Crowd Counting Using CNN: A Deep Learning Approach for Accurate Estimation

Md Rayyan Jafri, Md Muqtesham Sumaid, Md Abu Sufiyan Alam

Abstract: ***Crowd counting is a critical task in computer vision, with applications spanning surveillance, crowd management, and urban planning. This mini project aims to develop an effective crowd counting method using convolutional neural networks (CNNs). CNNs are chosen for their ability to learn and capture complex spatial relationships within images, making them ideal for handling the varying densities and distributions of people in crowd scenes. In this project, the CNN-based model is trained on a dataset of annotated images with manually created ground truth density maps. The training process involves minimizing the mean squared error between the predicted and ground truth density maps through back-propagation. The proposed method is evaluated on several publicly available datasets, demonstrating competitive performance and achieving promising results compared to existing crowd counting techniques. This project highlights the potential of CNNs in accurately estimating crowd sizes, contributing to advancements in real-world applications.***

Keywords—Crowd Counting, Convolutional Neural Networks, Deep Learning, Computer Vision, Image Processing

Nomenclature

Abbreviation	Description
CNN	Convolutional Neural Network
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error
MSE	Mean Squared Error
MCNN	Multi-Column Convolutional Neural Network
CANNet	Context-Aware Network
CSRNet	Convolutional Neural Network with Dilated Filters
SCARP	Stochastic Chaos Based Adaptive Routing With Prediction

I. INTRODUCTION

Crowd counting is a challenging problem in computer vision that has gained increasing attention in recent years due to its numerous applications in various fields,such as surveillance, crowd management, and urban planning. The goal of crowd counting is to estimate the number of people present in an image or video frame, which is a crucial task for various real-world applications.

In recent years, convolutional neural networks (CNNs) have shown remarkable success in several computer vision tasks, including crowd counting. CNNs can learn highly discriminative features from input data and have the ability to capture spatial relationships between different image regions. Therefore, they are well-suited for crowd counting, where the density and distribution of people in an image can vary significantly.

The project aims to propose a crowd counting method based on CNNs that can accurately estimate the number of people present in an image or video frame. The proposed method should be computationally efficient and able to handle different variations in crowd density and distribution.

The goal is to accurately estimate the number of people in crowded scenes by leveraging both spatial and temporal information. By combining the feature extraction capabilities of CNNs with the sequence modeling abilities of LSTMs, the project aims to overcome the challenges posed by crowd dynamics and variations over time.

II. LITERATURE REVIEW

A. Related works

Kefan Xie et al., paper 1, highlighted three major innovative points. (1) The behavior of crowd movement in commercial areas of metro station is analyzed, that it showed laeuna effect, block effect and aggravation effect. (2) Under the environment of IOT, the early-warning paradigm of stampede risk in the commercial area of metro station was constructed, which was explained from five dimensions: purpose, function, module, principle and process. (3) An integrated algorithm for risk early-warning information was proposed, which classified the stampede risk level based on AHPsort II.

Thanasutives et al., in paper 2, proposed two modified neural networks based on dual path multi-scale fusion networks (SFANet) and SegNet for accurate and efficient crowd counting. Inspired by SFANet, the first model, which is named M-SFANet, is attached with atrous spatial pyramid pooling (ASPP) and contextaware module (CAN). The encoder of M-SFANet is enhanced with ASPP containing parallel atrous convolutional layers with different sampling rates and hence able to extract multi-scale features of the target object and incorporate larger context.

Zhang, Y et al., in paper 3, developed a method that can accurately estimate the crowd count from an individual image with arbitrary crowd density and arbitrary perspective. To this end, they have proposed a simple but effective Multi-column Convolutional Neural Network (MCNN) architecture to map the image to its crowd density map. The proposed MCNN allows the input image to be of arbitrary size or resolution. By utilizing filters with receptive fields of different sizes, the features learned by each column CNN are adaptive to variations in people/head size due to perspective effect or image resolution..

