A Multi-Modal (CNN & AE) Feature Fusion Strategy for Crowd Count in Still Images

Report submitted in fulfillment of the requirements
for the Exploratory Project of

Second Year B.Tech.

by

Rishabh Bansal
18075049

S Ujjwal
18075052

Shayak Das
18075056

Under the guidance of

Prof. Rajeev Srivastava

Department of Computer Science and Engineering
INDIAN INSTITUTE OF TECHNOLOGY (BHU) VARANASI
Varanasi 221005, India
May 2017

Dedicated to

our parents, teachers and professors for their constant guidance and support in accomplishing the task.

Declaration

We certify that

1. The work contained in this report is original and has been done by ourself and the general supervision of our supervisor.
2. The work has not been submitted for any project.
3. Whenever We have used materials (data, theoretical analysis, results) from other sources, We have given due credit to them by citing them in the text of the thesis and giving their details in the references.
4. Whenever We have quoted written materials from other sources, We have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Place: IIT (BHU) Varanasi

Date: 12/June/2020

Rishabh Bansal
S Ujjwal
Shayak Das

B.Tech. Student
Department of Computer Science and Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Certificate

*This is to certify that the work contained in this report entitled "**A Multi-Modal (CNN AE) Feature Fusion Strategy for Crowd Count in Still Images** " being submitted by **Rishabh Bansal** (**Roll No. 18075049**) ,**S Ujjwal** (**Roll No. 18075052**), **Shayak Das** (**Roll No. 18075056**), carried out in the Department of Computer Science and Engineering, Indian Institute of Technology (BHU) Varanasi, is a bona fide work of our supervision.*

Place: IIT (BHU) Varanasi
Date: 12/June/2020

Prof. Rajeev Srivastava
Department of Computer Science and Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Acknowledgments

We would like to extend our gratitude to Prof. Rajeev Srivastava to give us the opportunity to invest in this Exploratory Project . The help and effort given by Sir Santosh Tripathi proved to be indispensable in the completion of this Project .

Place: IIT (BHU) Varanasi

Date: 12/June/2020

Rishabh Bansal

S Ujjwal

Shayak Das

Abstract

Estimation of crowd counting is a technique of counting number of people in a crowd. Crowd Count estimation have wide range of applications such as crime detection,congestion,public safety,crowd abnormalities,crowd anomalies,surveillance ,urban planning etc. The purpose of crowd estimation is to calculate concentration of crowd . The job of detecting face in crowd is complicated due to various reasons like hinderance in crowd,complications in human faces including color,pose,expressions . But thanks to Pattern recognition techniques counting performance has been steadily improved. Different Convolution Neural Networks (CNN) and Generative models have been proposed for crowd counting in still images. Although the existing models provide better performance but still there is scope to improve the performance of the crowd counting system. Based on these intuition we proposed a multi-modal fusion strategy using CNN and Auto-Encoders for crowd counting. The experiments are done by using two publicly available datasets like Pets-2009 and UCF-CC-50. The results outperform the state-of-the-art techniques.

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

Exponential increase in worldwide population influences day to day traffic, security and even causes threats to human life. Thus, it demands an efficient crowd analysis (CA) model for crowd monitoring. We have various crowd analysis related tasks but not limited to crowd count and/or density estimation (CCDE) [18, 16, 7, 22], crowd anomaly detection (CAD)[4, 12], crowd congestion level analysis[8], crowd scene understanding and crowd flow analysis. Among these CCDE is the backbone of CA. CCDE has two words such as crowd count defines total number of people in the scene and density estimation refers to crowd counting using density map analysis but these two terms combinedly used in the literature to address crowd count. The crowd count information not only measure crowd occupancy of a given area but also help to detect congestion levels[8], overcrowd or under-crowd, crowd disasters like panic, stampede[4], fight [12], public safety and space management which are then helpful in crowd monitoring. Now a day's development of smart cities demands efficient crowd counting mechanism for automatic bus scheduling[20]. The crowd disasters generally occur in public rallies, protests, festivals, shopping malls, etc. and causes many lives under threat. A statistic[1] shows that over thousands of people losses their life due to

crowd disasters. The figure-1 shows a incident due to over crowd. We can avoid such incidents by providing effective crowd monitoring system which depends on accurate crowd count. We can't deny the fact that the CCDE is entirely an interdisciplinary research area because the counting approaches can be used in other research topics like cell-[14], traffic monitoring[15], security-surveillance activities, wild-life preserve [19]. So, the diversity nature of CCDE motivate researchers in this field. The CCDE can be accomplished by using vison-based approaches as well as non-vison-based approaches. The former requires image or video processing tools to capture video or still images and a programming model to count crowd from the acquired image or video. But the later approach depends on sensors (Passive Infra-Red (PIR), CO2, Smart-Meter etc.) or IoT devices[24] to count crowd. The later approach has a drawback that these are limited to small area containing few numbers of crowd but not applicable to wide area scenarios containing thousands of crowds. But the vison-based approaches provide accurate count information in varying crowd densities. Again, the vison-based CCDE is better as well as main concern for research.

