Grided Heat-Map for Indoor CNN Crowd Counting

Saba Farshbaf Lame UCID: 30236369 ENEL645-Final Project
https://github.com/SabaIsCoding/645FinalProject

***Abstract*—Avoiding congested paths during evacuation or travel not only helps individuals arrive at their destination faster but also ensures their safety. In this study, we propose a machine learning-based approach for crowd counting in a public indoor space. Using a dataset of 2,000 frames containing over 60,000 pedestrian instances captured from a publicly accessible webcam in a mall, we generate a grid heat map that indicates congestion levels and the number of people in each grid. To achieve this, we compare the performance of YOLOv8, a pre-trained object detection model, with our custom Convolutional Neural Network (CNN) model. The accuracy of the models is evaluated using the Mean Absolute Error (MAE) and Mean Squared Error (MSE) to assess the predicted number of pedestrians. Our results show that the CNN model provided valuable insights into crowd density, with comparisons highlighting the strengths of this approach.**

***Index Terms*—crowd counting, density map, convolutional neural network, deep learning**

I. INTRODUCTION

Pathfinding has become one of the most important areas of research in various fields. Moving from a current location to a desired destination requires careful consideration of multiple factors, including the identification of connected paths, obstacles, accessibility, and congestion. To achieve minimum time cost, it is essential to have detailed information about these elements. Pathfinding becomes even more complex in unknown or dynamic environments, which may pose unpredictable dangers. The primary objective of pathfinding research is to identify short, safe, and less congested routes. This is particularly relevant in high-density scenarios, such as during large-scale events or gatherings, where disasters can occur due to overcrowding, as seen in events like the Hajj pilgrimage in Mecca [1]. Another critical aspect of crowd management is balancing the crowd density along paths, which is also a core goal of crowd counting. Accurate crowd counting and density estimation are vital challenges in image and video analysis. Video-based crowd counting, in particular, has proven to be a powerful tool for the early detection of overcrowded situations, which can significantly enhance crowd control measures [2]. However, in highly crowded scenes, factors such as mutual occlusion and varying scales complicate the task, leading to unsatisfactory performance of existing methods, thus hindering their practical adoption [3].

In this work, we propose a convolutional neural network (CNN)-based approach for crowd counting. Specifically, the CNN is trained to estimate crowd density maps at the grid level, and the predicted grid density maps are subsequently used to refine the pedestrian count. This hybrid approach leverages the strengths of CNNs such as their ability to learn complex features from images. While also employing regression methods to derive accurate crowd counts from the estimated grid density maps. Therefore, this method can be classified as a CNN-based crowd counting approach for final count prediction.

II. RELATED WORKS

Last crowd counting methods can be divided into three main categories: counting by detection [4], counting by regression [5], and CNN-based approaches [3].

Detection-based methods are based on target characteristics. Feature extraction is used in these methods, which can be divided into integral-based and parts-based extraction. The integral-based methods extract the features of the entire image. These methods do not have great accuracy in crowdedness as it is hard to extract all features. The paper [6] which used integral-based, provides a detailed analysis of pedestrian detection methods using monocular vision. It outlines a system composed of region-of-interest (ROI) selection, classification, and tracking for enhanced accuracy. The study evaluates characteristic extraction techniques such as Haar wavelets [7] and Histograms of Oriented Gradients (HOG) [8].

The part-based methods extract local features instead of global features. This paper [9] used head and shoulders detection as global features. It discusses that Omega-based shape which is a representation of head and shoulders shape is reliable. They had evaluated a lot of local features and found that Histograms of Oriented Gradients (HOG) [8] have been prone to be a good detector.

The paper [6] provides a detailed analysis of pedestrian detection methods using monocular vision. It outlines a system composed of region-of-interest (ROI) selection, classification, and tracking for enhanced accuracy. The study evaluates feature extraction techniques like Haar wavelets and Histograms of Oriented Gradients (HOG), with classifiers including SVMs [10] and neural networks. Experimental results highlight HOG with linear SVM as superior for high-resolution scenarios. This work is instrumental in understanding the trade-offs between accuracy and efficiency in pedestrian detection.

