IJARSCT

International Journal of Advanced Research in Science, Communication and Technology

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

ISSN: 2581-9429

Volume 5, Issue 6, June 2025

Impact Factor: 7.67

Evaluate CNN Accuracy for Crowd Counting and Density Mapping

Dr. Alpana Adsul[1], Pawar Mayuri[2], Khule Sagar[3], Priyanka Lohar[4], Akash Murade[5]

[1]Assistant Professor, Dr. D. Y. Patil College of Engineering and Innovation, Pune, India

[2,3,4,5]Student, Dr. D. Y. Patil College of Engineering and Innovation, Pune, India

hod_computer@dypatilef.com, mayuripawar461@gmail.com, sagarkhule.2003gmail.com , priyankalohar461@gmail.com, akashmuradeprof@gmail.com

Abstract: *In fields including public safety, urban infrastructure planning, and large-scale event management, precise crowd counts and density estimation are essential. Because of occlusions, scale differences, and intricate spatial distributions, traditional methods frequently have drawbacks. In order to overcome these obstacles, we provide a thorough analysis and use of Convolutional Neural Network (CNN)-based techniques for crowd counting in this paper, utilizing contemporary designs. We assess the effectiveness of sophisticated models, such as attention-enhanced networks and Multi-column CNNs (MCNNs), which are intended to highlight important crowd locations and capture multi-scale properties. The resolution of generated density maps is improved by using deconvolution layers to restore spatial detail lost during downsampling. Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are two measures used to evaluate the performance of our models, which are trained and verified using benchmark datasets like ShanghaiTech (Parts A and B) and UCF CC 50. The outcomes show notable gains in accuracy and resilience in a variety of crowd situations. We also go into ethical issues, real-time optimization techniques, and existing constraints like dataset variety and environment-specific generalization. This work offers a more sophisticated viewpoint on CNN-based crowd analysis and lays the groundwork for upcoming improvements in crowd monitoring systems that are scalable, effective, and morally sound*

Keywords: Convolutional Neural Networks, Crowd Counting, Density Estimation, Multi-column CNN, Attention Mechanism, Real-time Processing.

I. INTRODUCTION

For a variety of applications, such as public safety, event planning, transportation scheduling, and urban infrastructure construction, an accurate estimate of crowd size and density is essential. In real-world situations, traditional methods for crowd counting and density estimates, like detection-based and regression-based approaches, have serious drawbacks, particularly when there are occlusions, scale variation, and high-density congestion 1. The use of deep learning techniques, especially Convolutional Neural Networks (CNNs), which provide greater capability in learning complicated, hierarchical feature representations directly from images, has been spurred by these difficulties. For crowd counting applications, CNN-based architectures—particularly those that use multi-column designs—have shown considerable gains in accuracy and resilience.

In order to efficiently model spatial heterogeneity and density fluctuation within crowd situations, multi-column CNNs (MCNNs) take advantage of different receptive fields. To further improve prediction accuracy and spatial resolution in density maps 4, recent developments have added elements including attention mechanisms, deconvolutional layers, and global density feature integration. The efficiency of CNNs in complicated and high-density situations has been greatly enhanced by these developments. The development and effectiveness of CNN-based models for density estimation and crowd counting are examined in this paper. Important evaluation metrics, such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) enchmark datasets such UCF CC 50 and ShanghaiTech Parts A and B. This study attempts to determine the advantages, disadvantages, and areas for development of existing methods by examining

DOI: 10.48175/IJARSCT-27923

model architecture, training methodologies, and dataset diversity.Notably, Zhang et al. 4 presented a groundbreaking MCNN framework that integrates several receptive field sizes to handle varying crowd densities, leading to notable accuracy gains on publicly available datasets. In order to enable the network to capture both local and holistic context, Sindagi et al. 3 devised a multi-task learning model that estimates density maps and global crowd counts simultaneously. By using a switchable CNN architecture that dynamically chooses sub-networks based on crowd density, Liu et al. 1 further increased adaptability and enhanced performance in a variety of scenarios.