We have used concepts of Convolutional Neural Network (CNN), autoencoder, Multilayer Perceptron(MLP) in our model. In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analyzing visual imagery. [17] An autoencoder is a type of artificial neural network used to learn efficient data codings in an unsupervised manner. The aim of an autoencoder is to learn a representation (encoding) for a set of data, typically for dimensionality reduction, by training the network to ignore signal "noise".[10] A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN). The term MLP is used ambiguously, sometimes loosely to any feedforward ANN, sometimes strictly to refer to networks composed of multiple layers of perceptrons (with threshold activation). Multilayer perceptrons are sometimes colloquially referred to as "vanilla" neural networks, especially when they have a single hidden layer.[5]

Figure 1.1 A Crowd Image Example - Bank of Godavari,Datia District [source-https://blogs.timesofindia.indiatimes.com/random-harvest/stampedes-as-indian-as-taj-mahal/]

We have divided our entire model into three parts. In First part we have used simple Convolutional Neural Network to devise a model to estimate the crowd count in a static image. In Second part we have used autoencoder to encode the images in a compressed representation and then use this compressed representation to train a Multilayer Perceptron model to estimate crowd count in a static image. In Final task we ensemble the two previous models to form a new model and use it for the same task.

1.2 Motivation of the Research Work

Problems arises due to crowd can be solved if there is a proper surveillance system which can be achieved by Crowd count analysis methods. Surveillance system can be used for detection and count people, crowd level and also alarms as the presence of the dense crowd.

1.3 Organisation of the Report

Report is organised as Section 1.1 gives an overview of our estimating crowd count model and need of Crowd counting model for our society. Section 1.2 states the

1.3. Organisation of the Report

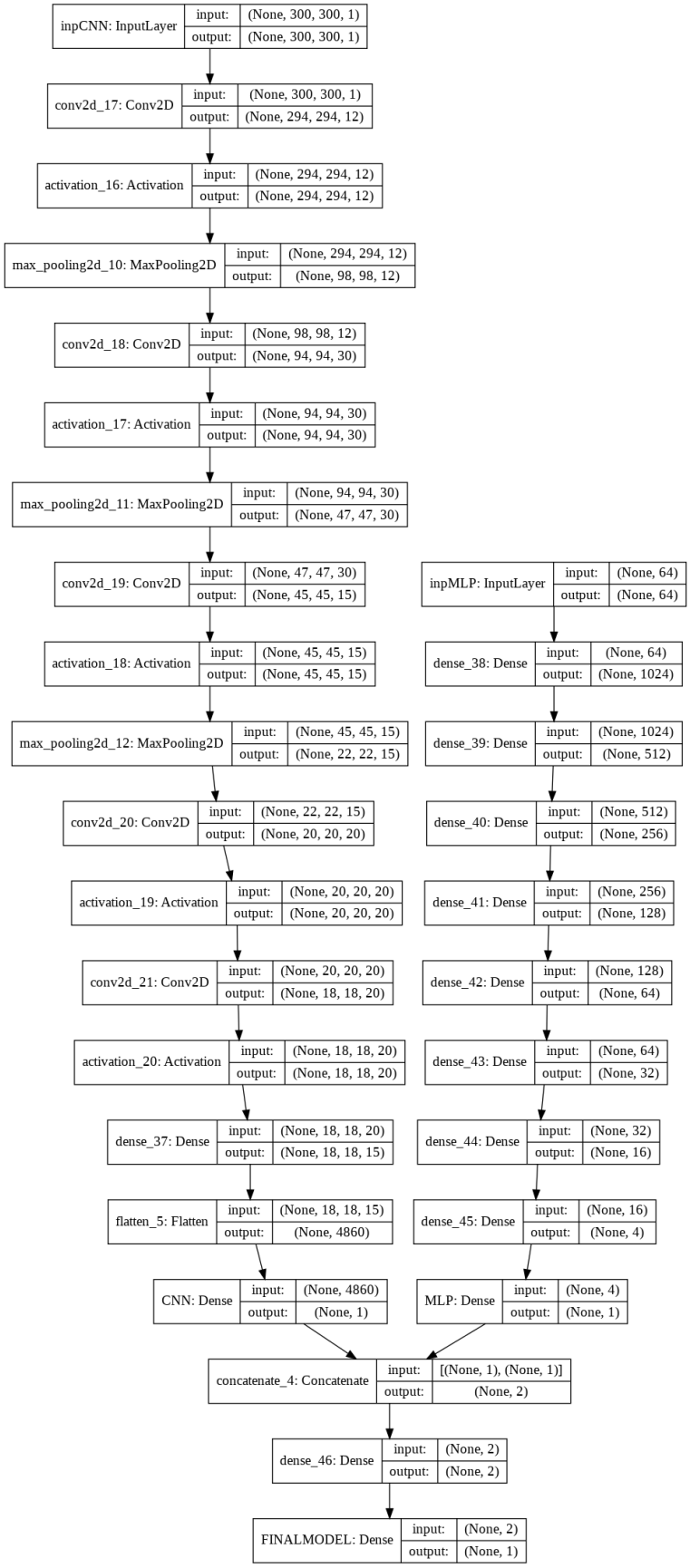

Figure 1.2 Overview of crowd count model

motivation behind the project.Section 1.3 states how the report is organized. Section 2 gives literature review of approaches used to solve crowd analysis problem. Section 3 gives a detailed analysis of our model. Section 4 describes results we obtained . In Section 5 we draw conclusions and future works. At the end , in the Appendix Code of the Project.