Recently, Deep Learning (DL) object detectors such as YOLO [11] and SSD [12] have been presented, which may perform dramatic detection accuracy in the sparse

scenes. They performance will present unsatisfactory results in highly crowded situations.

In [13] with feature mining, describes a multi-output regression framework for localized crowd counting. Instead of predicting the total count, the model predicts crowd counts in multiple regions by extracting and combining segment-based, structural-based, and local texture features. The datasets used are the proprietary Mall dataset and the UCID dataset [14]. The multi-output regression framework provides counting predictions for crowd by regularizing the least-square error minimization. For the evaluation the Mean Absolute Error (MAE) and Mean Square Error (MSE) [15] are used.

In the [16] they used a multi column CNN with 7 layers and a switchable objectives. They used patches with 72*72 pixels as input of the crowd CNN model. They introduced an iterative switching process in training deep crowd model to alternatively optimize the density map estimation task and the count estimation task every 6 epochs. The main task for the crowd CNN model is to estimate the crowd density map of the input patch and crowd counting. One loss function considered for density map and other loss function for crowd count. They set the scale weight of density loss to 10, and the scale weight of count loss to 1. Both losses use Euclidean distance as the objective function and are minimized via mini-batch gradient descent and back-propagation.

III. MATERIALS AND METHODS

This section discusses the dataset, data pre-processing techniques, data embedding, and the developed Deep Learning model.

1) Dataset Description: The mall dataset [13] was collected from a publicly accessible webcam to support research on crowd counting and profiling. It includes over 60,000 pedestrian instances across 2,000 video frames, with head position annotations for each individual in each frame. The dataset has a resolution of 640x480 pixels and a frame rate of less than 2 Hertz (Hz). A sample image of this dataset is shown in Fig. 1.

2) Exploratory Data Analysis: As mentioned earlier, the dataset includes both the count of people and head annotations for each individual. Fig. 2 shows provided data for head count and positions in each frame of the dataset. All photos are taken with a single camera from one angle. They do not differ in size or resolution. The area is not highly crowded, and people are sparse. The minimum number of people in each frame is 13, and the maximum is 53.

3) Data Pre-processing: Before directly using the data in building machine learning model, some pre-processings are done on images. In this pre-processing pipeline, random color adjustments are applied to the images to simulate variations in lighting and environmental conditions, improving the model's robustness. Since the images are captured from a single camera at a fixed angle, the image size remains unchanged. The augmentations include random adjustments to brightness in range of [-0.2, 0.2], random contrast value in range of [0.8, 1.2], random saturation in range of [0.8, 1.2], and random hue in range of [-0.1, 0,1]. These augmentations help the model learn to identify objects under a variety of lighting and color conditions, improving its ability to generalize to new and unseen images. The data is split into training, validation, and test sets, with 70%, 15%, and 15% of the data, respectively.

Fig. 1: Sample image from the mall dataset showing pedestrian instances [13].

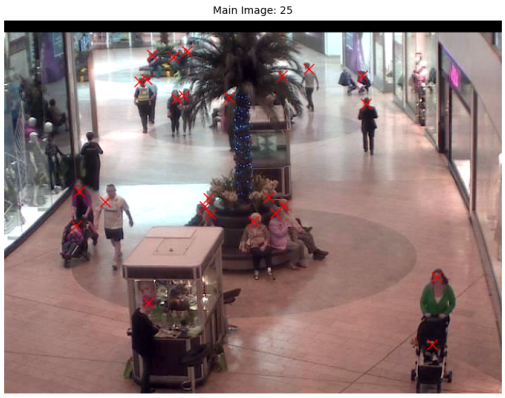
	frame_id	count	annotations	image_name
0	1	29	[[126.77986348122866, 60.70477815699661], [116...	seq_000001.jpg
1	2	30	[[57.155290102389046, 199.13481228668945], [87...	seq_000002.jpg
2	3	35	[[118.73899371069183, 43.77044025157227], [134...	seq_000003.jpg
3	4	31	[[140.87735849056602, 44.77672955974833], [151...	seq_000004.jpg
4	5	26	[[123.77044025157232, 51.82075471698107], [145...	seq_000005.jpg
...	...	...	...	...
1995	1996	27	[[454.8737166324436, 45.34599589322369], [462....	seq_001996.jpg
1996	1997	27	[[367.48151950718693, 59.14476386036961], [395...	seq_001997.jpg
1997	1998	25	[[22.512320328542103, 245.09958932238186], [73...	seq_001998.jpg
1998	1999	26	[[22.512320328542103, 290.43839835728943], [99...	seq_001999.jpg
1999	2000	26	[[87.56365503080085, 283.8675564681724], [241....	seq_002000.jpg

