Real-Time Crowd Density Estimation for Public Safety

{Enhancing Surveillance with Neural Networks}

ANJANIDEVI
[904038940]

INTRODUCTION

Why Real-Time Crowd Density Estimation?

- Prevents overcrowding accidents and disasters
- Enables immediate decision-making during events
- Enhances public safety in airports, malls, stadiums
- Designed for real-time operation on edge devices (e.g., NVIDIA Jetson)

Project Objective

Main Goals

- Build a real-time crowd density prediction system
- Achieve high accuracy with minimal computational overhead
- Optimize model for real-world, edge device deployment
- Maintain performance at ~30 FPS (frames per second)

Existing Method:

CSRNet : Baseline Approach

- **Architecture:**
  - VGG-16 feature extractor
  - Dilated convolutions to expand receptive field
- **Advantages:**
  - High counting accuracy
- **Limitations:**
  - Heavy model, slow for real-time use
  - High resource consumption (GPU needed)

PROPOSED METHOD

Enhancements Over CSRNet:

•**Model Pruning:**

- Reduce deep-layer filters (512 → 256)

•**Multi-Scale Feature Fusion:**

- Combine early and late layer features

•**Real-Time Optimization:**

- Target <33ms inference time

•**Optional Extension:**

- CNN-LSTM for frame sequence consistency

SYSTEM ARCHITECTURE :

Pipeline Overview

1.Capture live video frames

2.Normalize and preprocess images

3.Feature extraction (Modified VGG-16 + Dilated Convs)

4.Generate pixel-wise density map

5.Sum density map to estimate crowd count

6.Display density heatmaps for visual interpretation

DATASET DETAILS

UCSD Crowd Counting Dataset

•2,000 grayscale video frames (238×158 resolution)

•Head annotations for ground-truth density maps

•**Crowd Range:** 11 to 46 people

•**Split:**

- 80% Training
- 10% Validation
- 10% Testing

DATA PREPROCESSING

Preparing the Dataset

•Grayscale normalization

•Gaussian smoothing of annotated head points

•Sequence formation for LSTM consistency (optional)

•Data augmentation (cropping, rotation) to improve robustness

Model Overview :

Modified CSRNet + CNN-LSTM (Optional)

•**VGG-16 Backbone:** Pruned for lightweight feature extraction

•**Dilated Convolutions:** Larger receptive field without losing resolution

•**Feature Fusion:** Early and deep features combined

•**Temporal Module:** LSTM added (optional) for better video sequence tracking

MODEL TRAINING SETUP

Training Details

- **Loss Function:** Mean Squared Error (MSE)
- **Optimizer:** Adam (Learning Rate: 0.0001)
- **Batch Size:** 8 or 16 (grid search optimized)
- **Early Stopping:** Halt after 10 epochs of no validation improvement
- **Evaluation Metrics:**
  - MAE (Mean Absolute Error)
  - RMSE (Root Mean Squared Error)

OUTPUT AND RESULTS

Performance on UCSD Test Set

•**MAE:** < 5 persons

•**RMSE:** ~6–7 persons

•**Inference Time:** ~30 FPS (≤33ms/frame)

•**Output:**

- Density heatmaps
- Frame-wise crowd counts

✅ Real-time
✅ Lightweight
✅ Highly accurate

EXAMPLE OUTPUTS :

- Visual Results
- Input frame → Predicted Density Map
- Heatmap superimposed on original frame
- Crowd size estimation matches closely with ground truth
- *(Add sample images of input and corresponding heatmaps here)*

USES CASES

Applications

•Crowd management in public transport hubs

•Monitoring entry/exit points at large events

•Emergency evacuation planning

•Smart surveillance systems

•Festival and stadium crowd control

CHALLENGES FACED

Key Challenges

•Balancing speed and accuracy on resource-constrained devices

•Handling varying crowd densities (sparse vs dense)

•Overcoming low-quality video input (motion blur, lighting)

•Efficient model compression without losing precision

FUTURE WORK

Enhancements Ahead

•Test on larger, more diverse datasets (e.g., ShanghaiTech)

•Adaptive input resolution based on scene complexity

•Develop real-time alerting and warning systems

•Explore self-supervised and semi-supervised learning approaches

CONCLUSION

Summary

- Successfully designed a real-time, edge-deployable crowd estimator.
- Modified CSRNet delivers speed and accuracy for public safety use cases.
- Future work aims to make systems even more autonomous, scalable, and robust.

THANK YOU