido

La ecualización aún resalta bordes y texturas ocultas en las figuras, pero con menos ruido que la ventana 3x3. Sin embargo, se perciben zonas de ruido muy marcado.

Los objetos se hacen visibles, y percibimos mejor la definición de su interior. Los bordes tienen una definición aceptable, lo cual es muy evidente en las líneas paralelas.

Ventana (15, 15)

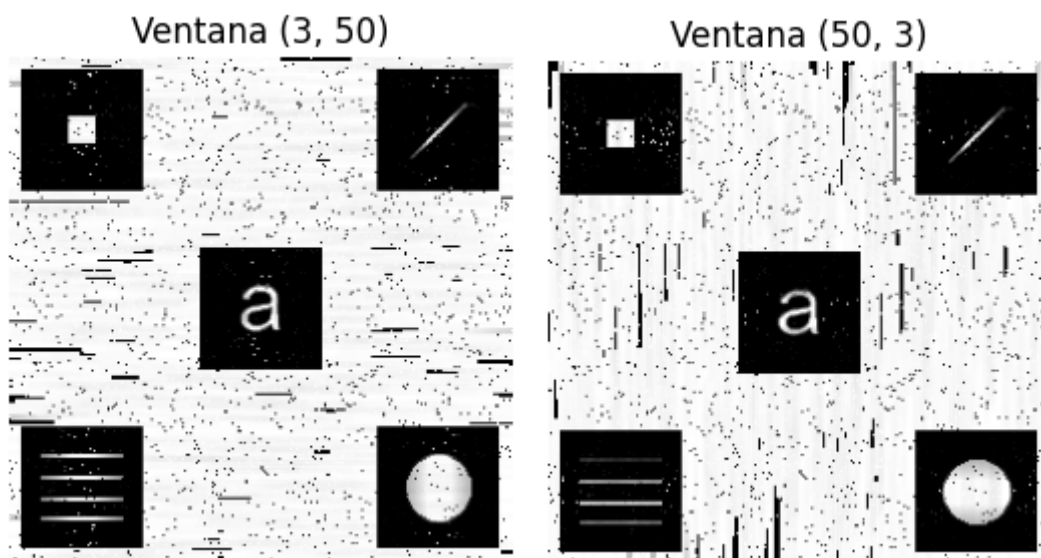


Sin embargo, es importante considerar que se percibe cierto nivel de ruido alrededor de los objetos.

Algunas figuras que antes parecían uniformemente negras comienzan a mostrar patrones internos que no eran perceptibles en la imagen original.

Conclusión: Este tamaño logra un equilibrio aceptable entre detalles y ruido, destacando tanto los bordes como las texturas internas, con menor nivel de ruido comparado con la ventana 3x3.

Ventana (3x50) y (50x3) – Realce Direccional del Contraste

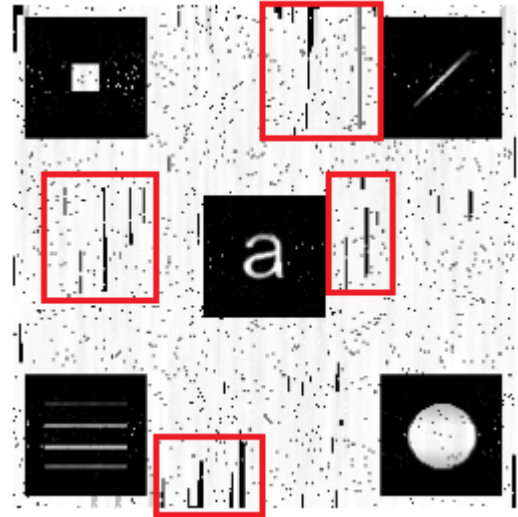


Con la ventana (3x50), se amplifican bordes y texturas en la dirección horizontal, mientras que los bordes verticales permanecen menos visibles.

Con la ventana (50x3), se resalta el contraste en la dirección vertical, haciendo que patrones alineados en esa dirección sean más evidentes.