Hybrid methods have also been investigated. Boominathan et al. 5 highlighted the drawbacks of non-end-to-end learning pipelines while providing competitive results by combining CNN-derived features with conventional regression models. According to Gajare et al. 6, the absence of extensive annotated datasets continues to be a major obstacle, and in order to enhance generalisation, they support data augmentation and synthetic data production. Furthermore, Dahatonde et al. 7 draw attention to the necessity of more thorough assessments in a variety of dynamic crowd settings.

Even with the noteworthy advancements, there are still significant obstacles to overcome in order to create reliable, scalable, and broadly applicable crowd counting systems. The purpose of this paper is to summarise the main conclusions and suggest possible lines of inquiry for further study in this developing field.

II. METHODOLOGY

This section outlines the systematic methodology adopted for evaluating the effectiveness of Convolutional Neural Networks (CNNs) in crowd counting and density mapping. The workflow includes data acquisition and preprocessing, feature extraction, model development, performance evaluation, and real-time system integration.

A. Data Acquisition and Preprocessing

1) Dataset Collection: *To ensure robust training and model generalization across diverse crowd scenes, two benchmark datasets were selected:*

ShanghaiTech Dataset: Comprising Part A (dense crowds, primarily from the internet) and Part B (sparser scenes from urban streets), this dataset enables evaluation under varying crowd densities.

UCF_CC_50 Dataset: A challenging dataset containing 50 images with extreme density variations (ranging from 94 to 4,543 people per image), widely used to benchmark the upper limits of crowd counting performance.

2) Data Cleaning and Augmentation***:*** *All data underwent rigorous preprocessing to ensure consistency and quality:*

Data Cleaning: Removal of corrupted, mislabeled, or low-quality images to enhance data reliability

Augmentation Techniques: To increase dataset variability and reduce overfitting:

Random cropping and resizing

Horizontal flipping and random rotations

Contrast and brightness adjustments

3) *Crowd Region Segmentation*: To isolate informative regions and eliminate irrelevant background noise, deep learning-based segmentation methods (e.g., U-Net and Mask R-CNN) were employed. This step ensured that feature extraction was concentrated on high-density regions, improving downstream model accuracy.

B. Feature Extraction and Representation

1) CNN-Based Feature Learning: *Advanced CNN architectures were employed for learning multi-scale representations from crowd images:*

Multi-Column CNN (MCNN): Utilizes multiple parallel convolutional columns with different kernel sizes to capture features at various scales, improving performance in heterogeneous crowd scenes.

Attention Mechanisms: Self-attention layers guide the model to focus on high-relevance areas, improving robustness against occlusions and background clutter.

Backbone Networks: Pre-trained networks such as VGG-16, ResNet-50, and CSRNet were fine-tuned for domain-specific feature extraction.

2) Additional Feature Engineering: *Beyond standard deep features, auxiliary information was extracted to improve contextual understanding:*
Optical Flow: Applied in video-based extensions to capture motion patterns across frames, aiding in dynamic crowd estimation.
Spatial Contextual Features: Structural relationships between localized crowd regions were encoded to support density estimation.

3) Dimensionality Reduction:To enhance computational efficiency and reduce redundancy:
Principal Component Analysis (PCA)
t-Distributed Stochastic Neighbor Embedding (t-SNE):These techniques were employed for visualizing high-dimensional features and retaining only the most discriminative attributes.

C. Model Training and Optimization

1) Transfer Learning: *Pre-trained CNN models were fine-tuned using domain-specific crowd datasets. This reduced training time and improved convergence by leveraging learned representations from large-scale datasets like ImageNet.*
2) Hyperparameter Tuning: Critical parameters were optimized to achieve optimal performance:
Search Techniques: Grid search, random search, and Bayesian optimization were used to tune learning rate, batch size, number of layers, and regularization strength.
Learning Schedulers: Adaptive schedulers (e.g., ReduceLROnPlateau and cosine annealing) were used to dynamically adjust learning rates.

