TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS

THESIS NO: 074MSICE019

Crowd Visualization and Counting by Smooth Dilated Convolutional Network

by
Sujan Khadka

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION AND COMMUNICATION ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
LALITPUR, NEPAL

September, 2021

Crowd Visualization and Counting by Smooth Dilated Convolutional Network

by

Sujan Khadka

074MSICE019

Thesis Supervisor

Prof. Dr. Subarna Shakya

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science in Information and Communication Engineering

Department of Electronics and Computer Engineering

Institute of Engineering, Pulchowk Campus

Tribhuvan University

Lalitpur, Nepal

September, 2021

COPYRIGHT ©

The author has agreed that the library, Department of Electronics and Computer Engineering, Institute of Engineering, Pulchowk Campus, may make this thesis freely available for inspection. Moreover, the author has agreed that the permission for extensive copying of this thesis work for scholarly purpose may be granted by the professor(s), who supervised the thesis work recorded herein or, in their absence, by the Head of the Department, wherein this thesis was done. It is understood that the recognition will be given to the author of this thesis and to the Department of Electronics and Computer Engineering, Pulchowk Campus in any use of the material of this thesis. Copying of publication or other use of this thesis for financial gain without approval of the Department of Electronics and Computer Engineering, Institute of Engineering, Pulchowk Campus and author's written permission is prohibited.

Request for permission to copy or to make any use of the material in this thesis in whole or part should be addressed to:

Head
Department of Electronics and Computer Engineering
Institute of Engineering, Pulchowk Campus
Pulchowk, Lalitpur, Nepal

DECLARATION

I declare that the work hereby submitted for Masters of Science in Information and Communication Engineering (MSICE) at IOE, Pulchowk Campus entitled “**Crowd Visualization and Counting by Smooth Dilated Convolutional Network**” is my own work and has not been previously submitted by me at any university for any academic award.

I authorize IOE, Pulchowk Campus to lend this thesis to other institution or individuals for the purpose of scholarly research.

Sujan Khadka
074MSICE019
September, 2021

RECOMMENDATION

The undersigned certify that they have read and recommended to the Department of Electronics and Computer Engineering for acceptance, a thesis entitled "**Crowd Visualization and Counting by Smooth Dilated Convolutional Network**" submitted by **Sujan Khadka** in partial fulfillment of the requirement for the award of the degree of "**Master of Science in Information and Communication Engineering**".

……………………………

Supervisor: Subarna Shakya, PhD
Professor,
Department of Electronics and Computer Engineering,
Institute of Engineering, Pulchowk Campus

……………………………

External Examiner: Subhash Dhakal
IT Director,
Department of National Id and Civil Registration,
Government of Nepal

……………………………

Committee Chairperson: Basanta Joshi, PhD
Program Coordinator, MSc in Information and Communication Engineering,
Department of Electronics and Computer Engineering,
Institute of Engineering, Pulchowk Campus

Date: September, 2021

DEPARTMENTAL ACCEPTANCE

The thesis entitled " **Crowd Visualization and Counting by Smooth Dilated Convolutional Network** ", submitted by **Sujan Khadka** in partial fulfillment of the requirement for the award of the degree of "**Master of Science in Information and Communication Engineering**" has been accepted as a bona fide record of work independently carried out by him in the department.

………………………..

Prof. Dr. Ram Krishna Maharjan
Head of the Department
Department of Electronics and Computer Engineering,
Pulchowk Campus,
Institute of Engineering,
Tribhuvan University,
Nepal.

ACKNOWLEDGEMENT

I would like to express my deep gratitude to **Department of Electronics and Computer Engineering** for incorporating research work as a part of our syllabus for realization of our knowledge in the real-world applications.

I am very grateful and pay my sincere gratitude to my supervisor **Professor Dr. Subarna Shakya** for his invaluable guidance, encouragement, support and every bit of help and providing me with an extraordinary experience during this research work. I am also hopeful of his continuous guidance and support in the days to come.

I am very grateful to **Dr. Ramkrishna Maharjan**, HoD of Electronics and Computer Engineering, Pulchowk Campus for his dedicated support. I would like to thank **Dr. Basanta Joshi**, coordinator of MSc. Information and Communication Engineering, Pulchowk Campus, for his constant effort and arrangement on thesis works.

Also, I would like to thank all the faculties at Department of Electronics and Computer Engineering, Institute of Engineering, Pulchowk Campus for their guidance and support in my learning process. Similarly, I am thankful to my family and friends for their support in this work.

Sujan Khadka

074MSICE019

ABSTRACT

With mass urbanization, culturally diversified country with many cultural and religious gatherings happening time and again, the need of crowd estimation from single image and information on the distribution of crowd in the same is deemed necessary. The old-fashioned way of keeping records, sensor-based counting fails when the crowd movement is dynamic and/or random. Task is challenging due to geometric distortion, perspective distortion, severe occlusion, illumination condition in the image. The forementioned challenges has been addressed by Deep Learning Convolutional Neural Network where in CNN is employed as a feature extractor and Smoothed Dilated CNN is used in the backend, that facilitates aggregation of multi-scale contextual information by increasing the receptive field with same resolution removing the gridding artifacts. Model is end-to-end trainable since it employs pure convolutional structure and can accept arbitrary size and resolution of input image for conversion into density map which is used for crowd counting. Training of the model begins with the generation of ground-truth density map which is computed based on geometry-adaptive kernel to account for perspective effect on the denser crowd and fixed kernel on the sparse crowd. ShanghaiTech dataset is been used which comprises of 1198 tagged images with a total amount of 330,165 persons. Comparison between dilation rate 2 and 4 for both Part_A and Part_B of ShanghaiTech dataset is made. Upon evaluation of the model with the Csrnet where smoothing of the dilated convolution is not implemented, the counting accuracy and quality of the density map for both Part_A and Part_B of the dataset has been significantly increased.