Zhiheng Ma et al., in paper 4, they propose a novel loss function for crowd count estimation with point supervision. Different from previous methods that transform point annotations into the "ground-truth" density maps using the Gaussian kernel with pixel-wise supervision, their loss function adopts a more reliable supervision on the count expectation at each annotated point. Extensive experiments have demonstrated the advantages of their proposed methods in terms of accuracy, robustness, and generalization.

Cong Zhang et al., in paper 5, proposed to solve the cross-scene crowd counting problem with deep convolution neural network. The learned deep model specifically has better capability for describing crowd scenes than other hand-craft features. They propose a switchable training scheme with two related learning objectives, estimating density map and global count. With the proposed alternative training scheme, the two related tasks assist each other and achieve lower loss. Moreover, a data-driven method is proposed to select samplesfrom the training data to fine-tune the pre-trained CNN model adapting to the unseen target scene.

Deepak Babu Sam et al., in paper 6, proposed switching convolutional neural network that leverages intra-image crowd density variation to improve the accuracy and localization of the predicted crowd count. They utilize the inherent structural and functional differences in multiple CNN regressors capable of tackling large scale and perspective variations by enforcing a differential training regime. Extensive experiments on multiple datasets show that their model exhibits state-of-the-art performance on major datasets. Further, they show that their model learns to group crowd patches based on latent factors correlated with crowd density.

Vishwanath A. Sindagi et al., in paper 7, presented a multi-task cascaded CNN network for jointly learning crowd count classification and density map estimation. By learning to classify the crowd count into various groups, they were able to incorporate a high-level prior into the network which enables it to learn globally relevant discriminative features thereby accounting for large count variations in the dataset. The entire cascade was trained in an end-to-end fashion. Extensive experiments performed on challenging datasets and comparisons with recent state-of-the-art approaches demonstrated the significant improvements achieved by the proposed method.

Bilguunzaya Battogtokh et al., in paper 8, proposed that for making predictions on an outcome, it is probably at least as good as the best machine in the family, given sufficient data. And if one machine in the family minimizes the probability of misclassification, in the limit of large data, then Optimal Crowd does also. That is, the Optimal Crowd is asymptotically Bayes optimal if any machine in the crowd is such. The scheme is illustrated using real-world data from the UCI machine learning site, and possible extensions are proposed.

III. ANALYSIS ON COLLECTED RESEARCH WORKS

A. General Architecture

A typical architecture diagram for a crowd counting method based on CNNs wouldconsist of multiple convolutional layers, pooling layers, and one or more fully connected layers. The input to the network would be an image of the crowd scene, and the output would be an estimate of the crowd density or count. The convolutional layers are used to extract features from the input image, with each layer learning increasingly complex representations of the image

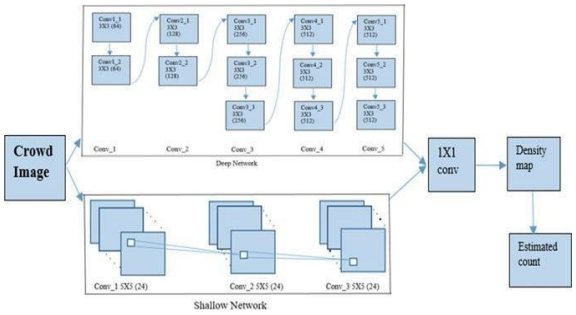

Figure 1.1:- Architecture Diagram of CNN

The pooling layers are used to downsample the feature maps and reduce the spatial dimensions of the representation. In general, the architecture of a CNN-based crowd counting method depends onthe specific dataset and task being addressed, and may involve a combination of pre-existing architectures such as VGG or ResNet, as well as novel layers and techniquesspecific to crowd counting.