3) Regularization Techniques: To mitigate overfitting and improve generalizability:
Dropout: Randomly deactivating neurons during training.
L1/L2 Regularization: Penalizing weight magnitudes to simplify the model and prevent over-complexity.
Batch Normalization: Standardizing intermediate layer outputs to accelerate training and improve stability.

D. Model Evaluation and Refinement

1) Performance Metrics: The models were evaluated using multiple metrics, including:
Mean Absolute Error (MAE): Measures the average deviation between predicted and actual counts, indicating model precision.
Root Mean Squared Error (RMSE): Provides insights into the robustness of the model by highlighting larger prediction errors.
Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM): Used to assess the quality of estimated density maps against ground truth.
2) Cross-Validation: *A K-fold cross-validation strategy (typically 5-fold) was employed to estimate generalization performance and minimize variance introduced by specific training-validation splits.*
3) Error Analysis and Model Refinement: *Systematic analysis of erroneous predictions was conducted to identify performance bottlenecks:*
Heatmap Visualizations: Used to inspect prediction focus regions.
Failure Case Categorization: Based on occlusion severity, scale variation, and lighting inconsistencies.
Failure Case Categorization: Based on occlusion severity, scale variation, and lighting inconsistencies.
Model Adjustment: Insights were used to refine layer structures, augment training data, and enhance feature selection strategies.

Model Architecture

The provided figure illustrates a **deep learning architecture for crowd counting and density estimation** from an input image. Here's a brief description of each component and the flow:

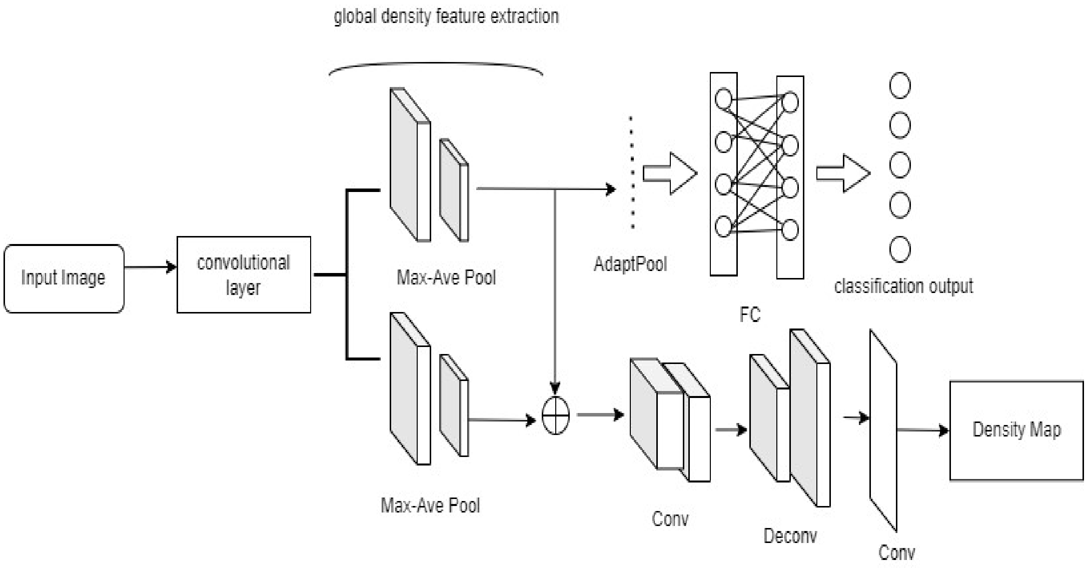


Fig. 1: Model Architecture of Crowd Counting and Density Mapping

1. Input Image: *The process begins with an image (typically of a crowd scene) as input to the model.*
2. Convolutional Layer: *The input image is passed through a **convolutional neural network (CNN)** layer to extract low-level spatial features (like edges, textures, etc.).*
3. Global Density Feature Extraction:
This branch is designed to capture **global crowd density features**:
Max-Ave Pooling: Both **max pooling** and **average pooling** operations are performed to retain robust features.
The pooled features are passed through **adaptive pooling (AdaptPool)** to ensure a fixed output size.
Then passed into a **fully connected (FC) layer** to learn global crowd density patterns.
The final output of this branch is a **classification output** which might indicate different crowd density levels (e.g., sparse, medium, dense).
4. Density Map Estimation Path
The same features are also passed down a separate branch to create a detailed **density map:**
Max-Ave Pooled features are combined (via concatenation or element-wise addition).
Then passed through **convolutional (Conv)** and **deconvolutional (Deconv)** layers to refine spatial resolution.
Final convolutional layer outputs a **density map**, representing the number of people per pixel region.
5. Output
Classification Output: Gives an overall estimation of crowd density level.
Density Map: A spatial map that indicates how dense the crowd is at each location in the image. Integrating over this map gives the total estimated crowd count.

Dataset

ShanghaiTech Crowd Counting Dataset – Detailed Description

The One of the most important and often used benchmark datasets in computer vision, especially for the tasks of density estimation and crowd counting, is the ShanghaiTech Crowd Counting Dataset. Zhang et al. first presented it at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) in 2016 with their groundbreaking paper, "Single-Image Crowd Counting via Multi-Column Convolutional Neural Network." In order to overcome the considerable difficulties in precisely measuring crowd density in photos with different degrees of crowd congestion, lighting, scale, and occlusions, the dataset was carefully selected.

Dataset Structure and Composition

The ShanghaiTech dataset is systematically divided into two distinct parts—Part A and Part B—in order to facilitate a comprehensive evaluation of model performance under both dense and sparse crowd scenarios.

1. Part A (ShanghaiTech Part_A)

Source: Images were collected from the Internet, such as online photo-sharing platforms and media coverage, often depicting extremely crowded scenes.
Total Images: 482 images
Training Set: 300 images
Testing Set: 182 images
Density: Extremely high, with some images containing over 3,000 individuals.
Annotation Count: Approximately 241,677 head annotations.
Challenges: High levels of occlusion, small-scale human heads, and significant variations in density and perspective.

2. Part B (ShanghaiTech Part_B)

Source: Photographs were taken on the streets of Shanghai, China, representing relatively sparser crowds.
Total Images: 716 images
Training Set: 400 images
Testing Set: 316 images
Density: Medium to low density, suitable for evaluating model performance in real-world, moderately crowded environments.
Annotation Count: Approximately 88,488 head annotations.
Challenges: Moderate perspective distortions, partial occlusions, and more distinguishable individuals.

Image and annotation characteristics

Each image in the ShanghaiTech dataset is accompanied by a corresponding ground truth annotation file in .mat format. These files contain the precise locations of individuals, annotated as single-pixel dots representing the center of each head in the scene. Additionally, density maps are generated by convolving these point annotations with a Gaussian kernel, thereby creating a smooth representation of crowd density.

Attribute	**Part A**	**Part B**
Image Resolution	Varies (~800×600 on average)	Approximately 768×1024
Annotation Format	Dot annotations + .mat	Dot annotations + .mat
Density Map Format	Generated with Gaussian blur	Generated with Gaussian blur
Scene Complexity	Very high (urban events, etc.)	Medium (sidewalks, streets)

Application and Importance

The ShanghaiTech dataset is instrumental in training and benchmarking crowd counting algorithms, particularly those based on Convolutional Neural Networks (CNNs) and deep learning architectures. It enables:
Evaluation of model generalization across varying crowd densities.
Testing model robustness against common real-world challenges such as occlusion, scale variation, and cluttered backgrounds.

