# Trabajo Práctico 1

## Lucía Masciangelo, Francisco Rodríguez y Barros, Julieta Texier

## 30-04-2024

## Procesamiento de Imágenes I

## Tecnicatura en Inteligencia Artificial

## Requisitos previos para correr los ejercicios:

1. Clonar el repositorio <https://github.com/franciscoryb1/PDI-TUIA-RodriguezYBarros>
2. Acceder a la carpeta: cd .\TP1
3. Tener python instalado en el entorno a utilizar. Se puede descargar desde <https://www.python.org/downloads/>
4. Instalar las librerías que serán usadas en los ejercicios:
   * OpenCV: pip install opencv-python
   * Matplotlib: pip install matplotlib
   * numpy: pip install numpy

## Ejercicio 1: Ecualización local de histograma

Este ejercicio utiliza la librería OpenCV para procesar imágenes y realizar la ecualización local del histograma y la librería matplotlib.pyplot para devolver la imagen.

Primero cargamos la imagen 'Imagen\_con\_detalles\_escondidos.tif' que se encuentra en la misma carpeta donde va a correr el script.

Creamos una función para ecualizar el histograma, a la cual le vamos a pasar la imagen y un tamaño de ventana.

Dentro de la función lo que hacemos es ir iterando sobre la imagen moviéndonos en ancho y en alto a lo largo de la imagen para obtener una ventana y en ella aplicar la ecualización local con la función cv2.equalizeHist(ventana). Una vez que se hizo la ecualización colocamos los píxeles ecualizados en la imagen de salida.

Esto lo hacemos para diferentes tamaños de ventana aplicando la función de la siguiente manera:

por ejemplo, **img\_salida1 = ecualizacion\_local\_histograma(img,10)** donde le decimos que el tamaño de la ventana es 10 y que la imagen a utilizar es img

Salida: Al correr el programa se abrirá un visualizador con la imagen original y los resultados obtenidos con diferentes valores de ventana, en la cuales se puede hacer zoom para apreciar mejor las mismas.

## Ejercicio 2: Corrección de multiple choice

Utiliza las siguientes librerías: cv2, numpy y matplotlib.

Archivos necesarios: Los 5 archivos que contienen las imágenes de los multiple choice que están en la misma carpeta donde va a correr el script.

Este ejercicio se divide en 4 ítems:

1. primero tenemos que poder dividir la imagen de los multiple choice en la cantidad de filas que tiene para poder analizar después si la respuesta es correcta o no para cada pregunta.

Para ello realizamos una función que tiene como argumento la imagen. Binarizo esa imagen para que me quede con 2 colores (blanco y negro). Saco el encabezado donde están los nombres, fecha, legajo, etc ya que para esta parte no lo necesito, es decir, recorto la imagen. Luego, detectamos las filas que tienen al menos un cero e identificamos sus índices, modifico y reordeno esos índices para poder detectar los renglones. Por último, obtengo esos renglones que los guardo en una lista, cada uno con un índice y su imagen.

Ahora que tenemos los renglones, en cada uno vamos a identificar si la respuesta fue correcta o no. Para ello primero calculamos cuantos pixeles blancos hay (iteramos sobre la lista de renglones), y después definimos si esos pixeles son menores a 200 significa que no seleccionó ninguna opción y por ende respondió mal; si la suma de los pixeles es mayor a 1100 significa que seleccionó mas de una respuesta y por ende también respondió mal; LU ACÁ FALTARÍA LA LOGICA QUE HACE PARA DECIDIR SI LE PEGÓ O NO A LA LETRA UNA VEZ QUE SE QUE TENGO UNA SOLA OPCION ELEGIDA

Para correrlo tengo que pasarle a mi función **corrección(img)** la imagen que quiero calificar.

Salida: Al correr el programa me devolverá si resolvió bien o mal cada ítem del multiple choice. Además, devuelve cuantes respuestas fueron correctas y si la persona aprobó o no, estos datos los vamos a usar en el apartado C.

1. LES DEJO ESTE TMB PARA HACER
2. Este ejercicio solo pide que analicemos los resultados de mis funciones anteriores, entonces podemos decir que:

Imagen\_1: Juan Perez aprobó con 20 respuestas correctas

Imagen\_2: Jorge desaprobó con 7 respuestas correctas

Imagen\_3: Pedro Monti desaprobó con 5 respuestas correctas

Imagen\_4: Alfredo desaprobó con 8 respuestas correctas

Imagen\_5: María Suarez no tuvo respuestas correctas, por ende desaprobó

1. El objetivo es devolver una imagen con los nombres de los que hicieron los exámenes y con alguna clasificación de aprobado o no aprobado.

Para eso primero creamos una imagen toda en blanco, y tomando la lista de imágenes que guardamos con los nombres va iterando en esa lista para ir pegando en nuestra imagen nueva un nombre abajo del otro. Al mismo tiempo va escribiendo al lado del nombre si la persona aprobó o desaprobó.

Salida: Devuelve una imagen con los nombres de los 5 exámenes y si aprobaron o no.