Chapter 2

Literature review and theoretical background

The Fig-2 shows a general structure for crowd detection system. The system takes image or video as input and extracts features by using proposed model. Then these features are used to perform crowd counting. The crowd counting problem can be broadly divided into following three types:

1.Detection based approaches

2.Regression based approaches

2.Density based approaches

The detection-based approaches focuses on detecting the number of human heads or faces or shoulders for crowd counting. But these methods do not perform well in large crowd scenes. The regression based approaches focuses on extracting either global or local features from the

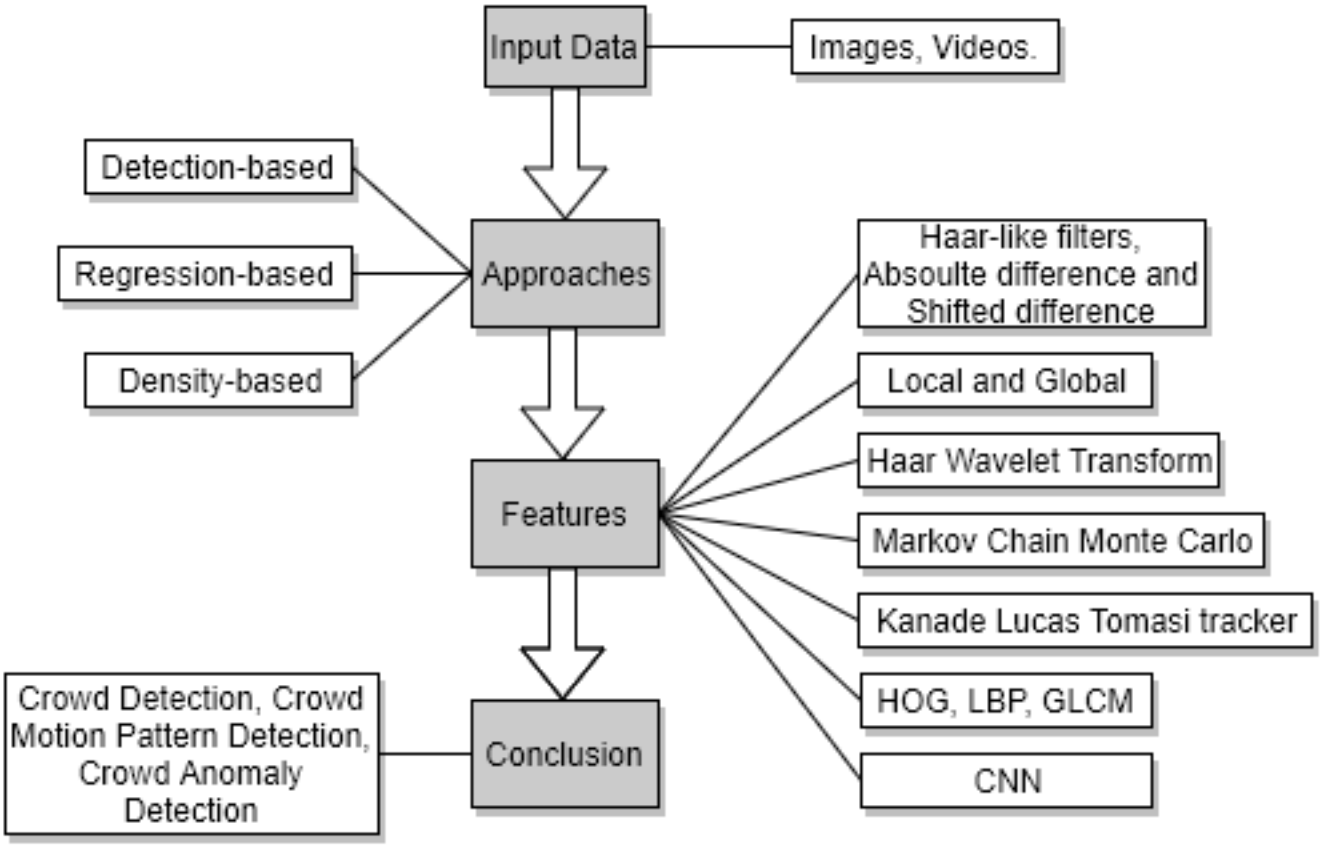

Figure 2.1 General Structure of Crowd Detection System [6]

