# MILESTONE-2: MODEL TRAINING & EVALUATION (PART A & PART B)

## Title Page

**Project Title:** Deep Vision Crowd Monitor: AI for Density Estimation and Overcrowding Detection
**Milestone-2:**Model Training & Evaluation
**Student Name: Anshla Pagdal**
**Mentor Name:** G.K.S Jyoteesh
**Date:** 6th December 2025

## Table of Contents

## 1. Introduction

Crowd counting is an important computer vision task used in surveillance, safety monitoring, and population analytics. In this milestone, we train and evaluate **CSRNet**, a deep neural network designed for density map estimation, using the **ShanghaiTech Part A** and **Part B** datasets.

Milestone-2 focuses on:

- Training CSRNet on both datasets
- Evaluating performance using MAE/RMSE
- Performing combined inference for both models

## 2. Dataset Description

### 2.1 ShanghaiTech Part A

- Contains high-density crowd images
- Highly congested scenes
- Ground truth: point annotations (mat files)
- More difficult due to extreme crowd density

### 2.2 ShanghaiTech Part B

- Low-density images taken from streets
- 400 training images, 316 testing images
- Ground truth: point locations
- Easier than Part A

For Part B, dataset splits used in training:

- **Train:** 340 images
- **Validation:** 60 images
- **Test:** 316 images

## 3. Model Architecture (CSRNet)

### 3.1 Frontend

- Based on **VGG16-BN pretrained on ImageNet**
- Uses first 33 convolutional layers

- Extracts spatial features

### 3.2 Backend

- Series of **dilated convolution layers**:
    - Dilated conv layers with dilation = 2
    - Resolve spatial resolution loss
    - Improve global receptive field

### 3.3 Output

- Generates a **density map**
- Sum of density map gives **predicted crowd count**

This architecture is widely used for crowd counting due to its strong performance in dense and sparse conditions.

## 4. Training Methodology

### 4.1 Preprocessing

- Images normalized using ImageNet mean & std
- Ground truth converted into Gaussian density maps
- Gaussian $\sigma = 5$

### 4.2 Training Settings

- Loss Function: **MSELoss (sum reduction)**
- Optimizer: **Adam**
- Learning Rate: **1e-5**
- Batch Size: **4**
- Number of Epochs:
    - Part A: Long training (Kaggle runtime)
    - Part B: 20 epochs + extended training to 30

### 4.3 Validation Metrics

I evaluate performance using:

- **MAE (Mean Absolute Error)**
- **RMSE (Root Mean Squared Error)**

Lower MAE/RMSE = better performance.

## 5. Results & Evaluation

### 5.1 Part A Results

#### Training Environment

- GPU: NVIDIA P100
- Total training time: 4h 37min
- Output files generated: 380

#### Final Evaluation

| Metric | Value |
|---|---|
| **Test MAE** | **73.24** |
| **Test RMSE** | **118.58** |

#### Interpretation

- Part A is a **high-density** dataset → higher error is expected.
- A test MAE of **73** is reasonable for CSRNet on Part A.

## 5.2 Part B Results

#### Training Summary

- Dataset: 400 train, 316 test
- Best model saved as `partB_best.pth`

#### Validation Performance

The model improved across epochs and achieved:

| Epoch | Best Validation MAE |
|---|---|
| **24** | **53.33** |

#### Test Performance

After loading best model:

| Metric | Value |
|---|---|
| **Test MAE** | **53.27** |
| **Test RMSE** | **66.71** |

### Analysis

- Part B results are significantly better than Part A, as expected.
- MAE around ~50 is good for low-density scenes.

## 6. Combined Inference (Part A + Part B)

A unified inference notebook was created to:

- Load trained Part A and Part B models
- Perform density map prediction
- Predict total count
- Visualize heatmaps

This demonstrates end-to-end functionality for both datasets.

## 7. Discussion

- **CSRNet** successfully learns density estimation for both sparse and dense crowds.
- **Training stability** improved after 10 epochs for Part B but fluctuated due to dataset variety.
- **Part A** had higher MAE because scenes are extremely congested.
- **Part B** performance (MAE ≈ 53) is strong and aligns with research papers.
- Dilated convolutions helped preserve spatial resolution during training.

## 8. Conclusion

Milestone-2 successfully:

- Trained CSRNet on both Part A & Part B
- Achieved good performance on Part B
- Built a combined inference pipeline
- Prepared training notebooks and model weights
- Evaluated models using standard metrics

This completes the model training and evaluation phase required for the project.