Crowd Monitoring Project

Project Setup :

=> Created virtual environment for the Project

=> install all the modules that are :

- torch
- torchvision
- opencv-python
- numpy
- pandas
- matplotlib
- pillow
- scipy
- scikit-learn
- plotly
- tqdm
- flask
- streamlit
- twilio
- requests
- pyyaml

ShanghaiTech Dataset :

=> The dataset was proposed in the paper "Single-Image Crowd Counting via Multi-Column Convolutional Neural Network" (Zhang et al., 2016).

=> It contains roughly 1,198 images in total with about 330,000 annotated head locations.

=> It is split into Part A and Part B:

- Part A: 482 images, higher crowd density (crawled from internet)
- Part B: 716 images, lower to moderate density (busy streets of Shanghai)

Further, each part is divided into training and testing subsets and each subset contain images and ground-truth folders.

- images/

  - Contains the crowd scene pictures — real images with many people.

- ground-truth/

- Contains the annotations for each image.

- Each .mat file (MATLAB file) contains the coordinates of each person's head position.

=> The folder structure looks like :

```
ShanghaiTech/
   |-Part_A/
         |-train_data/
                  |-images/
                  |-ground_truth/
         |-test_data/
                  |-images/
                  |-ground_truth/
|-Part_B/
         |-train_data/
                  |-images/
                  |-ground_truth/
         |-test_data/
                  |-images/
                  |-ground_truth/
```

Preprocessing :

- ⇨ Load Images
  - Images were loaded using OpenCV and converted from BGR -> RGB.
  - Needed because OpenCV uses **BGR**, but deep learning frameworks use **RGB**.
- ⇨ Resize Images
  - Images were resized to a uniform dimension (512×512 pixels) to ensure consistent model input.
  - Normalization and equality of input shape improves training performance.
- ⇨ Load and Scale Ground Truth Points
  - Coordinates from .mat files were read and rescaled to match resized images.
  - Without scaling, annotations would no longer match resized images.
- ⇨ Create Impulse Map
  - Blank map where each head position is marked with value 1.
  - Used to convert discrete head points into spatial density representation.

- ⇨ Generate Density Map Using Gaussian Filter
  - Gaussian blur was applied to impulse points:
  - Produces smooth density distribution indicating crowd intensity.
  - The sum of density = number of people.
- ⇨ Downsampling Ground Truth (Factor 8)
  - CSRNet outputs feature maps 1/8 of the input size due to pooling layers:
  - Input: 512 × 512
  - Output: 64 × 64
  - Multiply by 64 preserves total count because 8 X 8 = 64
- ⇨ Image Normalization :
  - Final images normalized using ImageNet statistics
  - Standardizes brightness, contrast, and dynamic range.
- ⇨ Converted Image & Density to PyTorch Tensors

Data Visualization :

=> Impulse Map

- A blank matrix of size equal to the image where all values are 0 and each head coordinate is marked as 1

=> Density Map

- Impulse map is converted into a smooth heatmap by applying a Gaussian filter

- Sum of all values in density map ~ total number of people

=> Heat Map

- Heatmap just applies a color mapping scheme over the density values.

- ❖ Image with annotated red head points that Confirms annotation correctness
- ❖ Impulse map to see Raw head locations visualized
- ❖ Gaussian smoothed density map to Ground truth label for model training
- ❖ Heatmap overlay on image to Realistic interpretation of density distribution