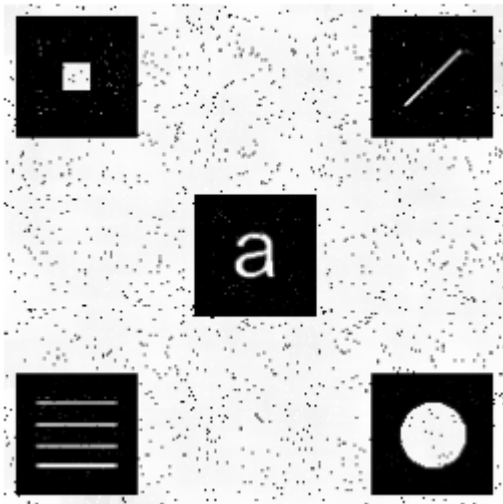
Estas ventanas también introducen cierto ruido, pero de manera más “estructurada”, con un desplazamiento muy marcado por la dirección de la misma.

Conclusión: Estos tamaños de ventana son útiles para resaltar patrones direccionales ocultos, especialmente en figuras que tienen bordes o líneas bien definidas. Sin embargo, en este caso agrega ruido direccional marcado.



Ventana (25x25) – Buen Detalle con Reducción Moderada de Ruido

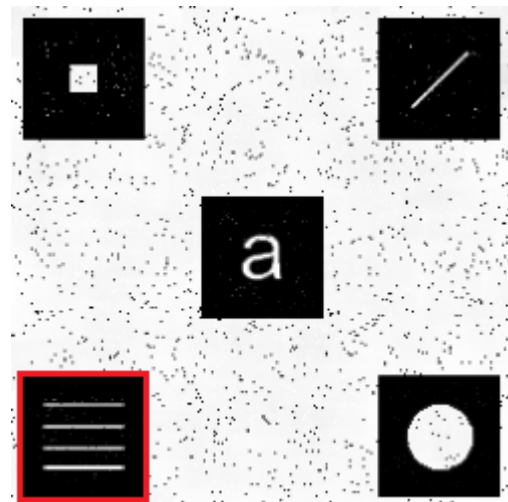
Ventana (25, 25)



Las texturas internas de las figuras y los bordes se resaltan adecuadamente, sin introducir tanto ruido como en las ventanas más pequeñas.

Patrones finos (como las líneas paralelas) se vuelven más visibles, y las zonas uniformes muestran cierta textura sin estar sobresaturadas por ruido.

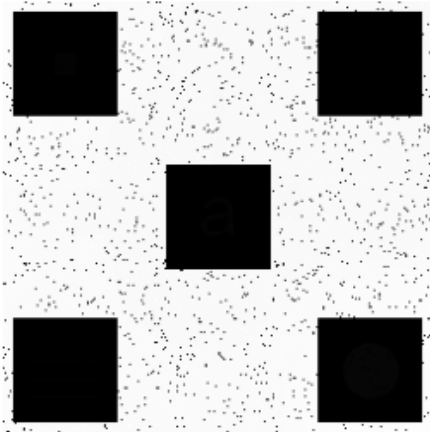
A pesar de eso, notamos una falta de definición en bordes (comparado con el caso 15x15), esto es especialmente notable en las líneas paralelas.



Conclusión: Esta ventana ofrece un buen balance, resalta detalles importantes sin amplificar demasiado el ruido, haciendo que sea una opción útil para imágenes con texturas y patrones medianamente complejos.

Ventana (250x250) – Suavizado Global, Pérdida de Detalles Finos

Ventana (250, 250)



La imagen procesada con esta ventana se ve más suave y uniforme, pero los detalles finos se pierden por completo.

Algunas diferencias de intensidad más grandes, como los bordes de las figuras geométricas, aún son visibles, pero detalles pequeños o texturas internas desaparecen.

Se introduce menos ruido, lo que hace que la imagen parezca más limpia, aunque con menos información revelada. Sin embargo, no se revelan los objetos ocultos,

debido a que la ventana tiene un tamaño excesivo.

Conclusión: Este tamaño de ventana es adecuado si se desea suavizar la imagen y reducir ruido, pero no es ideal para resaltar microdetalles o texturas internas. Los objetos pequeños ocultos no emergen con este tamaño.

Conclusión final

Los resultados del análisis muestran que el tamaño de la ventana tiene un impacto significativo en la calidad del contraste y el nivel de ruido en la imagen procesada, eso influye directamente en los objetos ocultos en la imagen. Las ventanas pequeñas (como 3x3) permiten resaltar microdetalles y bordes ocultos, pero incrementan notablemente el ruido. Por otro lado, las ventanas medianas (15x15 y 25x25) logran un equilibrio adecuado entre el realce de detalles y el control del ruido, por lo que son una opción versátil. Las ventanas alargadas (como 3x50 y 50x3) son útiles para resaltar bordes en direcciones específicas, aunque limitan la visibilidad en la dirección opuesta y el ruido tiende a formar líneas direccionales. Finalmente, las ventanas grandes (por ejemplo, 250x250) suavizan la imagen y reducen el ruido, pero a costa de eliminar detalles finos.