Keywords: crowd counting, density map, atrous convolution, smoothed dilated CNN

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	: Artificial Intelligence
ANN	: Artificial Neural Network
CNN	: Convolutional Neural Network
DCNN	: Dilated Convolutional Neural Network
FC	: Fully Connected
HOG	: Histogram of Oriented Gradients
MAE	: Mean Absolute Error
MESA	: Maximum Excess over Sub-Arrays
MSE	: Mean Squared Error
PSNR	: Peak Signal to Noise Ratio
SIFT	: Scale Invariant Feature Transform
SPP	: Spatial Pyramid Pooling
SS	: Separable and Shared
SSIM	: Structural Similarity Index
VGG	: Visual Geometry Group
VOC	: Visual Object Classes

1. INTRODUCTION

1.1 Background

With the increased population and urbanization, the information on the number of people attending an event, political rallies, protest, festival celebration, sports events, musical concerts etc. helps in a great deal. In most occasion, we need to monitor the public places and make sure it is not crowded or the density of people in public places is sparse so as to maintain public safety. The density of population waiting for a bus, metro, aircraft or subway train helps to manage our public transportation hub. In the history of mankind, several people have lost their lives in massive stampede at public events. Similarly, there could be instances when the behavior of people in the crowd needs to be studied. This all needs efficient crowd counting and visualizing mechanism.

From the beginning of mankind, attempts are continuously being made to reduce human labor leading numerous inventions. Manually counting the total number of people for every image is impossible because the world is moving towards automation. Moreover, manual counting bars the use in Real Time system of crowd counting and fails when the crowd is random and dynamic. Crowd counting can be taken as Computer Vision application and it has a great scope because of the practical usage in real-world applications as in crowd analysis, traffic crowd, urban management, public surveillance, queue management, monitoring and dispersing dense crowd to ensure public safety, etc. It helps mitigate possible accidents in the public areas, actively monitor the crowd size in public events like rallies and sports, departure-arrival time-table of public transportation. It thus aids traffic control and in private surveillance in firms. It can further be extended for vehicle counting system to build a smart traffic light system on a junction or highway. It can be used to analyze the animal migration, thus aiding wildlife study. More application like this requires the counting of commuters or herd, however, manually counting particular items from any image is a tedious task. Severe occlusion, geometric distortion, illumination condition and overlapping region imposes challenge on crowd counting. Moreover, the density and distribution of crowd also impose challenge on the task.

Digital images are made up of image pixels that are ordered in rectangular array. This rectangular array of pixels with individual pixel values do not carry meaningful information by themselves. Human beings picture the image as a whole and gain insight from it which is an easy task for humans. However, to understand object from the numerical data in the pixel is troublesome for computers. Similar to every kind of images, if we have an image of crowd and counting the number of objects present in the image requires understanding the image by the computer for which the image features need to be identified. This system uses Deep Learning technique for “crowd counting by density map estimation”. The density map is first generated and then estimate crowd count by deploying a Deep Learning CNN, specifically, CNN as a feature extractor and smoothed dilated CNN at the backend to enlarge the receptive field. The flexible architecture of CNN aids in concatenation of backend for the density map generation. Architecture uses the VGG-16 classifier front-end because of its numerous advantages. To mitigate the gridding artifacts introduced by the dilated CNN, smoothing is done by extracting deeper information of saliency. Dilated convolution/*atrous* convolution, is widely used method that increase the size of the receptive field but doesn’t increase the kernel size. In this way, the number of required parameters is vastly decreased making the model lighter and efficient in terms of computation; however, introduces gridding artifacts. Taking into account the gridding artifacts, smoothing block is added to the system.

1.2 Problem Statement

Crowd counting begins with the feature extraction from the image. Had it been a prefect picture with no distortion, perfect lighting, no shades and no overlaps then feature extraction would not be much of a concern. However, severe occlusion, geometric distortion, illumination condition and overlapping region imposes challenge on crowd counting. Therefore, crowd counting model needs to take in account the forementioned imperfections in the image and accurately model the problem with an effective feature extracting and generalizing capability. Unlike normal image processing, crowd counting requires dense prediction by capturing multiscale information. To address this issue a different kind of CNN, called the Dilated CNN is used that uses sparse kernel by interpolating with zeros so as to alternate the pooling and convolution layer. However, the DCNN introduces gridding artifacts because in the process of getting wider receptive field of view, wherein the dilation rate > 1, for computation of the output, adjacent units are computed from entirely separate set of units in the input. The local information becomes inconsistent and such information can become irrelevant across larger distance. Smoothing the dilated CNN mitigate gridding artifacts. SS convolution layer will be added before the dilated layer. SS convolution adds the dependencies among the nearby units producing smoothed feature maps. This helps preserve the local information and improve dense prediction. Even though we add a new layer, particularly SS convolution, only a few hundred parameters will be added. [14]

1.3 Objectives

The main objectives of this thesis work are as follows:

- To estimate the number of people in a crowd image
- To visualize crowd image by generating high-quality density map

1.4 Scope of the work

This work is the design of an automatic crowd visualization and counting system that can be used for public surveillance. It will help the respective authorities to monitor the public places and take necessary action so as to minimize accident or avoid massive stampede or disperse crowd in the crowded areas by knowing the crowd density. It can as well be beneficial in disaster management and emergency evacuation in case of fire outbreaks and calamitous events. Crowd analysis, queue management, traffic control, etc. can be its application and the applicable areas are almost every public area because of urbanization.

2. LITERATURE REVIEW

Crowd counting is one of the hot topics in the few succeeding years as a number of research has been going on. It has a great scope in computer vision with real world applications as in crowd analysis, traffic crowd, public surveillance, queue management and ensuring public safety, etc. The method employed for crowd counting has been changing with the availability of technology and newer algorithm.

Earlier crowd counting mechanism relied on counting by detection.[1] This method of detecting individual head and counting suffered in a very crowded scene wherein occlusion posed difficulty. Counting started with detection-based algorithm using moving-window-like detector. It first detects people and the counts them [2]. These methods employ well-trained classifiers like Haar wavelets [3] and HOG [4] to extract low-level features. Detection-based approach can be sub-divided into monolithic detection, part-based detection and multi-sensor detection. Monolithic detection performs satisfactorily only in sparse crowd. Crowded scene with occlusion and clutter can’t be addressed by this approach. Part-based detection aims to provide solution that can tackle partial occlusion. Multi-sensor-based detection captures image from more than one source placed at different geometric positions so as to handle the partial occlusion and region of overlaps. However, in a highly dense image with severe occlusion, distortion and varied illumination condition, where most of the targeted objects are concealed, these models perform poorly.