image and maps to the ground-truth crowd count. The density map estimation-based (DME) approaches focuses on finding ground-truth density maps from the image which is followd by feature extraction and regression with ground-truth density maps. Recently, Kumar et' al [11] proposed a model which does two related tasks. CDE and density classification as main and auxiliary tasks respectively. The VGG-16 is used as front end and shared by two submodules. One constitutes the dilated CNN for DME and another a CNN for density classification. The auxiliary task extracts scale related information that helps to im-

prove the performance of DME. Wang et' al [19] focused on domain adaptation strategy. To address the problems author proposed two modules like data collector labeller and crowd count network. Authors created large-scale wild scenario synthetic dataset called GCC. For domain adaptation authors proposed a human free effort in data annotation for GCC. Authors proposed a SSIM embedding Cyclic GAN (SE-CGAN) that translate synthetic data into real image. Two generators are used to minimize the SSIM training loss between the original and reconstructed image. It efficiently maintain local patterns and texture information. Finally the synthetic image is translated into photorealistic images and then used to pretrain the crowd count net which is built from spatially CNN. After completion of pretrain the count model is finetuned in real scenes. Liu et'al [13] proposed ADCrowdNet contains two concatenated modules. First one is an Attention Map Generator (AMG) which detects crowd region and detects congestion degree. The second one is a multi-scale deformable CNN called as DME generates high quality density maps. The DME takes pixel-wise product between an input image and the attention map as input. Each module contains two things a front-end made up from VGG-16 and a backend made up from multiple dilated CNN. Jiang et'al [9] proposed a Trellis-Encoder Decoder network (TEDNet) for crowd counting. The main focus is to generate high quality density maps. The contributions are as follows.

First, to address large-scale variations of objects, a multi-scale encoder is proposed where feature maps at different stages are hierarchically aggregated and in the multi-path decoder, multiple decoding paths are established at feature aggregated encoding stage. Second, skip connections are established across several paths to fuse multiscale features. Third, a combinatorial loss is introduced to maintain similarities in local coherence and spatial-correlation between density maps. Zhao et' al [23] proposed a heterogeneous task for density estimation. Three attributes need to be addressed for DME are, geometric, semantic and numeric attributes addressing to scale variation, clutter background and global count respectively. The proposed auxiliary task CNN (AT-CNN) can be viewed as encoder and decoder. The encoder encodes the heterogeneous features related to three auxiliary tasks and the decoder decoded the feature map into pixelwise density values. The three auxiliary tasks are crowd segmentation to address clutter background by binarizing the density map, depth prediction addressing scale variation in which depth maps are generated and combined with the binary map to create mask. And the third one is global count representing the numeric attribute. All these three tasks are learned in the encoder phase. Zhang et' al [21] proposed a multi-view crowd counting model. The deep model not only estimates densities for each view but also for 3D world ground plane. Authors proposed three fusion strategies. The late

fusion model fuses density maps of different camera views. The early fusion model fuses feature maps of different camera views. The multi-view multi-scale early fusion model ensures the alignment of features with consistent scales to the ground plane.

Chapter 3

Methods and Models

We proposed a multi-model feature fusion using CNN and Auto-Encoders for crowd counting. We divided the entire model into three parts:

1. Using a simple Convolutional Neural Network to device a model to estimate the crowd count in a static image.

2. Use an autoencoder to encode the images in a compressed representation, so that the model could learn important latent information, and use this compressed representation to train a Multilayer Perceptron model to estimate crowd count in a static image.

3. Ensemble the two previous models (i.e. Multi Modal Feature Fusion) to form a new model where we regress over the output given by the two models and use it for the same task.

The models are described in brief below:

3.1 CNN Model

A Convolutional Neural Network is a deep learning algorithm that is used to learn the various features from an image using some number of filters in various layers using the convolution, activation functions and max pooling operations. The architecture of a Convolutional Neural Network resembles the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex.

One simple way of learning features from the image is to flatten the pixels of the image into a one-dimensional input feature vector and train a Feed-Forward neural network for the task at hand. One of the reasons why ConvNet is preferred over a simple Feed-Forward neural network like the one above is that a simple Feed-Forward neural network is unable to capture the spatial and temporal dependencies in the image making the model less accurate. In CNN's however, the neurons look for increasingly sophisticated aspects of the image like some edges of a particular orientation or a particular structure or colour which might be the key features in prediction.

The images were, at first randomly shuffled and converted to greyscale which was resized to 300 X 300 dimensions. These images were then

fed to the CNN network which included multiple alternate Conv2D, ReLU and MaxPool layers and finally a couple of densely connected layers. The training and validation loss was monitored by comparing the predicted values with the ground truths (actual ground truth of the images) using the Euclidean Distance Loss and Adam optimizer. We also added the early-stopping callback condition to monitor the validation loss.

Our CNN model is as shown in figure 3.1

3.2 Autoencoder + MLP Model

An autoencoder is a fully connected deep network where, in each step, we reduce the dimension of the input vector and get an encoded vector, after which we try to rebuild the image by upscaling using the same dimensions in reverse order and monitoring the reconstruction loss. The encoded neurons store the latent information which are key factors in our prediction. The autoencoder helps us in representing the original images in a lower-dimensional space and is capable of learning non-linear manifolds(A manifold is a continuous, non-intersecting surface). The reconstruction loss captures the closeness of the reconstructed image and the original image. For an autoencoder to work there must be

some structure in the data.