La elección del tamaño de ventana depende del objetivo del procesamiento. Si buscamos resaltar detalles finos u objetos ocultos, las ventanas pequeñas o medianas son adecuadas, aunque es necesario manejar el ruido que pueda generarse. En casos donde se prefiera una imagen más suave y con menos ruido, las ventanas más grandes serán la opción ideal, aunque con la limitación de perder detalles pequeños u ocultos.

Problema 2 - Corrección de múltiple choice

En la figura 2 se muestra el esquema de un examen múltiple choice de 10 preguntas con cuatro opciones para cada una de ellas (A, B, C, D). La plantilla también tiene un encabezado con tres campos para completar datos personales (Name, Date y Class).

Name: JUAN PEREZ Date: 08/08/24 Class: 1

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

Figura 2: Esquema del examen.

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:

Inciso 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.

Por ejemplo:
Pregunta 1: OK
Pregunta 2: MAL
Pregunta 3: OK
...
Pregunta 10: OK

Documentamos el proceso de desarrollo de un algoritmo para detectar y evaluar las respuestas en exámenes escaneados utilizando técnicas de procesamiento de imágenes con OpenCV y Python. A continuación, se describen los pasos realizados, los contratiempos encontrados y las soluciones implementadas.

Carga de Imágenes y Preprocesamiento

El algoritmo comienza cargando las imágenes de los exámenes y preprocesándolas mediante binarización. Utilizamos la función `cv2.threshold` para invertir los colores de la imagen y facilitar la detección de contornos.

Segmentación de Imágenes

Después de preprocesar las imágenes, se detectan los contornos y se segmentan las imágenes en mitades, extrayendo las partes que contienen las preguntas. Este proceso se realiza dividiendo la imagen en dos mitades y luego en cinco segmentos horizontales por mitad, cada uno correspondiente a una pregunta.

Examen 1 - Pregunta 1

The gaseous layers of the atmosphere are held to Earth's surface by B.

A their weight
B gravity
C the sun
D none of the above

Examen 1 - Pregunta 2

78% of the Earth's atmosphere is made up of A.

A nitrogen
B oxygen
C carbon dioxide
D water vapor

Examen 1 - Pregunta 3

The layer of the atmosphere we live in is called the B.

A stratosphere.
B troposphere.
C mesosphere.
D exosphere.

Examen 1 - Pregunta 4

Most life in the ocean is found D.

A throughout all its waters.
B deep down in the depths.
C far from shore.
D on the surface and closer to shore.

Examen 1 - Pregunta 5

A biomes location on Earth depends upon D.

A climate
B amount of rainfall
C temperature
D all of the above

Examen 1 - Pregunta 6

The Earth's system that involves all our air is called the C.

A geosphere
B hydrosphere
C atmosphere
D biosphere

Examen 1 - Pregunta 7

The Earth's system that involves all our water is called the B.

A geosphere
B hydrosphere
C atmosphere
D biosphere

Examen 1 - Pregunta 8

The Earth's system that involves all our rock is called the A.

A geosphere
B hydrosphere
C atmosphere
D biosphere

Examen 1 - Pregunta 9

The Earth's system that involves all living things is called D.

A geosphere
B hydrosphere
C atmosphere
D biosphere

Examen 1 - Pregunta 10

97% of Earth's water is found in B.

A lakes
B the ocean
C our underground aquifers
D the clouds

Por algún motivo aún desconocido, la segmentación invierte el orden de las columnas (en otras palabras, el lado derecho es, en realidad, el lado izquierdo), esto implica que alteramos el orden de las respuestas correctas para lograr que coincidan con el orden que devuelve el algoritmo.

Detección de Líneas

En el paso siguiente, se aplica la Transformada de Hough para detectar líneas en cada segmento de pregunta. Este es un paso crítico para determinar la ubicación de las respuestas marcadas por los estudiantes.

Recorte de Preguntas

Una vez detectadas las líneas, se recortan las imágenes de las preguntas basándose en la posición de estas líneas. Sin embargo, se presentaron algunos problemas durante esta etapa.