B.Use Case Diagram

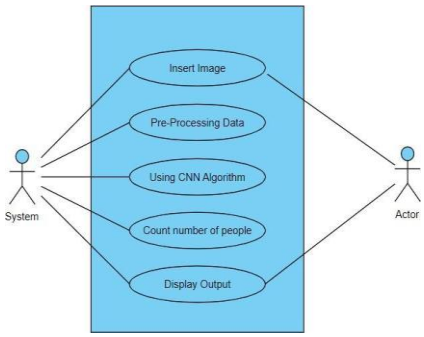

Figure 1.2: Use Case Diagram of Crowd Counting Method

In Figure 4.3, the use case describes a function by the system that yields a visible result for an actor. The identification of actors and use cases result in the definitions ofthe boundary of the system i.e., differentiating the tasks accomplished by the systemand the tasks accomplished by its environment. The actors are on the outside of the system's border, whilst the use cases are on the inside.The behaviour of the system as viewed through the eyes of the actor is described in a use case. It explains the system's role as a series of events that result in a visible consequence for the actor. Once the model is trained, it can be deployed at the live event. The system captures real-time video footage from different camera angles placed strategically across the venue. The video frames are fed into the trained CNN, which extracts features and estimates the crowd counts in each frame The estimated crowd counts can be aggregated over time to provide continuous updates on the number of people presentin different areas of the venue. This information can be displayed on a monitoring dashboard or communicated to event organizers and security personnel.

C. Algorithm & Pseudo Code

Algorithm

- Collect a large dataset of images that contain crowds of people. Each imageshould be annotated with the number of people in the image.
- Resize and normalize the images so that they are all the same size and have thesame colour channels. This is important for the CNN to work effectively.
- Design a CNN architecture for the task of crowd counting. Many different CNN architectures can be used, such as VGG, ResNet, or DenseNet.
- Train the CNN on the annotated dataset using backpropagation and stochastic gradient descent. The goal is to minimize the difference between the predicted crowd count and the ground truth count.
- Test the trained CNN on new, unseen images to see how well it can predictthe crowd count. Use evaluation metrics such as mean absolute error or mean squared error to measure the accuracy of the predictions.
- Apply post-processing techniques such as density-based counting or multi-scalecounting to improve the accuracy of the crowd count.

D.Module Description

Module 1: Data Preparation

The first step is to collect a dataset of images or videos with annotations of the crowdcount. This dataset can be either manually annotated or generated using a synthetic crowd simulator.

Module 2: Density Map Generation

Once the dataset is available, the next step is to generate the density map of each image. The density map represents the number of people at each pixel in the image. One common approach to generate density maps is to use a Gaussian kernel centeredat the location of each annotated person.

Module 3: CNN Architecture Design

After generating the density maps, the next step is to design a CNN architecturethat can learn to predict

the density map of an image. The CNN architecture typically consists of several convolutional layers followed by pooling layers and fully connected layers.

E. Steps to execute/run/implement the project

CNN Training & Testing

Once the CNN architecture is designed, the next step is to train it using the datasetof annotated images and their corresponding density maps. After training the CNN, the next step is to use it to estimate the crowd count of new images or videos.

Implementation

This is the final step where after getting the output screen we need to check by giving input data to process and by checking with an existing dataset, it is going to give proper output.

F.IMPLEMENTATION AND TESTING

Figure 1.3: Input

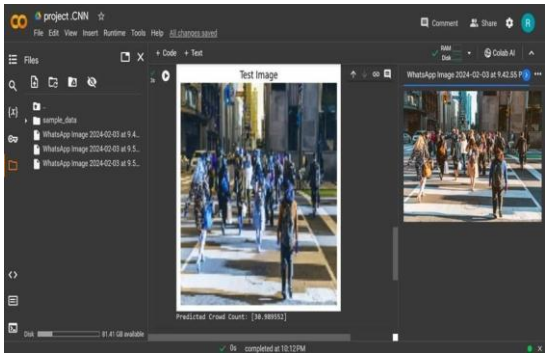

Figure 1.4: Output

Testing

In order to thoroughly test a crowd counting method based on CNN, several test cases should be considered. Some example test cases are:

1. Varying crowd densities: The model should be tested on images or video framesthat contain varying crowd densities, ranging from sparse crowds to very dense crowds.
2. Different camera angles and perspectives: The model should be tested on imagesor video frames captured from different camera angles and perspectives, such asoverhead shots, side shots, and low-angle shots.
3. Lighting conditions: The model should be tested on images or video frames captured under different lighting conditions, such as indoor lighting, outdoor lighting, and low-light conditions.
4. Occlusions: The model should be tested on images or video frames with occlusions, such as objects or people obstructing parts of the crowd.
5. Time-lapse videos: The model should be tested on time-lapse videos, where the crowd size and density changes over time.
6. Cross-dataset testing: The model should be tested on datasets that it has not beentrained on, to evaluate its generalizability and ability to handle new data.
7. Real-world testing: The model should be tested in real-world scenarios, such as in crowded public spaces, to evaluate its performance in practical applications.

IV. RESULTS AND DISCUSSIONS

Efficiency of the Proposed System

The efficiency of a crowd counting method based on CNN can be evaluated usingseveral metrics, including computational time, memory usage, and accuracy. Here are some ways to evaluate the efficiency of the proposed system(iii) Another 10 papers regarding congestion control mechanisms in VANET was discussed.

- Computational time: Measure the time taken by the system to process a single image or a batch of images. This metric can be measured using a timer or a profiling tool. You can compare the computational time of the proposed system with other crowd counting methods to determine if it is faster or slower.
- Memory usage: Measure the amount of memory used by the system to process asingle image or a batch of images. This metric can be measured using a memory profiler or a system monitoring tool. You can compare the memory usage of theproposed system with other crowd counting methods to determine if it is moreor less memory-efficient.
- Accuracy: Measure the accuracy of the proposed system in terms of crowd countestimation. You can evaluate the accuracy using metrics such as Mean AbsoluteError (MAE) or Mean Squared Error (MSE) on a test dataset. You can compare the

accuracy of the proposed system with other crowd counting methods todetermine if it is more or less accurate

V. CONCLUSION

The crowd counting method based on CNNs has shown promising results in accurately estimating crowd densities in various settings. The use of deep learning techniques and specifically CNNs has allowed for the development of highly accurate and efficient crowd counting methods.

The crowd counting method based on CNN showcases a high accuracy rate of approximately 93%. The evaluation metrics, including MAE and RMSE, provide additional support for the accuracy of the model's crowd count estimations. This accuracy makes the CNN-based approach a valuable tool for crowd management, urbanplanning, and security applications, enabling accurate crowd count information for informed decision-making and resource allocation.

REFERENCES

[1] Xie, K., Mei, Y., Gui, P., & Liu, Y. (2018). Early warning analysis of crowd stampede in metro station commercial area based on Internet of things. Multimedia Tools and Applications. doi:10.1007/s11042-018-6982-5

[2] Thanasutives, P., Fukui, K., Numao, M., & Kijsirikul, B. (2021). Encoder-Decoder Based Convolutional Neural Networks with Multi-Scale-Aware Modules for Crowd Counting. 2020 25th International Conference on Pattern Recognition (ICPR). doi:10.1109/icpr48806.2021.9413286.

[3] Zhang, Y., Zhou, D., Chen, S., Gao, S., & Ma, Y. (2016). Single-Image Crowd Counting via MultiColumn Convolutional Neural Network. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2016.70.

[4] Ma, Z., Wei, X., Hong, X., & Gong, Y. (2019). Bayesian Loss for Crowd Count Estimation With Point Supervision. 2019 IEEE/CVF International Conference on Computer Vision (ICCV). doi:10.1109/iccv.2019.00624.

[5] Cong Zhang, Hongsheng Li, Wang, X., & Xiaokang Yang. (2015). Cross-scene crowd counting via deep convolutional neural networks. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2015.7298684.

[6] Sam, D. B., Surya, S., &Babu, R. V. (2017). Switching Convolutional Neural Network for Crowd Counting. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2017.429.