We then use this encoded vector as the input to our Multilayer Perceptron model which is a densely connected Feed-Forward neural network. In this way, the model is able to capture some of the useful latent information which might not be captured by our CNN network.

(We used greyscale images which were resized to 300 X 300 dimensions.)

The images are first flattened and fed to the autoencoder, which learns to reproduce the image by first encoding it (we call this encoded representation the "bottleneck" layer) and then decoding this encoded representation. After training the autoencoder for 1000 epochs we extract the "bottleneck" layer from this autoencoder model, thus giving us the model to convert an image to its encoded representation and extract the latent information from the image.

After this step we obtain the encoded representations for all the images in our training, testing and validation sets and then use these encoded representations to train another Feed Forward deep neural network to estimate the crowd count.

The model of our autoencoder and Feed Forward deep neural network is as shown in figure 3.2 and 3.3

3.3 An ensemble of the above 2 models

We saw how both of the above models captured unique information in their respective models. Now the goal is to combine the features of both the models so that a better robust model can be built which takes into consideration the useful features learned using the above models. Ensembling multiple models can boost the performance manifolds. This type of model is generally more flexible and scalable relative to the diverse datasets. Ensembling multiple models helps in not only adding a bias that in turn counters in reducing the variance of the individual models but also increases the accuracy relative to the individual models.

Multiple models may be combined in various ways, the simplest being taking the average of the output of the individual models. But here we concatenate the output of the above two models and do a regression on 2 variables to get the final predicted crowd count of the images.

The first part of the model is essentially the same as the above two models (described above). We just concatenate the outputs of the models and add 1 more dense layer before we get the regression output from

the final densely connected layer. The input, however, is a list of 2 inputs, namely input for the CNN model and the encoded input for the MLP model. The loss is monitored using the Adam optimizer and mean squared error (MSE) loss.

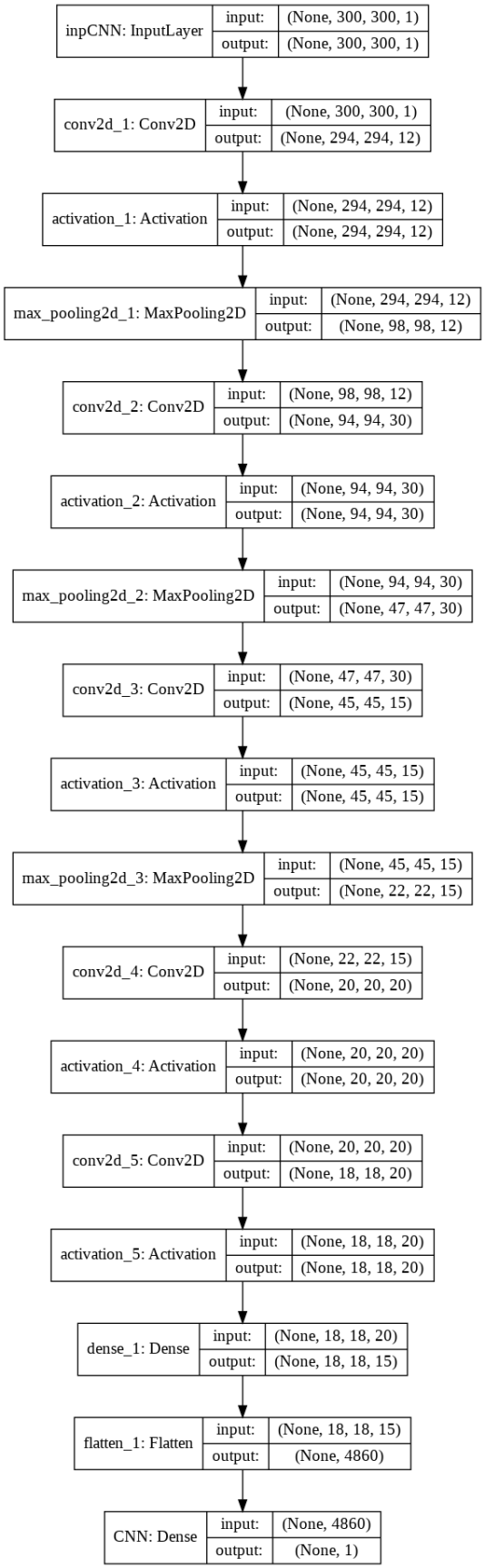

Figure 3.1 Cnn Model Architecture[source-Keras generated image]