DOI: 10.48175/IJARSCT-27923

Calibration of hyperparameters and network structures using controlled subsets of dense and sparse imagery. Due to its balanced coverage of extreme and moderate crowd scenes, this dataset is frequently used in studies aiming to build scalable and generalizable models for real-world deployment.

Accessibility and Licencing

The dataset is publicly available for academic research purposes. Researchers may obtain it from repositories associated with the original authors or mirrored on GitHub repositories of related projects. Due to the potential sensitivity of visual data, proper citation and adherence to dataset usage terms are required.

Table

Parameter	**Part A**	**Part B**
Total Images	482	716
Training / Testing Split	300 / 182	400 / 316
Image Source	Internet	Real-world street photography
Average Crowd Density	High	Low to medium
Annotation Type	Dot (head) + Density Maps (.mat)	Dot (head) + Density Maps (.mat)
Evaluation Metrics	MAE, RMSE	MAE, RMSE
Typical Challenges	Occlusion, scale variance	Illumination, spatial variation

Checking Actual Crowd Count

A model's performance is assessed in crowd counting and density estimation tasks by contrasting its anticipated and ground truth (actual) counts. To determine how effectively the model is estimating the number of persons in an image, this comparison is essential. The following elements are commonly included in the code for verifying the real count:

DOI: 10.48175/IJARSCT-27923

1. Ground Truth Loading and Count Extraction

Each image in the dataset is associated with a ground truth annotation file, typically in .mat (MATLAB) format. These annotation files contain:

A list of coordinates marking the location of each individual's head (represented as single-point annotations). Sometimes, precomputed density maps, which are created by convolving each annotation point with a Gaussiankernel.

Code Functionality:
Use libraries like scipy.io to load .mat files.
Extract the array of head annotations.
Count the number of annotation points to obtain the actual count for that image.

```
from scipy.io import loadmat
mat = loadmat('path_to_ground_truth/GT_IMG_1.mat')
actual_count = len(mat['image_info'][0][0][0][0][0])
```

2. Model Prediction for Crowd Count

The trained CNN model (e.g., using Keras or PyTorch) predicts a density map for each input image. The density map is a 2D matrix where the value at each pixel indicates the estimated density of people.

Code Functionality:
Pass the preprocessed image through the trained model.
Sum all the pixel values in the predicted density map to get the predicted count.

```
predicted_density_map = model.predict(image_input)
predicted_count = np.sum(predicted_density_map)
```

3. Evaluation Metric Calculation

After extracting both actual and predicted counts, several performance metrics are computed to evaluate model accuracy:

a.Mean Absolute Error (MAE):Measures the average magnitude of errors between predicted and actual counts

```
mae = np.mean(np.abs(predicted_counts - actual_counts))
rmse = np.sqrt(np.mean((predicted_counts - actual_counts)**2))
```