Direct regression-based method overcomes the problem of occlusion and high background clutter. Direct regression-based method usually crops the image patches first and generates features from them. Low-level features were generated from foreground and texture features. The relations among extracted features are learnt to calculate the number of particular objects. Regarding the target of regression, it could be the object counts or the object density. Basically, it has two components; low level feature extraction and regression modelling. Idrees et al. [5] introduced Fourier analysis and SIFT in feature extraction. Crowd density is then regressed from interest-point based counting, head detection and frequency analysis. Direct regression-based approach does not take into account the spatial distribution of the crowd i.e., saliency. Instead of tedious detection and object localization for counting, Lempitsky et al. [8] estimated an image density. Count of image in a specific region is achieved by integrating over the image region. It employs MESA distance as the distance metric and linear mapping of local patch features. Regression is then used to obtain corresponding density map. Comparing to two approaches object counting and object density estimation, the object density provides more insight as it maintains the object location. Object density, rather than just counting objects, provides a general approximation of their positions, which aids in understanding crowd behavior. Pham et al. [6], realizing difficulty in linear mapping to map local patch features, employed non-linear mapping and random forest regression for density map.

Density map approach uses density map as the intermediate representation of the input image, followed by summing up over a region to obtain the count. Unlike the direct regression-based estimation which does not consider the spatial distribution of the crowd, density map generation achieves better performance. The density map estimation methods need density map generation which is done either from dot-map annotation or deep learning models. The most popular method is by convolving the head annotated dot-map with a Gaussian Kernel [7].

In recent years deep learning has evolved with the digital era. Tasks related to computer vision, AI, natural language processing, etc. are dominated by Deep Learning. Walch and Wolf [9] estimated density map directly from input image and provided improvement in terms of layered boosting and selective sampling. Multiple network scales image variedly; trains the model and concatenates the output of multiple networks to obtain the final density map. Karianakis et al. [10] proposed hybrid method based on boosting. CNN extracts low-level features based on which the object candidates are determined. It is followed by AdaBoost to build a final classifier. In [11], CNN is used to count the dense crowd directly without estimating the density function. Previous state-of-the-art technique multi-column CNN (MCNN) [20] is based on the multiscale architecture however the network is very deep and require large amount of model training time. Later experiment showed that its branch structure is not significantly effective. Babu et. al. [21] proposed a switching CNN architecture based on patch to address the variation in local crowd density in a scene. Three different independent regressor network having varied receptive field is used and each patch is sent to one of these three specified networks to compute density map. Fisher Yu et al. [24] developed a newer approach in convolutional network for dense prediction. They proposed dilated CNN to aggregate multi-scale information systematically. This was done without losing the resolution that too by expansion of receptive field in exponential scale. This designed network increased the accuracy in dense prediction, when incorporated to existing systems for semantic segmentation. It paved the path of semantic segmentation differently with that of image classification.

Yuhong et al. [13] employs CNN for extracting features from the input image and backend architecture that generated density map. Architecture uses the VGG-16 layers in the front-end that serves as the density level classifier. Dilated CNN is employed in the backend architecture that has larger receptive field to capture multiscale information without significant increase in the parameters. It is end-to-end trainable model. Dilated convolution can be used in pixel-to-pixel prediction task for images providing wider application in the areas of semantic image segmentation, machine translation, video detection and audio generation [14]. Dilated filters insert the holes in the process of enlarging the receptive field. [15] However, the DCNN introduces gridding artifacts because in the process of getting wider receptive field of view, adjacent units are calculated from completely different set of input units. This brings inconsistency of local information and the information can get irrelevant across larger distance. In recent approaches, methods have been developed to remove gridding artifacts by introducing stacked dilated CNN. In [16], [17] additional blocks of dilated CNN were added to avoid the gridding problem. But this method introduced millions of extra training parameters which ultimately makes the model inefficient in terms of computation. Wang and Ji [14], proposed two different approaches namely Group Interaction Layer and Separable and Shared Convolution for smoothing. Unlike introducing block of stacked dilated convolution layers, this approach required negligible number of extra parameters to be learnt. This helps make the model lighter. Yasarla et. al. [25] proposed multi-stream architecture for pixelwise semantic segmentation which employed the use of smoothed dilated convolution for removing griding artifacts. Chollet [26] changed the popular model of Inception to “extreme” Inception called Xception where pointwise convolution and depth-wise convolution is used so that the cross-channel correlation and the spatial correlation are separated. This led to improved performance.

Image quality assessment is done by measuring the quality of the generated output. The quality of image with respect to Human Visual System and by computing one of the most popular metric Peak Signal-to-Noise Ratio (PSNR) is contradicting. MSE and PSNR although simple to calculate in terms of mathematics do not quite relate to visual quality. The pixels of every highly structured natural image have strong dependencies among themselves. Zhou et. al. [30] proposes three components namely: luminance, contrast and structure to combinedly yield SSIM measure for image quality assessment. Hang et. al [28] used SSIM and MS-SSIM for performance evaluation for Image Restoration. Cao et. al. [29] used a Euclidean Loss and SSIM Loss function to account for the local correlation in the density map. The model used Inception architecture as the encoder and set of convolutions and transposed convolution as the decoder. The use of Local Pattern Consistency Loss helped enhance the performance of the model.

3. RESEARCH METHODOLOGY

3.1 Convolutional Neural Network

CNN belongs to the class of ANN that captures features form the image, from basic to high level features at the deeper level. Feature extraction is achieved, when CNN convolves the input data with a kernel producing a transformed feature map. Here, kernel is also referred as filter. In ML, weights are learnable parameters and the kernel weight are modified so as to extract features from the image. CNN is widely used and the most popular, since it adjusts weights automatically to find the most promising feature. [18]

Typical CNN consists of input layer, output layer, and single or stacked hidden layers that consists of a number of neurons. Arrangement of neurons is in 3-D (width, height, and depth) and the input volume is transformed to an output volume. Apparently, hidden layers are the ones where most of the magic happens. They are combination of convolution, normalization, pooling and fully connected layers. CNNs use multiple convolution layers to transform input volumes to higher levels of abstraction. CNN is widely popular since it preserves spatial information. Figure 1 shows the typical CNN model.