2000 rows × 4 columns

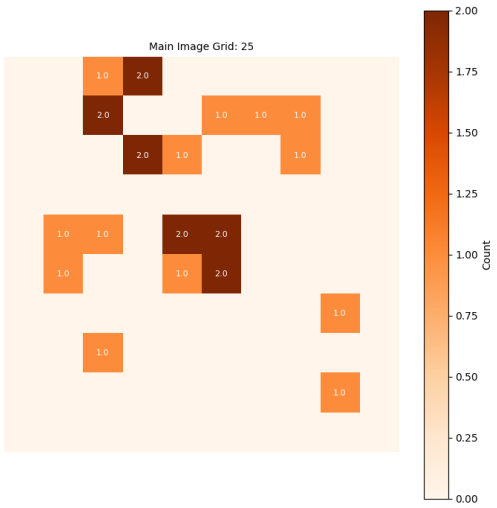
Fig. 2: Data frame from the dataset with head position information [13].

4) Data Embedding: In this work, for the pathfinding goal within an indoor environment to select the least crowded paths and areas, the area blueprint is considered as a grid. Using the head annotations provided in the dataset, a 10*10 grid heatmap [17] is created to visualize the distribution of people across the area. Each grid cell covers a region of 64*48 pixels in the image. Grids with fewer people or less crowding are displayed in darker colors. The color bar on the right side of each heatmap indicates the relationship between color intensity and the corresponding count of people in each grid.

In Fig. 3a for more clarification, crosses are used to represent the head in the images. While Fig. 4b represents the head count in each grid.



(a) A image from mall dataset with crosses in each head.



(b) A grid heat map for Fig. 3a that shows number of heads in each grid and total of 25 heads in image.

Fig. 3: Fig. 3a shows a sample images and Fig. 3b shows its grid heat map from the Mall dataset.

5) Models: In this research, a single column, modified version of a CNN model proposed in [16] and whole image-based inference is used. The input of this model is a combination of 640*480 image and its grid map base on heads position. Each grid covers 64*48 region of main image. The model utilizes a sequential architecture where layers are added linearly. It starts with a 2D convolutional layer using 32 filters of size 7*7, followed by ReLU activation and same padding to preserve the input's spatial dimensions. A 2*2 max pooling layer reduces the spatial dimensions by half. A second convolutional layer, identical to the first, is followed by another 2*2 max-pooling layer. A third convolutional layer with 64 filters of size 5*5 is applied, again using ReLU activation and same padding, followed by a 2*2 max-pooling layer for further down sampling. After the convolutional layers, the output is flattened into a one dimensional vector and passed through two fully connected layers: the first with 1,000 neurons and ReLU activation, and the second with 400 neurons, also using ReLU. The final output layer consists of 100 neurons with a linear activation function, representing a 10*10 grid. This output is reshaped into a 10*10*1 tensor to match the desired grid structure, with a ReLU activation layer applied after reshaping to ensure non-negative values. This architecture combines convolutional layers for feature extraction and dense layers for high-level representation learning, ultimately producing a structured 10*10 grid.