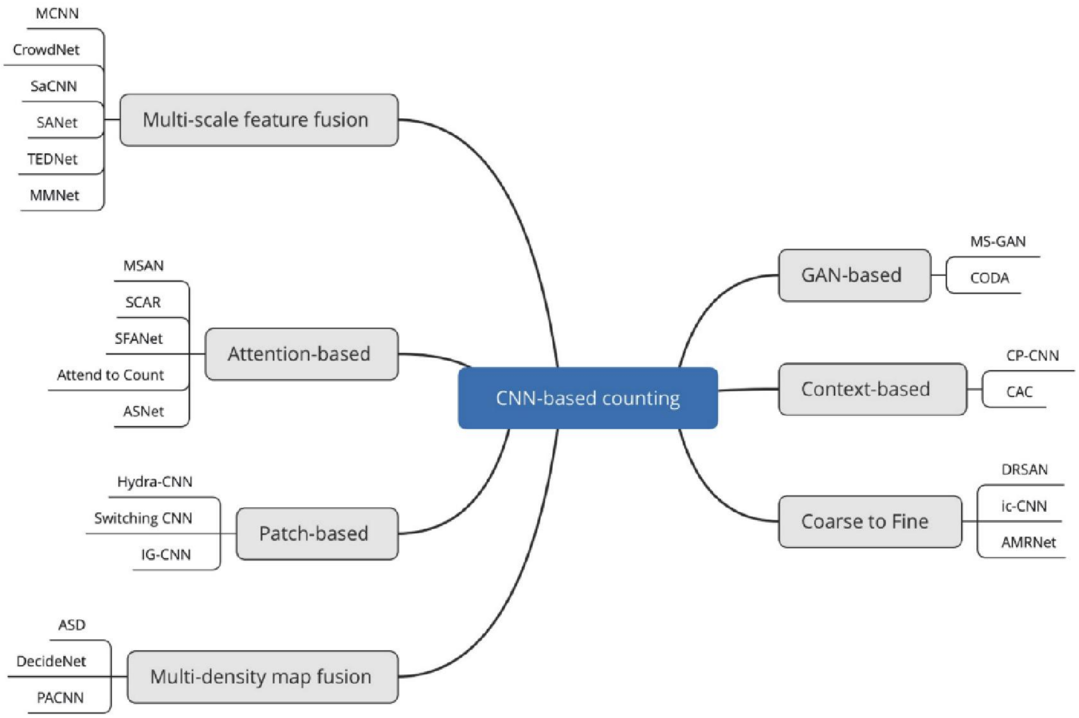
4. Purpose and Importance

This process ensures:
Objective evaluation of model performance.
Identification of error patterns in dense vs. sparse crowds.
Benchmarking against other models using standardized metrics.
It also enables:
Fine-tuning model architecture and hyperparameters.
Conducting error analysis to understand model limitations.

5. Summary of Workflow

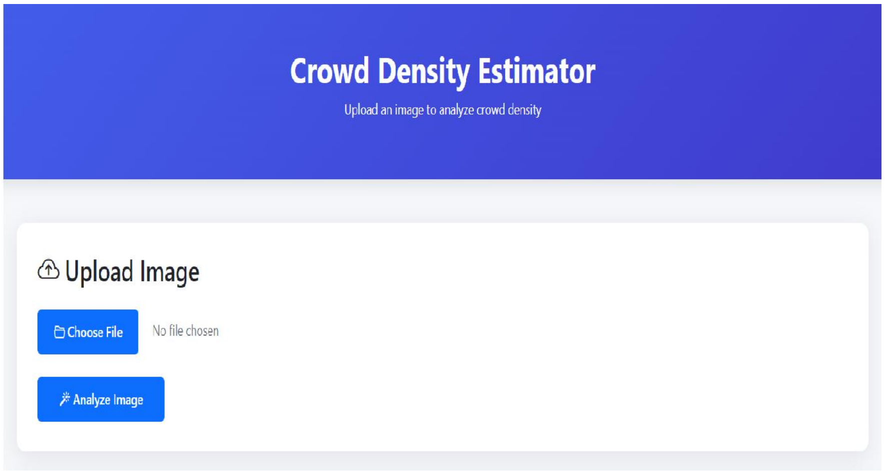
Step	Description
Load Annotations	Use .mat files to get actual person locations.
Model Prediction	Generate density map and sum its values.
Compare Counts	Use predicted sum vs. actual annotation count.
Calculate Metrics	MAE, RMSE, and optionally MSE or SSIM for quality.