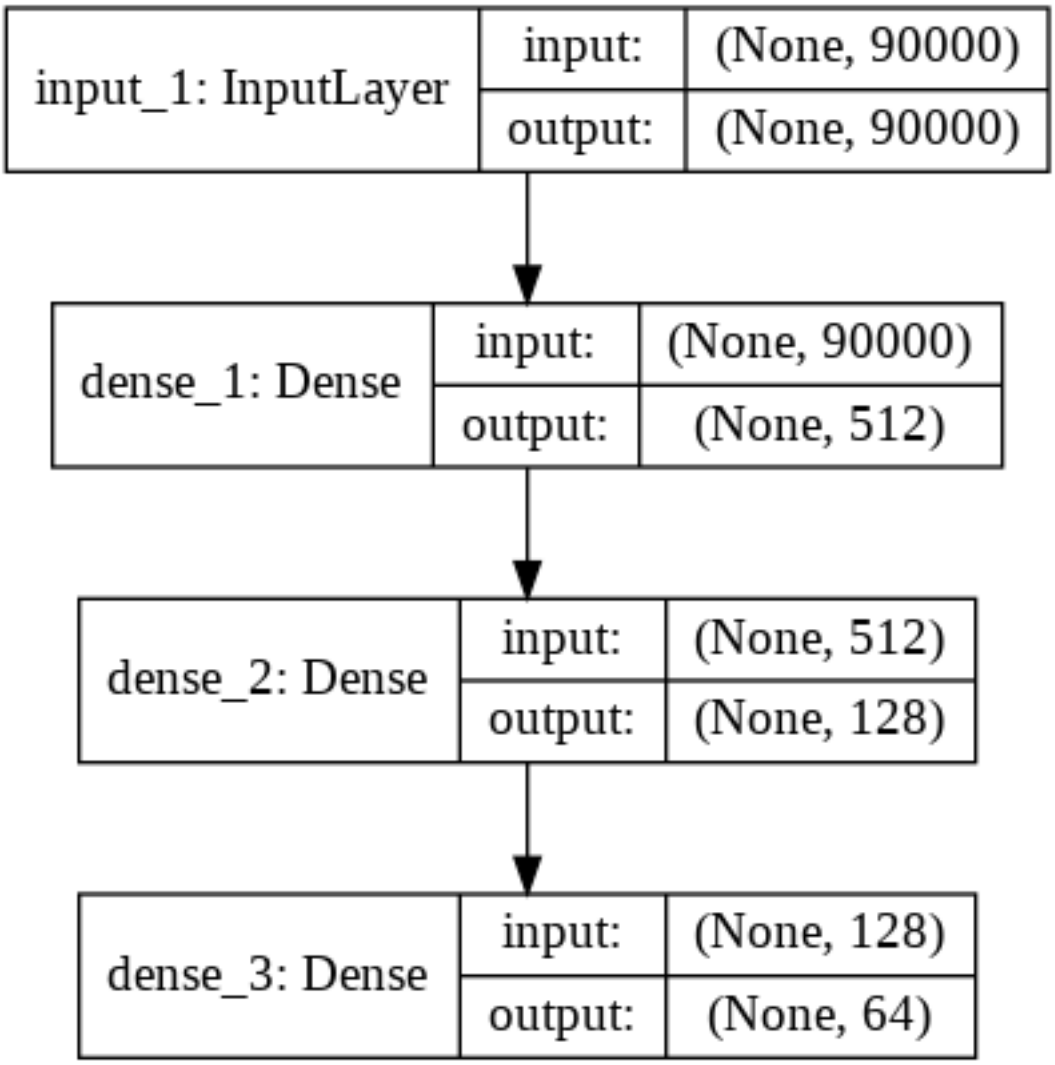

Figure 3.2 Encoded Model structure[source-Keras generated image]