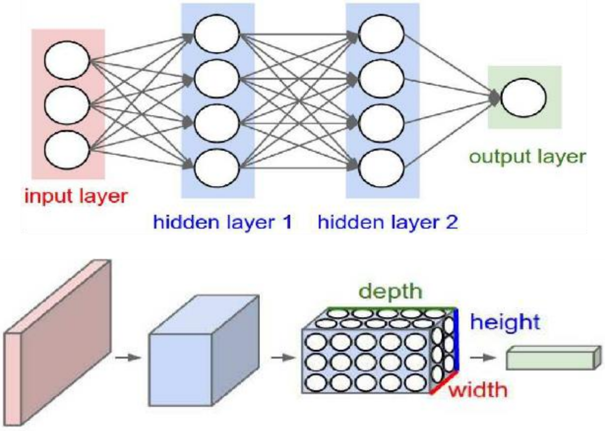


Figure 1: Convolution Neural Network

The convolution operation requires operands to have same dimensionality. In CNN, convolution operation is the sum of the dot product between input and filter. To access over the full spatial dimension of input, kernel is stride using the sliding window technique. This results in the convolved features. Most real-world problems are highly complex and non-liner that is why, we need non-linear activation in the neural network. In fact, a feed forward NN with any number of hidden layers but just the linear activation is same as the linear NN with no hidden layer. So, to add non-linearity in the model, the convolved features are passed through the activation function.

2D convolutional operation is denoted by Equation 1 and Equation 2.

$$(m * n)(t) = \int_{-\infty}^{\infty} m(\tau)\, n(t-\tau) d\tau \tag{1}$$

$$(m * n)(t) = \int_{-\infty}^{\infty} m(t-\tau)\, n(\tau) d\tau \tag{2}$$

Figure 2 helps to visualize the feature map generated from CNN. With an input image in Pascal VOC, it represents the feature map of conv$_5$ filter. The arrow represents the strongest response and their corresponding position in the image. Green rectangle indicates receptive field with strongest response.

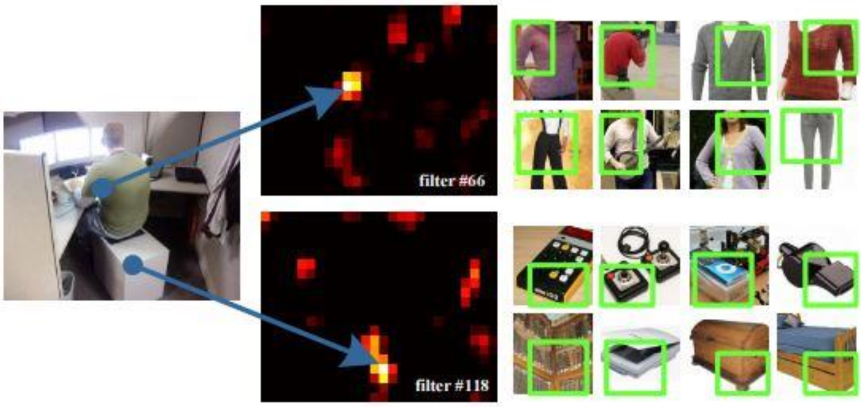


Figure 2: Feature map of CNN

CNN provides generalization which is not achieved with linear mapping. Activation Layer squashes the value into a range to bring non-linearity in the output. Nonlinearity makes the training faster and more accurate. Activation determines which node to fire among a bunch of different nodes. TanH, Rectified Linear Unit (ReLU), ArcTan, Sigmoid, Exponential Linear Unit (ELU), etc. are some of the most used activation functions.

Convolution Layer is followed by the Pooling Layer. In order to avoid overfitting, pooling layer is used to summarize the feature map. Summarized, in the sense that it does not precisely position the features from CNN, rather forwards a summarized features for further operation. The first step in pooling is to partition the input image into subregions such that there is a set of non-overlapping rectangles. Then for each of the subregion, pooling operation summarizes a value. Spatial resolution is thus reduced by pooling layer thereby reduction in the parameters count and avoidance of overfitting. It is essential to note that pooling reduces the spatial dimension but the channel dimension/depth dimension is unchanged. The two of the most widely used layers are:

max and average-pooling. The maximum value of the sub-region is output for further operation by max-pooling; whereas the average value of the sub-region is output by average-pooling.

The extracted feature map is pooled; followed by flattening that transform the entire pooled feature matrix to a single column. Here, spatial dimension of input is collapsed to channel dimension. The output is called a feature matrix which is one-dimensional array. This helps the succeeding fully connected layer to fully process the pooled feature map. Fully Connected Layer is somewhat similar to the hidden layer in ANN and is used to optimize the objective (like class scores). FC layer works upon flattened input. Consider a classification task, where we get the predicted class in output layer. Error prediction is done then backpropagated to improve the prediction. In addition, soft-max layer is used to classify an object with probabilistic values between 0 and 1.

3.2 VGG16 network

VGG16 is a CNN model used by Simonyan and Zisserman in ILSVR competition in 2014 [19]. The neurons of 16-layered VGG16 covers the receptive field with larger size. Convolution layers: 3 x 3 filter, stride = 1; same padding and max pool layer: 2 x 2 filter; stride = 2 followed by FC layers and a soft-max resulting output is the architecture. Even though it is slow to train, it has better accuracy and is easy to implement.

3.3 Depth-wise Separable and Shared convolution

If we consider 2 spatial dimension (width and height) and depth dimension (channel), standard convolution layer learns filters in a 3D space. This is to say, a single convolving kernel performs two tasks simultaneously

- Mapping cross-channel correlations
- Mapping spatial correlations

Depth-wise Separable Convolution [22] factors the same operation so that it can independently operate for cross-channel correlation and spatial correlation. i.e., separates depth and spatial dimension of a filter. This avoids convolution across all the channel thereby; few numbers of connections hence less parameter and lighter model. Reduced parameter also means computationally cheaper and reduced overfitting. In order to achieve this, we perform depth-wise convolution and then pointwise convolution. Depth-wise convolution refers to the channel-wise n

X n spatial convolution applied to a single channel at all the times unlike standard convolution. For a C channel, we have C, n X n spatial convolution. Similarly, pointwise convolution is convolution with filter 1 X 1so as to change the dimension.

Assume we have input data with the dimensions: D_f x D_f x M, where D_f x D_f is the image size and M is the number of channels. Assume there are N filters/kernels, with the dimensions: D_k x D_k x M. If we perform a standard convolution, the output size will be: D_p x D_p x N. [22,23]

Following Figure 3 illustrates standard convolution where:

Total number of multiplications required= N x D_p^2 x D_k^2 x M.