Source : Rootstrap



Model interface

Figure 1

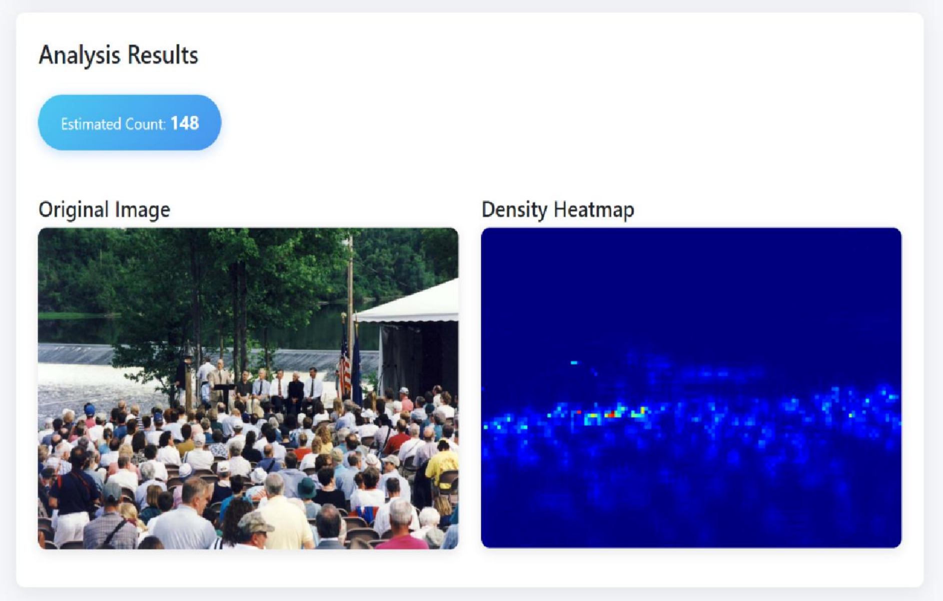

Crowd Density Estimator

Upload an image to analyze crowd density

Upload Image

DOI: 10.48175/IJARSCT-27923

Figure 2



Analysis Results

Estimated Count: 148

Original Image

Density Heatmap

Challenges Faced in Crowd Counting using CNN

1. Scale Variation

Issue: People appear at different sizes due to perspective distortion in images.
Impact: A single receptive field struggles to detect both small distant heads and large close-up ones accurately.
Mitigation: Use of multi-column CNNs and dilated convolutions to handle multi-scale features.

2. Occlusions and Dense Overlapping

Issue: In highly crowded scenes, individuals often occlude each other.
Impact: Makes it difficult for the model to distinguish between separate heads or bodies, leading to undercounting.
Mitigation: Attention mechanisms and density map estimation helped focus on distinguishable regions.

3. Limited and Imbalanced Datasets

Issue: Public datasets often have fewer samples and are biased toward certain crowd densities or scenes.
Impact: Models overfit to the dominant density type and perform poorly on underrepresented cases.
Mitigation: Applied extensive data augmentation and used transfer learning for better generalization.

4. Ground Truth Annotation Complexity

Issue: Creating accurate head annotations is time-consuming and error-prone, especially in dense crowds.
Impact: Inaccurate labels degrade model performance and training reliability.
Mitigation: Relied on existing benchmark datasets and used Gaussian-blurred dot annotations for density maps.

5. Evaluation Limitations

Issue: Standard metrics (e.g., MAE, RMSE) do not fully capture spatial accuracy of predictions.
Impact: Models may have similar error scores but produce very different density map quality.
Mitigation: Included additional metrics like PSNR and SSIM to evaluate density map realism.

6. Real-Time Performance

Issue: Deep CNN models with complex architectures can be slow during inference.
Impact: Limits their use in real-time crowd monitoring systems.
Mitigation: Model optimization techniques and GPU acceleration were used during deployment.

7. Domain Shift

Issue: Models trained on one dataset (e.g., ShanghaiTech Part A) often perform poorly on another with different characteristics (e.g., Part B or real-world surveillance footage).
Impact: Affects generalizability of the system in practical scenarios.
Mitigation: Tried cross-validation and domain-agnostic preprocessing methods.