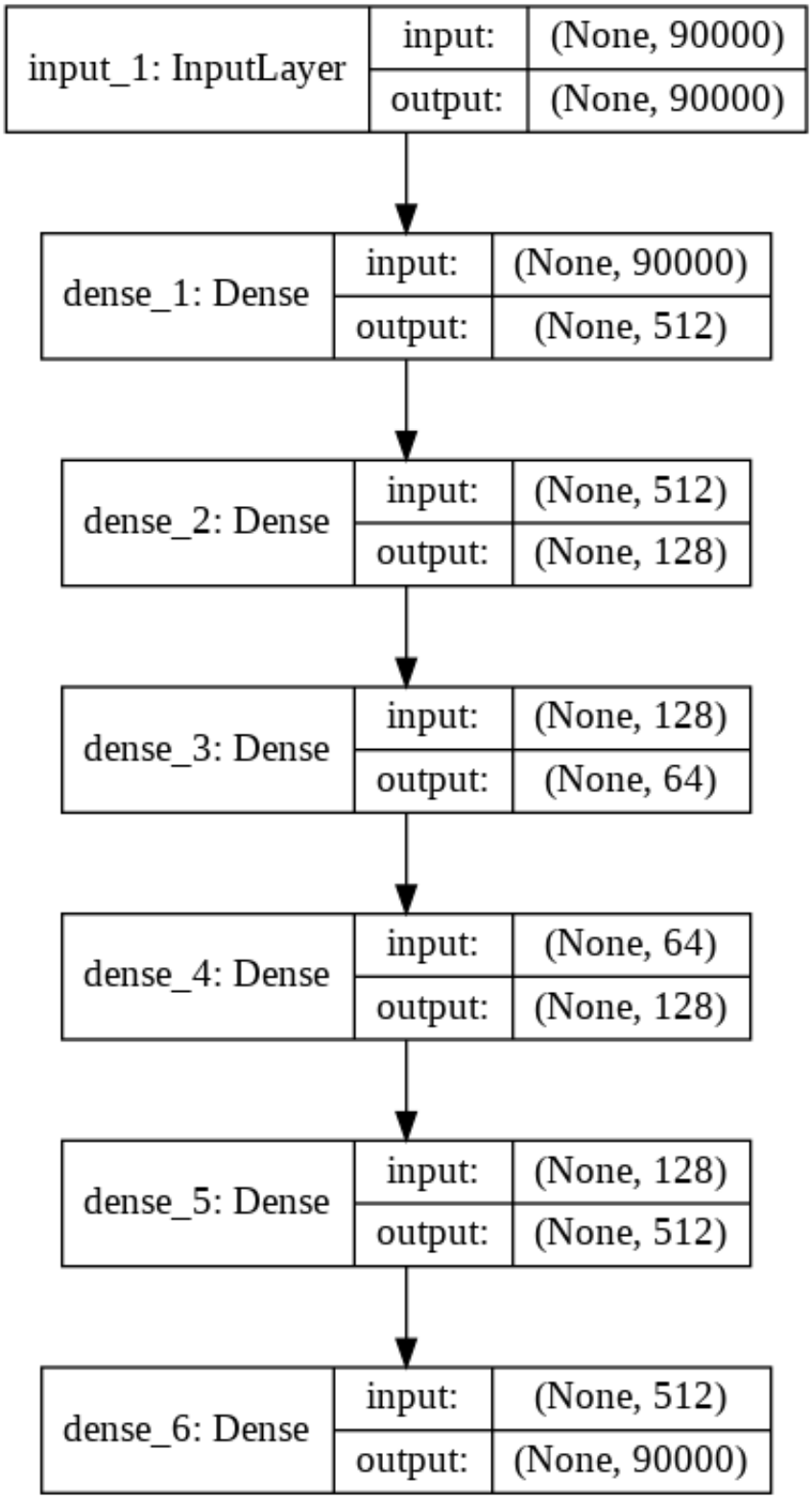

Figure 3.3 Autoencoder Model Architecture
[source-Keras generated image]

Chapter 4

Results and analysis

For experiment, we have used two publicly available datasets namely, Pets-2009[2] and UCF-CC-50[3]. We have takes scene-1 of Pets-2009. The main parameter monitored during the model training was mean squared error (MSE). After prediction, we also calculated mean absolute error (MAE) separately. We compared our results with Zhang [21] for Pets-2009 and we are getting better result for scene-1. The MAE and MSE can be calculated by using the following equation.

$MAE - 1/N * \sum_{i=1}^{N} |GCi - ECi|$]

$MSE = 1/N * \sum_{i=1}^{N} |GCi - ECi|^2$

where GCi= The ground truth of crowd count for ith image, ECi=Estimated Count using the proposed model for ith image., GM= Ground truth density map.

The precise data from the resuls obtained on PETS-2009 dataset:

MODEL	MAE	MSE
CNN	1.55885251	2.28859554
Autoencoder+MLP	0.44475882	0.57255282
Ensemble	0.43970564	0.51677697
Zhang [18]	1.66	

Result of five fold cross validation on augmented UCF dataset using autoencoder and MLP model

In the UCF dataset, we had 50 images representing dense crowd and various orientations, obstacles etc. We first performed data augmentation to create some more training images as 50 images cannot do the job, by performing operations like rotation, width shift, height shift, zoom, brightness change etc. and generated 1000 images which were divided into 5 sets to perform five-fold cross-validation. Then five iterations are performed and in each iteration we took 200 images for validation from the i'th set and the remaining 800 images for training, saving the model after each iteration.

We were unable to get good results because of only 50 unique images in the dataset and were unable to learn the features of such a dense crowd from such a limited dataset. Although we tried data augmentation to increase the training set, it could not increase the number of important features learned for crowd count in such a dense image. Moreover, we had to split the limited data into training validation and testing sets, thus limiting further the flexibility of the model.

The precise data from trained models has been listed below

FOLD	MAE	MSE
First	188.05045	94913.52
Second	184.90318	100127.59
Third	181.49823	85586.43
Fourth	167.44281	56425.895
Fifth	183.69685	85107.59

Chapter 5

Conclusions and Discussion

Crowd count estimation methods are one of the challenging problems of Computer vision and deep learning.

There are three approaches such as Detection-based approach, Regression-based approach, Density-based approach. The deep learning model is very efficient for crowd counting and analysis in which Convolutional Neural Network is our basic framework to learn efficient features for counting. In our model we ensemble CNN and autoencoder model because Ensembling multiple models can boost the performance manifolds. This type of model is generally more flexible and scalable relative to the diverse datasets. Ensembling multiple models helps in not only adding a bias that in turn counters in reducing the variance of the individual models but also increases the accuracy relative to the single models. To get better results model requires ensembling.

