DEEP VISION CROWD MONITOR: AI FOR DENISTY ESTIMATION AND OVERCROWDING DETECTION

Intern Name: Ananya Soppari

Table of Contents

1. Project Description

The DeepVision Crowd Monitoring System aims to estimate crowd density from input images using deep learning. The primary objective is to convert raw crowd images into structured data suitable for training CNN-based regression models such as VGG, CSRNet, and MCNN. The system uses image data along with ground truth density annotations to learn spatial distribution of people. This milestone focuses on understanding the dataset, exploring the sample distributions, visualizing the ground-truth points, and performing full preprocessing according to mentor-specified guidelines.

2. Dataset Description (ShanghaiTech Dataset)

The ShanghaiTech Crowd Counting Dataset is a large-scale, real-world image dataset consisting of two parts—Part A and Part B. It is widely used in crowd counting research and benchmarks. It contains high-density scenes, variable lighting, different camera angles, and diverse environments.

Part A Characteristics:
- Contains highly congested scenes.
- Images collected from the internet.
- Large variation in crowd density (few people to thousands).
- Comes with .jpg images and corresponding .mat files containing head-location annotations.

Part B Characteristics:
- Outdoor street scenes with moderate crowd density.
- More stable lighting and clearer visibility.
- Includes images and matching .mat ground truth files.

Ground Truth Format (.mat files):
Each .mat file stores an array of (x,y) coordinates representing head positions in the image. These coordinates are later used to generate Gaussian-based density maps.

3. Environment Setup

The following software and libraries were used:
- Python 3.10+
- Jupyter Notebook
- NumPy, Pandas, Matplotlib, Seaborn
- OpenCV (cv2) for image processing
- SciPy (for .mat file handling)
- PyTorch (torch, torchvision)
- PIL (Pillow) for RGB image loading

The GitHub repository was cloned and all work was maintained in a personal branch as

instructed. All code files (.ipynb) and the final PDF are committed and pushed only to the intern's personal branch.

4. Data Exploration (EDA) – Part A & Part B

Exploratory Data Analysis was performed separately for Part A and Part B.

Key EDA Steps:
• Total number of training and testing images identified.
• Sample images displayed using Matplotlib.
• Ground truth head-position points visualized by overlaying coordinates on the image.
• Crowd-count distribution plotted using seaborn histograms.
• Computed min, max, and mean people count for both Part A and Part B.
• Checked image resolutions and plotted scatter distribution of widths and heights.
• Created summary statistics tables (mean, std, percentiles) for counts.

EDA results showed that Part A contains extremely dense crowds (hundreds to thousands per image), while Part B contains moderate-density street scenes. Image sizes vary significantly, requiring resizing during preprocessing.

5. Data Preprocessing

The preprocessing pipeline follows the exact instructions provided by the mentor and includes the required operations for deep learning with VGG-based architectures.

Step 1: Load Images in RGB Format
Images were loaded using PIL and converted to RGB to ensure proper channel ordering, since some libraries load in BGR.

Step 2: Resize Images to Standard Dimensions (1024×768)
Original images have inconsistent dimensions. Each image is resized to a fixed width (1024) and height (768). Ground truth head-points are scaled using the same width and height scaling ratios.

Step 3: Generate Full-Resolution Density Map
Using the scaled point annotations, a Gaussian kernel (ksize=15, σ=4) is applied to produce the density map. The density map integrates to an approximate count of the total number of heads.

Step 4: Downsample Density Map by 8× (Mentor Requirement)
The density map is downsampled by a factor of 8 to match the output stride of the VGG backbone. Example:
Full map: 768×1024 → Downsampled: 96×128.

Step 5: Multiply Downsampled Map by 64 (8×8)
Downsampling reduces total density mass. Multiplying by 64 preserves the original crowd count, as instructed.

Step 6: Normalize Images Using ImageNet Statistics (Mandatory for VGG)
The following ImageNet mean and standard deviation were applied:
mean = [0.485, 0.456, 0.406]
std = [0.229, 0.224, 0.225]
This ensures compatibility with pretrained VGG architectures.

Step 7: Convert Images and Density Maps to PyTorch Tensors
- Images transformed to tensors with shape [3, H, W]
- Density maps reshaped to [1, H/8, W/8]
- A batch is created with dimensions [batch_size, channels, height, width].

Step 8: Build PyTorch Dataset and DataLoader
A custom Dataset class loads processed images and density maps. The DataLoader creates mini-batches, shuffles data, and feeds them efficiently during training.

All mentor instructions (RGB loading, 0–1 scaling, ImageNet normalization, 8× downsampling, ×64 scaling, and tensor conversion) have been fully implemented exactly as required in both Part A and Part B preprocessing notebooks.

6.SUMMARY

In Milestone 1, the DeepVision Crowd Monitoring project successfully completed the initial phases of dataset understanding and preprocessing. Both parts of the ShanghaiTech dataset (A and B) were explored, with EDA revealing differences in crowd density, image resolutions, and scene types. Images were loaded, resized, and normalized according to VGG-based model requirements, while corresponding ground truth density maps were generated and downsampled for compatibility. A structured PyTorch dataset and DataLoader pipeline were created to efficiently feed images and density maps for future training.

This milestone ensured that all preprocessing steps—including RGB conversion, Gaussian-based density map creation, downsampling, scaling, and normalization—were implemented precisely according to mentor instructions. The processed dataset is now ready for the next phase of model training and evaluation.

7.CONCLUSION

The preprocessing and data exploration steps completed in Milestone 1 provide a solid foundation for crowd density estimation. All images and density maps have been standardized, normalized, and converted to tensors suitable for deep learning models. With a robust dataset and DataLoader pipeline in place, the project is well-prepared for the subsequent stages of model development, training, and testing.