Recorte Examen 1 - Pregunta 1



Problemas Encontrados

Problema de Detección Duplicada:

En algunas imágenes (en particular el examen 3 y 4), el algoritmo detecta dos líneas en lugar de una para una misma pregunta. Esto provoca que el algoritmo intente procesar líneas que no deberían ser consideradas, rompiendo la ejecución.

```
Pregunta 2: MAL
12
12
D
Pregunta 3: MAL
Sin respuesta
Pregunta 4: MAL
7
Traceback (most recent call last):
  File "c:\Users\Franco-SEC\Desktop\TUIA\TUIA\PDI\TP1\tp1-pdi\p2-a.py", line 183, in <module>
    letra_marcada = detectar_respuesta(renglon)
  File "c:\Users\Franco-SEC\Desktop\TUIA\TUIA\PDI\TP1\tp1-pdi\p2-a.py", line 155, in detectar_respuesta
    if contar_segmentos(binaria) == 1:
  File "c:\Users\Franco-SEC\Desktop\TUIA\TUIA\PDI\TP1\tp1-pdi\p2-a.py", line 174, in contar_segmentos
    for y in range(0, altura, altura // 10): # Dividir la imagen en 10 secciones horizontales
ValueError: range() arg 3 must not be zero
```

Utilizaremos esta ejecución como ejemplo, el algoritmo en la función “detectar_lineas” debería devolver siempre renglones con 12 píxeles de altura (esto definido a partir de lo que analizamos de la imagen), sin embargo, en algunos casos devuelve 2 grupos de 7 píxeles, como es en el caso de la pregunta 9 (considerar que el algoritmo tiene invertidas las columnas) del examen 3.

Se intentó variar el umbral de detección, los parámetros de las líneas, pero no logramos un resultado que sea viable para todos los casos.

Preguntas Vacías:

En el examen 5, la pregunta 9 fue detectada como vacía, aunque debería haber contenido. Esto se debe a que el algoritmo no detecta correctamente los segmentos marcados. Está relacionado con el error anterior.

Soluciones Implementadas

Para abordar los problemas de detección duplicada de líneas, se implementó un filtrado de líneas para evitar considerar líneas que están demasiado cerca unas de otras. Esto ayuda a consolidar las líneas y reducir la complejidad en el procesamiento.

Con esta solución logramos generalizar más la detección de líneas, aunque sin lograr una funcionalidad plena.

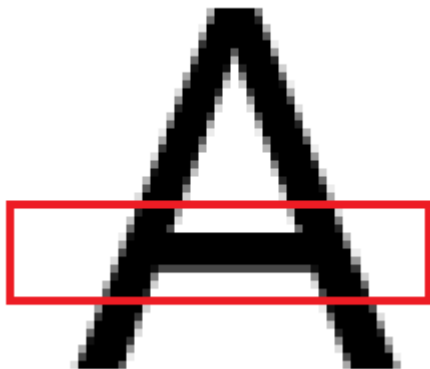
Evaluación de Respuestas

Para la evaluación, se implementaron varias opciones intentando generalizar, finalmente nos decidimos por una función que determina la respuesta en el examen a partir de varias validaciones.

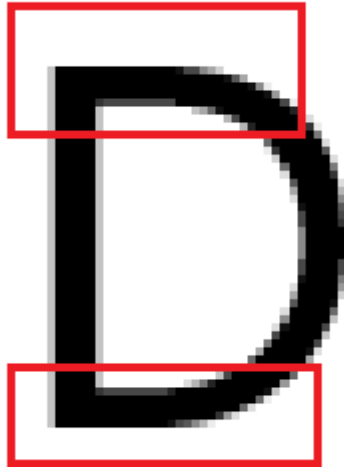
Primeramente, se pre procesa la imagen mediante una binarización, de esta forma eliminamos los grises de la ecuación.

En segundo lugar, analizamos si la imagen contiene píxeles blancos, en caso de que no existan, entonces la respuesta está en blanco. Luego se ejecuta una función que determina la cantidad de segmentos horizontales (con un mínimo de 4 píxeles de largo); esto nos permite detectar 3 casos.

- Si existe 1 solo segmento horizontal, entonces es una A



- Si existen 2 segmentos horizontales, entonces es una D



- Si existen 3 segmentos horizontales, entonces es una B



- Si ninguno de los casos es correcto, entonces se evalúa si hay contornos cerrados, en caso de no haberlos, por descarte obtenemos una C
- En caso de una detención inesperada de algún contorno cerrado, devolvemos "Error"

Finalmente, se evalúan las respuestas marcadas contra las respuestas correctas.