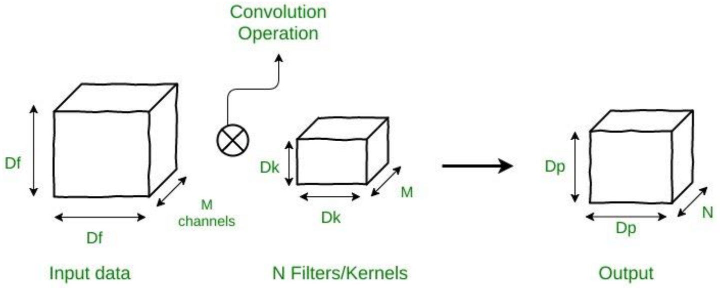


Figure 3: Standard Convolution operation

Following Figure 4 illustrates Depth-wise Separable Convolution [23] where:

Total number of multiplications required = Depth-wise multiplication + Point-wise multiplication

$$= \text{M} * D_k^2 * D_p^2 + \text{M} * D_p^2 * \text{N}$$

$$= \text{M X } D_p^2 \text{ X } (D_k^2 + \text{N})$$

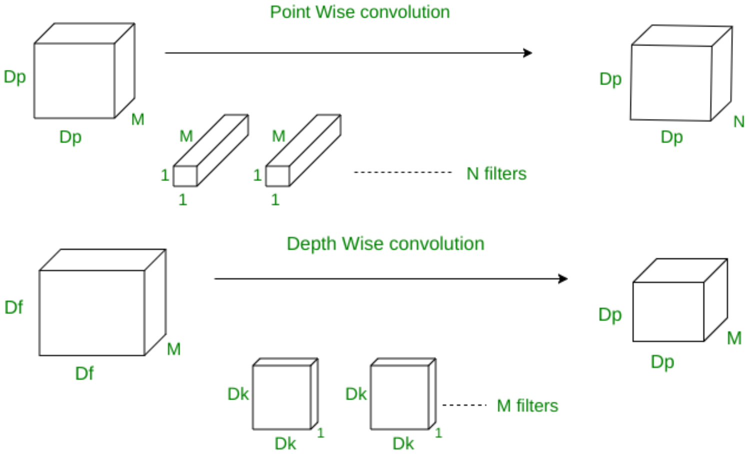


Figure 4: Depth-wise Separable Convolution

For each block in the input feature maps, SS convolution can add neighboring information. With respect to the number of connections across channels, Separable and Shared (SS) Convolution is more efficient than Standard Convolution. C^2 filters connect all C channels present in the input to all C channels in the output in a conventional convolution. For the same operation, Separable Convolution require C number of filters as it only connect the i^{th} channel in the output to the i^{th} channel in the input. SS is the one where only one filter is shared across all the channels. 'Shared' term in SS convolution means on the Depth-wise Separable convolution, same C filters are utilized by all input and output channels-pairs. Separable Convolution and SS convolution is shown in Figure 5.

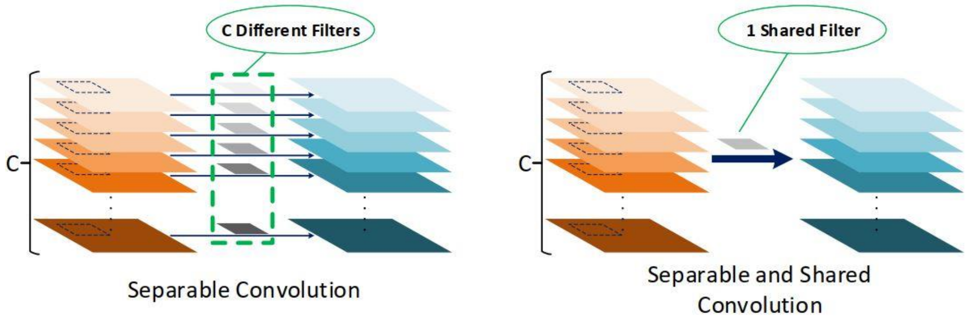


Figure 5: Separable Convolution and SS convolution

3.4 Dilated Convolutional Neural Network

DCNN, also known as *atrous* convolution, is a variation of CNN that increase the receptive field without increase in the parameters. It does so without increasing the kernel size but by filling the empty position with holes. Dilation is achieved by expanding the size of the filter. With dilated stride r, small-sized kernel k, filter is extended to [k + (k-1) (r-1)]. The distance to where the filter elements are matched in the input matrix is determined by the Dilation Coefficient D. If D = 1, it is standard convolution. Enlarged receptive field by Dilated CNN helps us in context assimilation. It provides broader view of the image hence, capturing more contextual and multi-scale information from the image. However, it doesn't change the spatial resolution. It requires less parameters, so, is computationally efficient. It is used in semantic segmentation, WaveNets (Conversion of text to audio). Figure 6 shows dilated convolution with kernel size 3 X 3, leftmost section has dilation rate 1, center has dilation rate 2 and rightmost section has dilation rate 4. It can be seen that the receptive field is enlarged by filling the empty position with holes. This clearly shows that the number of parameters does not increase significantly.[14]

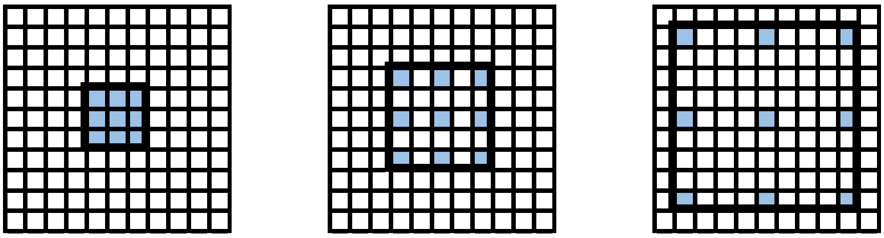


Figure 6: Dilated convolution with varied dilation rates

3.4.1 Decomposition view of DCNN

A dilated convolution operation can be illustrated by its decomposition as shown in Figure 7. Following 3 steps decomposes the operation:

Step 1: If r be the dilation rate and d be the spatial dimension, then input feature maps undergo periodical subsampling with a factor of r. Consequently, the inputs are deinterlaced to r^d separate sets of feature maps. These feature maps are of reduced resolution.

Step 2: The intermediate feature maps obtained from step 1, are fed into a standard convolution with same weighted filter as of original dilated convolution. The filter is shared for all the groups.

Step 3: After obtaining different r^d sets of feature maps, re-interlace them to the former original resolution.