YOLOv8n [18] is a pre-trained model designed for object detection. As the nano version of the YOLOv8 architecture, it's optimized for real-time performance, making it ideal for tasks that require fast, efficient object detection on edge devices, like surveillance systems, autonomous vehicles, and mobile apps. The model is built with convolutional layers, activation functions, pooling layers, and fully connected layers. For our study, we used YOLOv8n as a benchmark to compare with our CNN model. While YOLOv8n can detect various object categories, we focused specifically on its ability to detect humans in images for the comparison.

A. Model Evaluation

To evaluate the crowd count, we use Mean Absolute Error (MAE) and Mean Squared Error (MSE) [15].

1) Mean Absolute Error (MAE): MAE is a commonly used evaluation metric in regression models. It represents the sum of the absolute differences between the predicted and ground truth values. A lower MAE indicates better model performance. it defined as follows:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{1}$$

where:

- n is the number of samples,
- y_i is the true value,
- $\hat{y}_i$ is the predicted value.

2) Mean Squared Error (MSE): MSE is another widely used evaluation metric for regression models. It represents the sum of the squared differences between the predicted and ground truth values. A lower MSE indicates a better fit of the model to the data. The formula for Mean Squared Error (MSE) is given by:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2}$$

where:

- n is the number of samples,
- y_i is the true value,
- $\hat{y}_i$ is the predicted value.

Both metrics have good results when their values are small, indicating that the model's predictions are close to the true values.

IV. RESULTS AND DISCUSSION

In this section, the results of prediction of both YOLOv8n and our CNN model are compared. Our training phase inculdes 100 epochs with 1e-6 of learning rate. As YOLO is a pre-trained model, we only run it for test phase.

A. Model Results

The results of both YOLOv8n and our CNN model for prediction of total count of people in the image and their grid heat map are presented while running them on images of test set. The predicted locations of heads in each image is presented as crosses on image and predicted number of people in each grid is presented in their predicted grid heat map.

1) Result of crowd count: The results in Table I indicates the differences between average of real count, maximum count and minimum count in test set and prediction count of CNN and YOLOv8n models. Our model achieve better results than YOLOv8n in detection. Our CNN detected 26.5, 39.5, and 15.5 for average count, maximum count, and minimum crowd count respectively and YOLOv8n detected 16.4, 31, and 6.0 for average count, maximum count, and minimum crowd count respectively.

TABLE I: The average maximum, minimum crowd count of each model is compared with real data.

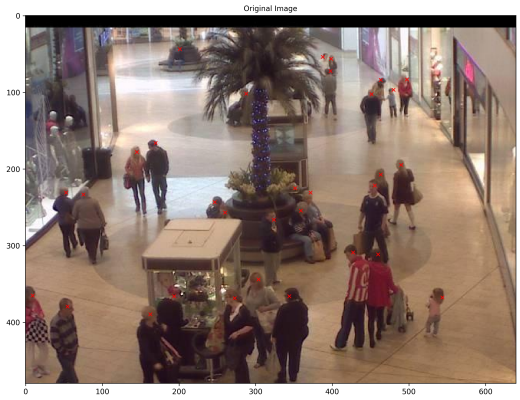
Model count	***Average count***	***Maximum count***	***Minimum count***
Ground truth	31.4	49.0	17
CNN	26.5	39.5	15.5
YOLOv8n	16.4	31.0	6.0

2) Crowd evaluation comparison: The results in Table II shows the MSE and MAE between real count in each grid with CNN and YOLOv8n prediction. The closer amount to 0 indicates less difference. For both of MSE and MAE, our CNN model achieves better result of experiments. While our CNN MSE is 0.2068 and MAE is 0.2158, the YOLOv8n MSE is 0.4130 and 0.2471 per grid.

