

Práctica 0: Manejo básico de librerías

Reconocimiento de patrones

12/02/26

1. Objetivo

Al termino de esta practica el alumno conocerá el manejo básico de librerías como numpy, matplotlib, opencv y pandas, así como la manipulación básica de imágenes, series de tiempo y dataframes.

2. Desarrollo

2.1. Requisitos previos

Para poder realizar los siguientes ejercicios es necesario tener instaladas las librerias correspondientes. Una forma sencilla de hacerlo es mediante la terminal de un editor de codigo como Visual Studio Code. En esta terminal se escriben los siguientes comandos:

```
1 pip install numpy
2 pip install pandas
3 pip install opencv-python
4 pip install matplotlib
5 pip install pillow
6 pip install librosa
```

Una vez que se tienen instaladas estas librerias es posible corroborar si se instalaron adecuadamente mediante el comando:

```
1 pip freeze
```

Al ejecutar este comando en la terminal se desplegara una lista con las librerias y las versiones de estas, viéndose algo como:

```
1 numpy==1.26.4
2 pandas==2.2.2
3 opencv-python==4.10.0.84
4 matplotlib==3.8.4
5 pillow==10.3.0
6 librosa==0.11.0
```

Una vez cumplidos estos requisitos continuamos con el siguiente desarrollo.

2.2. Ejercicios a resolver

2.2.1. Imágenes

1. Carga y despliega una imagen a color usando las librerias OpenCV, matplotlib y pillow.

2. Imprime el tipo de dato y el tamaño de la imagen.
3. Muestra la imagen en pantalla.
4. Realiza las siguientes conversiones en el espacio de color de la imagen y muestra los resultados:
 - RGB a escala de gris
 - RGB a HSV
5. Muestre el histograma de intensidades de la imagen en escala de grises.
6. Normaliza los valores de píxel entre 0 y 1.
7. Usando funciones de Python ya existentes, aplica las siguientes operaciones y despliega los resultados:
 - Reajusta el tamaño de la imagen aumentando a 5 veces su tamaño original.
 - Reajusta el tamaño de la imagen disminuyendo a 3 veces el tamaño original.
 - Rota la imagen a 45, 90 y 180 grados.

2.2.2. Series de tiempo

1. Carga el archivo .wav proporcionado.
2. Grafica la señal cargada, despliega el numero de muestras y la frecuencia de muestreo.
3. Calcula y muestra la siguiente información: valor maximo, valor minimo, y energia de la señal.
4. Extrae la información entre el segundo 1 y 2, y grafica unicamente este segmento.
5. Genera: una señal senoidal de 440 Hz que dure 2 segundos y ruido blanco de la misma duración. Combina ambas señales (señal + ruido) y visualiza las tres señales en el dominio del tiempo
6. Crea una señal que sea suma de 3 senoidales (por ejemplo: 220Hz, 440Hz y 880Hz), y calcula la FFT de la señal.
7. Grafica el espectro de frecuencias e identifica los picos que corresponden a las frecuencias originales

2.2.3. DataFrames

1. Crea un DataFrame con datos de 100 estudiantes:
 - nombre (aleatorio)
 - edad (18-25)

- calificación (0-100)
 - materias inscritas (6-10)
2. Imprime la descripción de los datos (columnas y renglones), tipo y las 10 primeras filas de los datos.
 3. Muestre estadísticas básicas como percentil, media, mínimo, máximo y desviación estándar de los datos.
 4. Filtra estudiantes con:
 - calificación >80
 - edad <23
 - materias inscritas = 8

Despliga las filas que cumplieron con estos requisitos.

5. Haz una copia del dataframe creado y reemplaza valores por faltantes intencionales (NaN) aleatoriamente de tal forma que al menos 10 filas tenga valores faltantes. Verifica que estos valores solo aparezcan en las columnas edad, calificación y materias.
6. Identifica las filas con valores faltantes y despliegalas.
7. Rellena NaN numéricos con la mediana de la columna y despliega nuevamente estas filas.