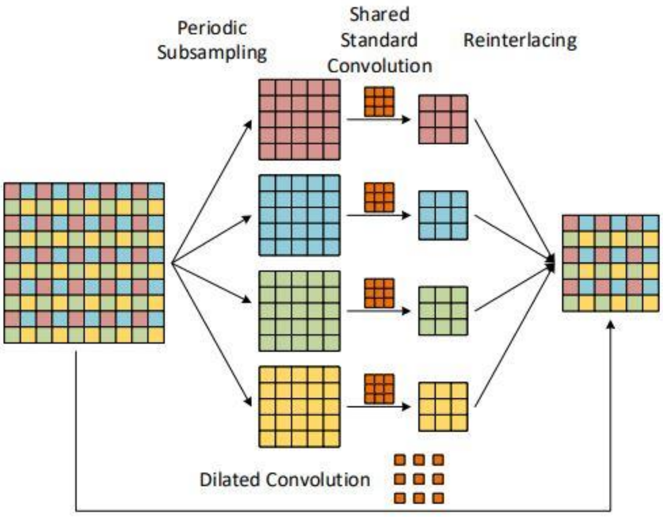


Figure 7: Decomposition View of DCNN

If we have a 10 × 10 feature map, kernel size of 3 × 3 and r = 2, dilated convolutions will generate a 6 × 6 feature map. During decomposition, periodical subsampling is done to the input feature map to produce $2^2 = 4$ sets of 5 × 5 feature maps. These 4 sets of feature maps undergo shared standard conv to produce 4 sets of 3 × 3 feature maps. Lastly, the sets of feature maps are re-interlaced that generate 6 × 6 output feature map. This generated dimension is same as the original dilated CNN. [14]

3.4.2 Smoothed DCNN

If the dilation rate > 1, gridding artifacts are produced in DCNN because adjacent units in the output are calculated from the input using entirely different set in the input units. This brings inconsistency of local information and the information can get irrelevant across larger distance.

Wang and Ji [14], smoothed the dilated convolution itself, unlike the earlier models that smoothed by introducing a block of cascaded DCNN. If we add dependencies among r^d groups of intermediate feature maps then smoothing can be achieved. Two different approaches for smoothing are provided:

- Group Interaction Layer
- Separable and Shared Convolution

The first approach is shown in Figure 8, where we add a group interaction layer before re-interlacing such that the intermediate group's dependency is established. This is equivalent to perform pixel-by-pixel fully-connected operation on the feature output using convolution or insertion of a SS block-wise FC layer after the dilated convolutional layer. In this approach, r^{2d} extra parameters are added while training. As shown in Figure 8, we need 2^{2*2} =16 connections and feature map after de-gridding operation is represented by grey color.

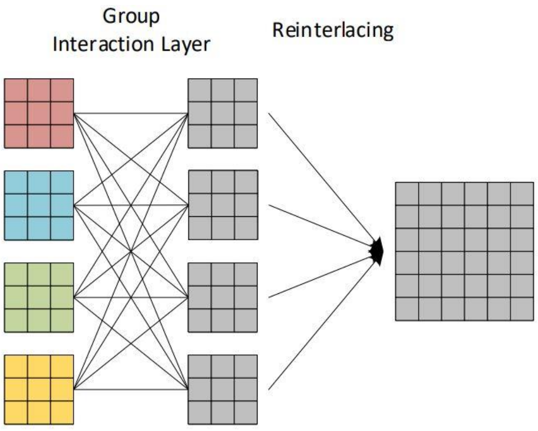


Figure 8: De-gridding method by adding Group Interaction Layer

Second approach involves insertion of a SS convolution before deinterlacing, to add dependency as shown in Figure 9. Kernel of $(2r - 1)^{d}$ size is used for SS convolution that involves sharing of the filter. i.e., same filter is used/shared by all channel pairs of input-output. In Figure 9, $(2*2-1)^{2}$ (= 3 X 3) kernel size is used for SS operation.

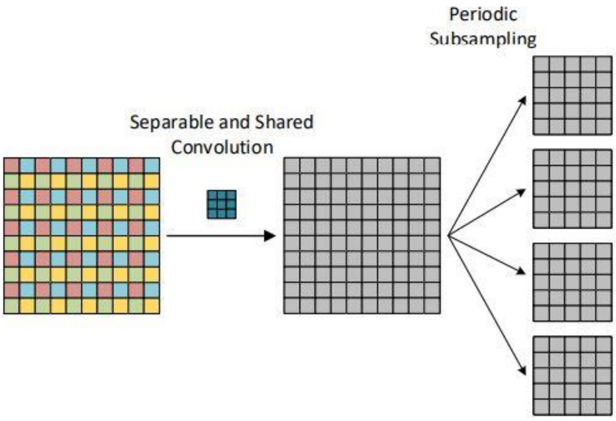


Figure 9: De-gridding method by adding SS convolutional layer

3.5 Object counting by Density Map Estimation

The trend of object counting has been changed to the density map estimation method. Deep Learning techniques uses point like annotation to remedy the problems introduced by earlier bounding box approach. With the estimation of the density map, objects can be indirectly counted. A density map is obtained by convolving with a Gaussian Kernel and normalizing it so that integrating it gives the number of objects. In the process, the number of persons per unit pixel is represented by spatial values. In the dataset, heads of the people are annotated later these head-point annotations are convolved with Gaussian Kernel to obtain density map as in Figure 10. The spatial summation in the density map gives the count.

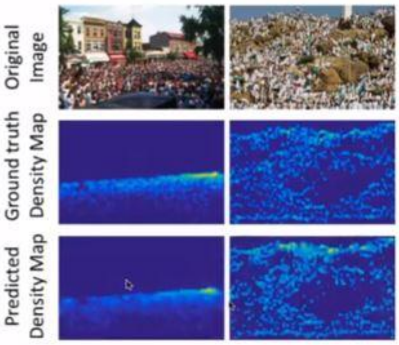


Figure 10: Ground Truth and Density Map Prediction on a crowded image

3.6 System Flowchart

The flowchart of the network is as shown in Figure 11.