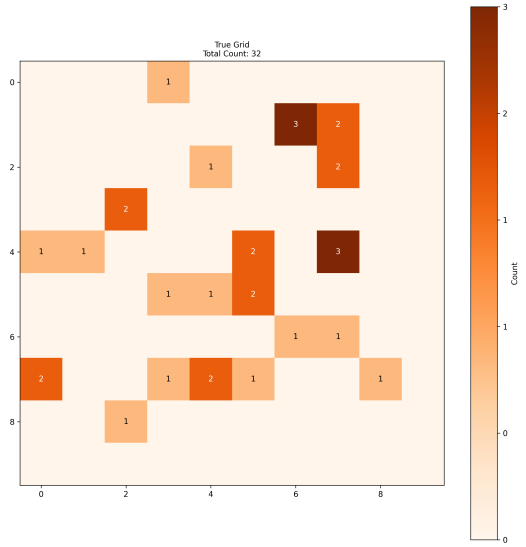
TABLE II: The table shows the results of CNN grid model and YOLO's MSE and MAE per each grid.

Model	***MSE per grid***	***MAE per grid***
CNN	0.2068	0.2158
YOLOv8n	0.4130	0.2471

3) Grid heat map visualization: In this part, the results of grid heat map are shown for both our CNN grid based model and YOLOv8n. However, YOLOv8n is not able to create grid heat map, but we generate a grid heat map based on its detected positions. The Fig. 4a is a random sample from test set includes 32 heads inside. The Fig. 4b is the representation of Fig 4a grid heat map. In Fig. 5b and Fig. 5a total predicted head counts are 25 and 23 respectively. The head counts predicted with CNN is rounded to the closest integer number.



(a) An image from test set with crosses on each head.



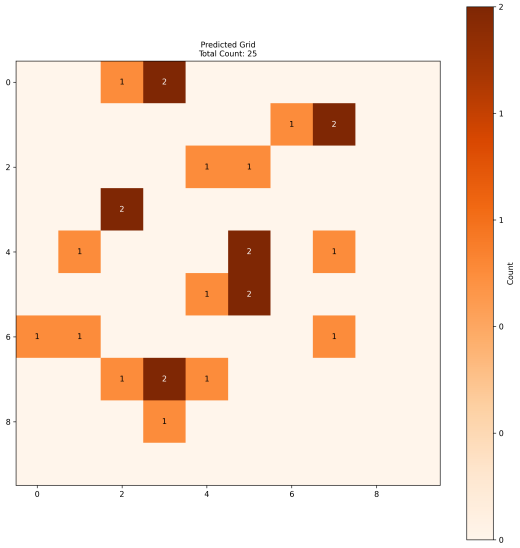
(b) A grid heat map for Fig 4a that shows number of heads in each grid and total of 32 heads in image.

Fig. 4: Fig. 4a shows a sample image from test set and Fig. 4b shows its grid heat map.

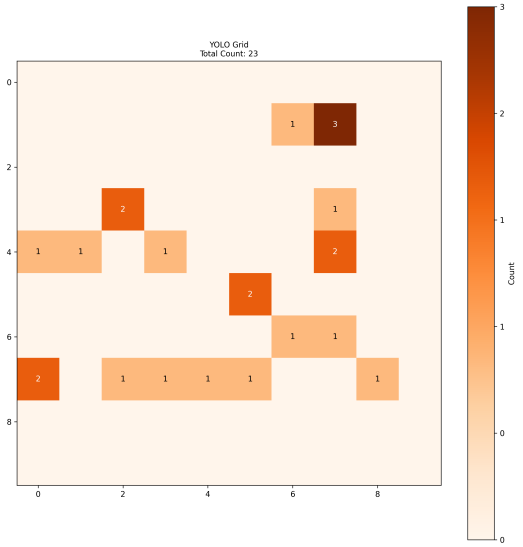
4) Strength and Limitations: Both the CNN grid-based model and YOLOv8n have their strengths, making them suitable for different tasks in computer vision. The grid-based model excels in accurate object counting and spatial analysis, making it ideal for applications like crowd counting and area detection. It provides detailed insights into object distribution, which is crucial for crowd management and urban planning.

In contrast, YOLOv8n is optimized for real-time object detection and fast localization, making it ideal for applications like autonomous driving or live object tracking. While YOLOv8n is faster, the grid-based model offers superior accuracy in tasks that require precise counting and density analysis.

Ultimately, the grid-based model is especially valuable in scenarios where accuracy is key, and as it evolves with more training, it can complement real-time detection methods like YOLOv8n for more comprehensive solutions.