Future Directions

Model that we have implemented gives good results on still images. But to control problems related to crowd, model requires to learn not only from still images but also from real time videos. But due to lock-down and limited resources we were unable to implement that. Thus in future, if we got opportunity we would like to extend our model to crowd count in real time videos . We would also like to implement our idea on more complex and real world datasets and adding support to crowd sentiments analysis.

Chapter 6

Datasets

1.Pets-2009[2]

2.UCF-CC-50[3]

Bibliography

[1] List of human stampedes.

[2] Pets 2009.

[3] Ucf-cc-50.

[4] Garima Ahuja and Kamalakar Karlapalem. Crowd congestion and stampede management through multi robotic agents. *arXiv preprint arXiv:1503.00071*, 2015.

[5] Robert A Beezer, T Hastie, R Tibshirani, and J Friedman Springer. The elements of statistical learning: Data mining, inference and prediction. by. 2002.

[6] Mayur Chaudhari and Archana Ghotkar. A study on crowd detection and density analysis for safety control. *International Journal of Computer Sciences and Engineering*, 6:424–428, 04 2018.

[7] Yaocong Hu, Huan Chang, Fudong Nian, Yan Wang, and Teng Li. Dense crowd counting from still images with convolutional

neural networks. *Journal of Visual Communication and Image Representation*, 38:530–539, 2016.

[8] Lida Huang, Tao Chen, Yan Wang, and Hongyong Yuan. Congestion detection of pedestrians using the velocity entropy: A case study of love parade 2010 disaster. *Physica A: Statistical Mechanics and its Applications*, 440:200–209, 2015.

[9] Xiaolong Jiang, Zehao Xiao, Baochang Zhang, Xiantong Zhen, Xianbin Cao, David Doermann, and Ling Shao. Crowd counting and density estimation by trellis encoder-decoder networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6133–6142, 2019.

[10] Mark A Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

[11] Abhay Kumar, Nishant Jain, Suraj Tripathi, Chirag Singh, and Kamal Krishna. Mtcnet: Multi-task learning paradigm for crowd count estimation. *arXiv preprint arXiv:1908.08652*, 2019.

[12] Lazaros Lazaridis, Anastasios Dimou, and Petros Daras. Abnormal behavior detection in crowded scenes using density heatmaps and optical flow. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 2060–2064. IEEE, 2018.

[13] Ning Liu, Yongchao Long, Changqing Zou, Qun Niu, Li Pan, and Hefeng Wu. Adcrowdnet: An attention-injective deformable convolutional network for crowd understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3225–3234, 2019.

[14] Qian Liu, Anna Junker, Kazuhiro Murakami, and Pingzhao Hu. Automated counting of cancer cells by ensembling deep features. *Cells*, 8(9):1019, 2019.

[15] T Nathan Mundhenk, Goran Konjevod, Wesam A Sakla, and Kofi Boakye. A large contextual dataset for classification, detection and counting of cars with deep learning. In *European Conference on Computer Vision*, pages 785–800. Springer, 2016.

[16] Chong Shang, Haizhou Ai, and Bo Bai. End-to-end crowd counting via joint learning local and global count. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1215–1219. IEEE, 2016.

[17] MV Valueva, NN Nagornov, PA Lyakhov, GV Valuev, and NI Chervyakov. Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Mathematics and Computers in Simulation*, 2020.

[18] Chuan Wang, Hua Zhang, Liang Yang, Si Liu, and Xiaochun Cao. Deep people counting in extremely dense crowds. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1299–1302, 2015.

[19] Qi Wang, Junyu Gao, Wei Lin, and Yuan Yuan. Learning from synthetic data for crowd counting in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8198–8207, 2019.

[20] Biao Yang, Weiqin Zhan, Nan Wang, Xiaofeng Liu, and Jidong Lv. Counting crowds using a scale-distribution-aware network and adaptive human-shaped kernel. *Neurocomputing*, 2019.

[21] Qi Zhang and Antoni B Chan. Wide-area crowd counting via ground-plane density maps and multi-view fusion cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8297–8306, 2019.

[22] Youmei Zhang, Faliang Chang, Mengdi Wang, Fulei Zhang, and Chao Han. Auxiliary learning for crowd counting via count-net. *Neurocomputing*, 273:190–198, 2018.

[23] Muming Zhao, Jian Zhang, Chongyang Zhang, and Wenjun Zhang. Leveraging heterogeneous auxiliary tasks to assist crowd counting.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12736–12745, 2019.

[24] Han Zou, Yuxun Zhou, Jianfei Yang, and Costas J Spanos. Device-free occupancy detection and crowd counting in smart buildings with wifi-enabled iot. *Energy and Buildings*, 174:309–322, 2018.

Appendix A

Below contains Code links of our Crowd count model(To view Click on them):

1. CNN Model
2. Autoencoder Model
3. Ensemble Model