MILESTONE 3 REPORT

AKSHARA S

TABLE OF CONTENTS

1. INTRODUCTION

After completing the data preprocessing and model training phases, the next critical stage in the **Deep Vision Crowd Monitor** system is **model testing and inference**. This phase plays a vital role in evaluating how well the trained deep learning model generalizes to unseen data and performs under real-world conditions. Unlike training, where the model learns patterns from labeled data, testing focuses on validating the model's ability to produce accurate and stable predictions without further parameter updates.

The model_test-2.ipynb notebook is specifically designed to deploy the trained crowd density estimation model and perform inference on test images. During this process, the model generates predicted density maps that represent spatial crowd distribution and produces numerical crowd count estimates. These outputs help assess the effectiveness, robustness, and practical usability of the model for real-world crowd monitoring and safety applications.

2. OBJECTIVE OF MODEL TESTING

The primary objective of the model testing module is to verify the correctness and reliability of the trained crowd density estimation model. This phase ensures that the model behaves consistently when exposed to new and unseen data, which is essential for deployment in real-time monitoring systems.

The key objectives of this module include:

- Loading a pre-trained crowd density estimation model with its learned weights
- Performing forward inference on unseen crowd images without updating model parameters

- Generating predicted density maps that capture spatial crowd concentration
- Estimating the total crowd count by aggregating density values
- Visualizing and interpreting the model's outputs to validate prediction quality

Together, these objectives help confirm whether the model is suitable for deployment in real-world crowd surveillance scenarios.

3. MODEL ARCHITECTURE OVERVIEW

The crowd estimation model used in this testing phase is based on a **fully convolutional neural network (FCN)**architecture specifically designed for **density map regression**. Unlike traditional object detection models that rely on bounding boxes, this architecture predicts continuous density values for each pixel in the image.

The model removes fully connected layers to preserve spatial information and outputs a **single-channel density map**that corresponds to the distribution of people across the image. Each pixel value in the density map represents the estimated contribution of a person at that location. The total crowd count is obtained by summing all density values in the output map, making the model particularly effective for highly dense and occluded crowd scenes.

4. ENVIRONMENT & DEPENDENCIES

To ensure consistent and reliable inference, the testing process is executed in a controlled software environment that matches the training setup. This consistency helps avoid compatibility issues and ensures stable model behavior.

The following tools and libraries are used:

- **Python 3.x** for core programming and execution
- **PyTorch** for model loading, tensor operations, and inference
- **NumPy** for numerical computations and array handling
- **OpenCV** for image loading and preprocessing
- **Matplotlib** for visualization of outputs
- **SciPy** for supporting numerical operations related to density maps

These dependencies collectively enable efficient model execution and accurate visualization of results.

5. MODEL LOADING AND INITIALIZATION

In this stage, the trained model is loaded using PyTorch's model serialization utilities. The model architecture is first instantiated, after which the saved weights from training are restored. Once loaded, the model is switched to **evaluation mode** to disable training-specific layers such as dropout and batch normalization updates.

Additionally, gradient computation is disabled during inference to improve computational efficiency and ensure deterministic output behavior. This setup ensures that the model produces consistent predictions and that no accidental parameter updates occur during testing.

6. INFERENCE PIPELINE

The inference pipeline defines the step-by-step process through which input images are transformed into crowd density predictions. First, test images are loaded from the dataset and converted into the appropriate tensor format expected by the model. Necessary preprocessing steps, such as normalization and dimension reordering, are applied to maintain compatibility with the trained network.

The processed images are then passed through the model in a forward pass, resulting in predicted density maps. The final estimated crowd count is calculated by summing all values in the density map. This approach allows the system to estimate crowd size without explicitly detecting individual people, making it effective for dense and overlapping crowd scenarios.

7. POST-PROCESSING AND VISUALIZATION

Post-processing focuses on making the model's predictions interpretable and visually meaningful. The predicted density maps are converted into visual heatmaps using color mapping techniques, where different colors represent varying levels of crowd density.

These heatmaps are displayed alongside the original input images, allowing for direct visual comparison between the observed crowd and the predicted density distribution. This visualization step helps validate whether the model correctly focuses on actual crowd regions and produces smooth, spatially coherent density estimates.

8. PERFORMANCE INTERPRETATION

Model performance in this testing phase is primarily evaluated using **qualitative analysis**. This includes examining the visual alignment between predicted density maps and actual crowd locations, assessing the smoothness and continuity of density transitions, and verifying whether the estimated crowd counts appear reasonable when compared to the visual content of the images.

Although quantitative metrics such as Mean Absolute Error (MAE) and Mean Squared Error (MSE) are typically used during validation, qualitative interpretation during inference provides valuable insight into real-world usability and prediction stability.

9. CONCLUSION

The model testing and inference module confirms that the **Deep Vision Crowd Monitor** system is capable of reliably estimating crowd density and generating meaningful crowd count predictions on unseen data. The testing results demonstrate the model’s robustness, spatial accuracy, and suitability for real-world crowd monitoring applications.

This phase marks an important step toward deployment, as it validates the effectiveness of the trained model in practical scenarios and prepares the system for integration into live surveillance and safety monitoring environments.