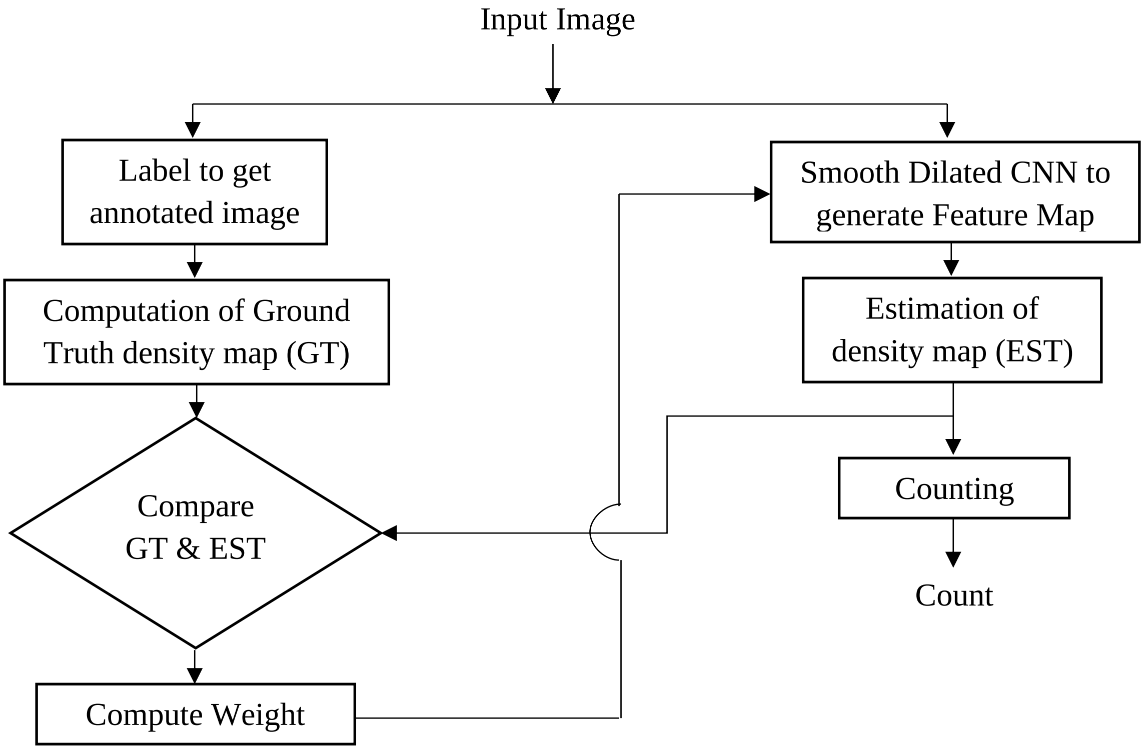

Input Image
Label to get annotated image
Smooth Dilated CNN to generate Feature Map
Computation of Ground Truth density map (GT)
Estimation of density map (EST)
Counting
Compare GT & EST
Count
Compute Weight

Figure 11: System Flowchart

3.7 System Block Diagram

Network employs CNN as a feature extractor using VGG16 fusing with the backend for the generation of density map. It contains contains two different configuration with same front end structure but different backend structure. To make the network more efficient, more number of conv layers with small kernels are employed. Since the output of the front-end is already reduced in spatial dimension, downsampling and pooling operation is replaced with the dilated CNN in the backend. This helps acquire multi-scale features from its dilated kernels. Similarly, Separable and Shared (SS) convolution technique to smooth the DCNN will be applied in order to remove the gridding artifacts. Shared filter is used in the depth-wise separable convolution.

Overall block diaram is shown in Figure 12.

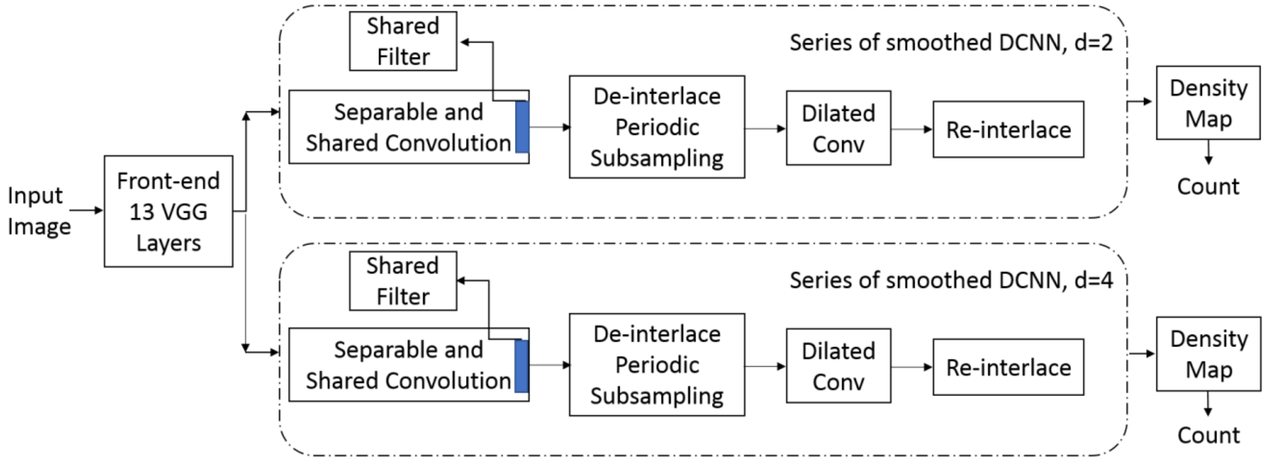

Input Image
Front-end 13 VGG Layers
Shared Filter
Separable and Shared Convolution
De-interlace Periodic Subsampling
Dilated Conv
Re-interlace
Series of smoothed DCNN, d=2
Density Map
Count
Shared Filter
Separable and Shared Convolution
De-interlace Periodic Subsampling
Dilated Conv
Re-interlace
Series of smoothed DCNN, d=4
Density Map
Count

Figure 12: System Block Diagram

Two different backend architecture with dilation rate 2 and 4 has been evaluated in order to determine the one with the better result i.e., better accuracy. Introduction of SS convolution layer with shared layer is used to remove the gridding artifacts that is introduced by the Dilated CNN. Since the dataset comprises of two parts: Part_A and Part B consisting of images with dense and sparse crowd respectively, model is developed with respect to dilation rate 2 and dilation rate 4 for both Part_A and Part B.

Overall network configuration is depicted in Table 1.