Resultados

El algoritmo ha funcionado correctamente para el examen 2, pero ha presentado problemas en el procesamiento de los exámenes 3 y 4 debido a la detección de líneas, y en el examen 5, la pregunta 9 fue identificada como vacía. Estas inconsistencias requieren revisiones adicionales en la detección de líneas y en el procesamiento de imágenes para asegurar que todas las preguntas sean evaluadas correctamente.

Conclusiones

El desarrollo de este algoritmo ha demostrado ser un proceso desafiante pero educativo. Los contratiempos encontrados ofrecen una oportunidad para mejorar la robustez del algoritmo. A medida que se realizan ajustes y mejoras, el sistema se volverá más preciso en la detección y evaluación de respuestas en exámenes escaneados.

Problema 2 b - Validación de datos del encabezado

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.

Por ejemplo:

Name: OK

Date: MAL

Class: MAL

Asuma que todos los campos ocupan un solo renglón y que se utilizan caracteres alfanuméricos y barra inclinada “ / ”.

En la figura 3a se muestra un ejemplo donde los campos del formulario están todos correctamente cargados, mientras que en la figura 3b se muestra otro ejemplo donde todos los campos están cargados de forma incorrecta.

Name: JUAN PEREZ Date: 08/08/24 Class: 1

Figura 3a: Todos los campos cargados correctamente.

Name: JUAN Date: 08/08 Class:

Figura 3b: Todos los campos cargados incorrectamente.

Descripción del problema

El objetivo de este problema es validar los datos de un encabezado presente en una imagen binaria. La imagen representa un formulario con tres campos: "Name", "Date", y "Class". La validación debe realizarse en base a las siguientes restricciones:

- Name: debe contener al menos dos palabras y no más de 25 caracteres.
- Date: debe tener exactamente 8 caracteres en un formato específico (dd/mm/aa).
- Clases: debe consistir en un solo carácter.

Para resolver este problema, se ha desarrollado una serie de funciones en Python utilizando la librería OpenCV, con el objetivo de segmentar la imagen, extraer los datos de cada campo, y posteriormente validarlos de acuerdo a los criterios mencionados.

Implementación

El objetivo de esta implementación es realizar la validación de los datos del encabezado de una imagen binaria, que representa un formulario con tres campos: "Name", "Date", y "Class". Para lograrlo, desarrollamos una serie de funciones en Python utilizando la librería OpenCV, que nos permite procesar imágenes y extraer la información relevante de cada campo.

Preprocesamiento de la imagen: La función `preprocess_image()` realiza una binarización de la imagen para facilitar la detección de texto. Se utiliza un umbral estático para convertir la imagen en blanco y negro, lo que permite identificar los caracteres de forma más clara.

Eliminación de las etiquetas de texto: La función `remove_text_labels()` elimina el texto de las etiquetas ("Name:", "Date:", "Class:") en cada región del encabezado. Utiliza operaciones morfológicas para detectar y eliminar los caracteres, asegurando que solo queden los datos introducidos por el usuario.

Segmentación del encabezado: La función `segment_header()` divide el encabezado en tres regiones distintas para los campos "Name", "Date" y "Class". Calcula la posición de cada campo basándose en la proporción de la imagen y elimina los subrayados para obtener una región limpia.

Extracción de los valores: La función `extract_field_value()` convierte la región segmentada en una cadena de texto mediante una representación binaria, eliminando caracteres especiales y espacios innecesarios. Esto permite obtener una representación de texto de cada campo.

Validación de los campos: La función `validate_field_values()` verifica los criterios de cada campo:

- Name: debe contener al menos dos palabras y no más de 25 caracteres.
- Date: debe tener 8 caracteres y una estructura de fecha (formato dd/mm/aa).
- Class: debe ser un solo carácter.

La función imprime el estado de cada campo indicando si es "OK" o "MAL".

Recorrido de los exámenes: Finalmente, procesamos cada imagen de examen, extraemos los valores de los campos del encabezado y validamos su contenido para informar el estado de cada campo.

Conclusión final

No logramos implementar una función que valide correctamente los datos del encabezado. Si bien pudimos segmentar la imagen en tres partes y visualizar los datos en cada campo, no conseguimos desarrollar una función efectiva para la validación. Esto deja pendiente el desarrollo de un enfoque que permita realizar la validación adecuada de los campos "Name", "Date" y "Class".

Aca podemos ver las 3 segmentaciones por examen que pudimos lograr:

ESTEBANALVAREZ

11/07/24 1