(a) CNN predicted grid heat map for Fig. 4a with predicted count on each grid and total of 25 heads in image.



(b) YOLOv8n predicted grid heat map for Fig. 4a with predicted count on each grid and total of 23 heads in image.

Fig. 5: Fig. 5a shows the predicted grid heat map for Fig. 4a with the CNN model, and Fig. 5b shows the predicted grid heat map for Fig. 4a with the YOLOv8n model.

V. CONCLUSIONS

In this paper, we explored two different methods for crowd counting and object detection: a CNN grid-based model and YOLOv8n. While YOLOv8n is well-known for its ability to detect and localize objects in real-time, our CNN grid-based model excels at providing detailed crowd counts across specific areas of an image. This makes it especially useful for applications like monitoring crowd density or finding less crowded pathways in pathfinding scenarios. By combining both the crowd count and the grid map of the image, our method offers a reasonable level of accuracy in estimating the number of people in a given space.

When compared to YOLOv8n, our CNN grid-based model showed smaller differences between the predicted and actual crowd counts, particularly in terms of spatial accuracy. However, there is still room for improvement. One key step would be to train our model on a wider range of datasets with varying crowd sizes and densities. This would help our model perform better across different environments.

Looking ahead, we plan to refine our model by adding more layers and adjusting various parameters to boost its accuracy. We're also excited to experiment with multi-column CNN architectures, which could improve the robustness and precision of our approach. Ultimately, by combining both crowd counting and real-time detection, we hope to create a more effective system for managing crowds in busy public spaces, ensuring better safety and smoother navigation for everyone.

REFERENCES

[1] The Guardian, "A history of hajj tragedies," 2006. [Accessed: July 1, 2013].

[2] C. C. Loy, K. Chen, S. Gong, and T. Xiang, "Crowd counting and profiling: Methodology and evaluation," in *Modeling, Simulation and Visual Analysis of Crowds: A Multidisciplinary Perspective*, pp. 347–382, Springer, 2013.

[3] B. Li, H. Huang, A. Zhang, P. Liu, and C. Liu, "Approaches on crowd counting and density estimation: a review," *Pattern Analysis and Applications*, vol. 24, pp. 853–874, 2021.

[4] H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. Al-Maadeed, N. Rajpoot, and M. Shah, "Composition loss for counting, density map estimation and localization in dense crowds," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 532–546, 2018.

[5] A. C. Davies, J. H. Yin, and S. A. Velastin, "Crowd monitoring using image processing," *Electronics & Communication Engineering Journal*, vol. 7, no. 1, pp. 37–47, 1995.

[6] M. Enzweiler and D. M. Gavrila, "Monocular pedestrian detection: Survey and experiments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 12, pp. 2179–2195, 2008.

[7] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, pp. 137–154, 2004.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1, pp. 886–893, Ieee, 2005.

[9] M. Li, Z. Zhang, K. Huang, and T. Tan, "Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection," in *2008 19th international conference on pattern recognition*, pp. 1–4, IEEE, 2008.

[10] M. A. Chandra and S. Bedi, "Survey on svm and their application in image classification," *International Journal of Information Technology*, vol. 13, no. 5, pp. 1–11, 2021.

[11] J. Redmon, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pp. 21–37, Springer, 2016.

[13] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature mining for localised crowd counting.," in *Bmvc*, vol. 1, p. 3, 2012.

[14] A. B. Chan and N. Vasconcelos, "Counting people with low-level features and bayesian regression," *IEEE Transactions on image processing*, vol. 21, no. 4, pp. 2160–2177, 2011.

[15] T. Hastie, "The elements of statistical learning: data mining, inference, and prediction," 2009.

[16] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 833–841, 2015.
[17] N. Gehlenborg and B. Wong, "Heat maps," *Nature Methods*, vol. 9, no. 3, p. 213, 2012.
[18] Ultralytics, "Yolov8: Usage examples," 2023. Accessed: 2024-12-05.