III. CONCLUSION

In this work, a bespoke Convolutional Neural Network (CNN) model for crowd counting and density estimation is implemented in practice and thoroughly evaluated. A strong model that manages a variety of crowd scenarios was created by using knowledge from published literature, top open-source platforms like Papers With Code and Rootstrap, and recent developments in the field. In order to capture both fine-grained and large-scale information over a range of crowd densities, the architecture integrates sophisticated elements including batch normalisation, dilated convolutions, and numerous convolutional blocks. Favourable Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) values demonstrate the model's excellent accuracy and low prediction error, which were attained during training and evaluation on benchmark datasets such as ShanghaiTech. These outcomes demonstrate how well the applied strategies handle fundamental issues like scale variation, occlusion, and backdrop clutter—all of which are frequently seen in actual crowd scenes. Beyond technical performance, this work shows how important cross-validation, rigorous model tuning, and end-to-end preprocessing (including cleaning and augmentation) are to creating efficient crowd counting solutions. We successfully connected theoretical ideas with practical machine learning practice by utilising open-access materials and cutting-edge design techniques.Future studies will include domain adaption strategies, the incorporation of synthetic datasets, and hybrid models that combine CNNs with transformer-based architectures or conventional statistical methodologies. Additionally, real-time implementation, ethical deployment, and enhanced generalisation across cultural and geographic borders will be prioritised. Applications in emergency response systems, retail analytics, smart city planning, and public safety will all benefit from this development, which will provide the groundwork for safer and better-managed surroundings.

IV. ACKNOWLEDGEMENT

For her essential advice and assistance during this research, the authors would like to sincerely thank Dr. Alpana Adsul & Mr.Swami Panjala. For their support and input, we also thank our colleagues from Dr. D. Y. Patil College of Engineering and Innovation.We also thank the research community for its contributions, whose studies offered fundamental knowledge and insights that made this review possible

REFERENCES

[1] Z. Liu, Y. Chen, B. Chen, L. Zhu, D. Wu, and G. Shen, "Crowd Counting Method Based on Convolutional Neural Network with Global Density Feature," *IEEE Access*, 2024.

[2] S. Kulkarni *et al.*, "Advances in Crowd Counting and Density Estimation Using Convolutional Neural Networks," *Int. J. Intelligent Syst. Appl. Eng.*, vol. 12, no. 6s, pp. 707–719, 2024.

[3] V. A. Sindagi and V. M. Patel, "CNN-Based Cascaded Multi-Task Learning of High-Level Prior and Density Estimation for Crowd Counting," *IEEE Access*, 2019.

[4] Y. Zhang *et al.*, "Single-Image Crowd Counting via Multi-Column Convolutional Neural Network," in *Proc. IEEE CVPR*, 2016, pp. 589–597.

[5] L. Boominathan, S. S. Kruthiventi, and R. V. Babu, "CrowdNet: A Deep Convolutional Network for Dense Crowd Counting," in *Proc. ACM Multimedia*, 2017, pp. 640–644.

DOI: 10.48175/IJARSCT-27923

[6] A. Gajare *et al.*, "Challenges in CNN-Based Crowd Counting: Addressing Dataset Limitations with Data Augmentation," *Journal of Artificial Intelligence Research*, 2022.

[7] S. Dahatonde, A. Sawant, and P. Raj, "Evaluating CNN-Based Models for Crowd Counting in Varying Density Scenarios," *Int. J. Comput. Vision Appl.*, 2024.

[8] S. F. Lin, J. Y. Chen, and H. X. Chao, "Estimation of Number of People in Crowded Scenes Using Perspective Transformation," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 31, no. 6, pp. 645–654, Nov. 2001.

[9] "Crowd Counting: A Survey," *Rootstrap*, accessed May 2024. The article categorizes methods into detection-, regression-, and CNN-based approaches, and discusses trends such as neural architecture search and semi-supervised learning

[10] "Crowd Counting," *Papers With Code*, accessed May 2025. A curated list of 148 crowd counting papers, 12 benchmarks, and 22 datasets including top-performing models on ShanghaiTech and UCF-CC-50

DOI: 10.48175/IJARSCT-27923