Table 1: Network Configuration

Network Configuration	
Dilation Rate = 2	Dilation Rate = 4
Arbitrary sized input color image	
conv k = 3, n = 64, d = 1	
conv k = 3, n = 64, d = 1	
Max - Pool 2 x 2, s = 2	
conv k = 3, n = 128, d = 1	
conv k = 3, n = 128, d = 1	
Max - Pool 2 x 2, s = 2	
conv k = 3, n = 256, d = 1	
conv k = 3, n = 256, d = 1	
conv k = 3, n = 256, d = 1	
Max - Pool 2 x 2, s = 2	
conv k = 3, n = 512, d = 1	
conv k = 3, n = 512, d = 1	
conv k = 3, n = 512, d = 1	
SS-conv k = 3, n = 512, p =1	SS-conv k = 7, n = 512, p =3

conv k = 3, n = 512, d = 2	conv k = 3, n = 512, d = 4
SS-conv k = 3, n = 512, p =1	SS-conv k = 7, n = 512, p =3
conv k = 3, n = 512, d = 2	conv k = 3, n = 512, d = 4
SS-conv k = 3, n = 512, p =1	SS-conv k = 7, n = 512, p =3
conv k = 3, n = 512, d = 2	conv k = 3, n = 512, d = 4
SS-conv k = 3, n = 256, p =1	SS-conv k = 7, n = 256, p =3
conv k = 3, n = 256, d = 2	conv k = 3, n = 256, d = 4
SS-conv k = 3, n = 128, p =1	SS-conv k = 7, n = 128, p =3
conv k = 3, n = 128, d = 2	conv k = 3, n = 128, d = 4
SS-conv k = 3, n = 64, p =1	SS-conv k = 7, n = 64, p =3
conv k = 3, n = 64, d = 2	conv k = 3, n = 64, d = 4
conv k = 1, n = 1, d = 1	

3.8 Dataset

ShanghaiTech Dataset is one of the mostly used crowd dataset containing 1198 annotated images with 330165 persons. Part_A and Part_B contains 482 and 716 images respectively. Part_A is divided into train set of 300 images and test set of 182 images, collected from the internet. Following Figure 13consists of sample images from ShanghaiTech Part_A.

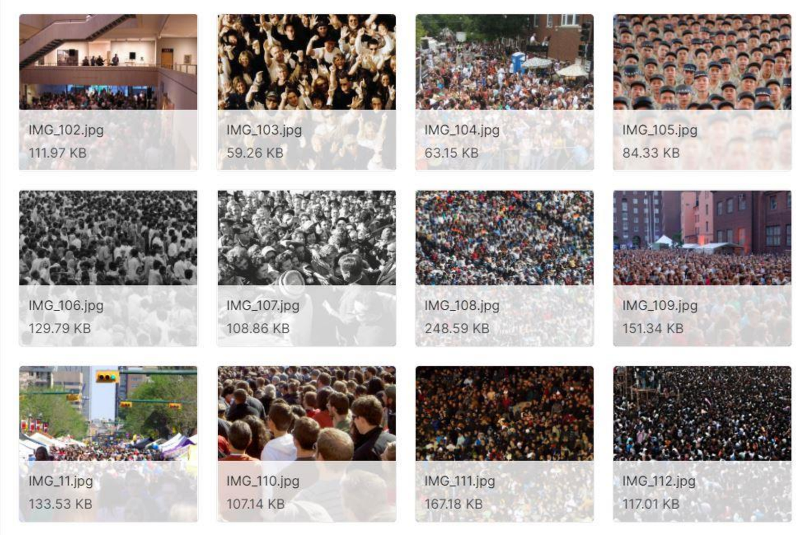


Figure 13: Sample images from Part_A of ShanghaiTech dataset

Part_A is divided into train set of 400 images and test set of 316 images, collected on the streets of Shanghai City. Following Figure 14 consists of sample images from ShanghaiTech Part_B.

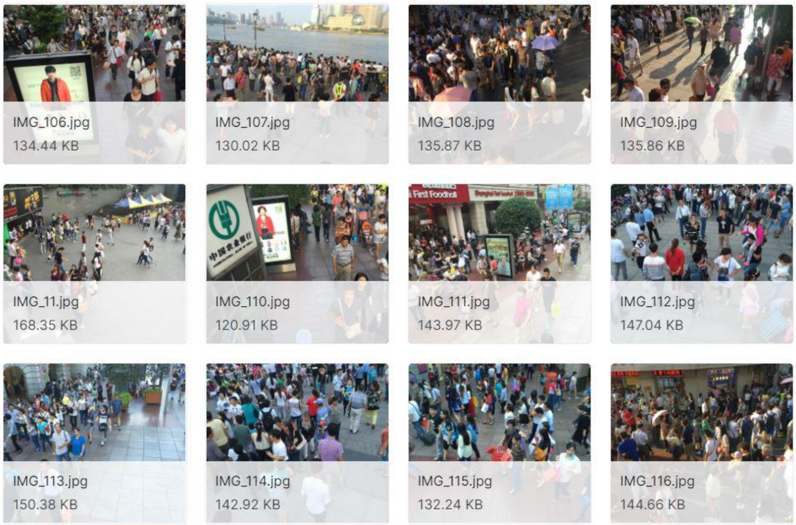


Figure 14: Sample images from Part_B of ShanghaiTech dataset

There is a MATLAB file associated in the dataset containing head annotation i.e., tagged by a dot near to the center of the person’s head. Total of 330,165 annotated people with (x, y) coordinates for the head position is in the mat file. Following Figure 15 is a sample of head annotation for a sample image.

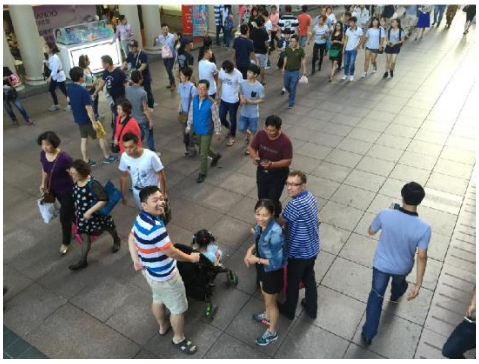


Figure 15: Head annotation of sample image from MAT file

3.9 Ground Truth generation

Density map needs to be generated from the annotation file by blurring each head annotations. In order to address perspective distortion on the human head, varying blur radii according to different head size is needed. However, it is impossible for varying blur radii in each image in each head position. The average distance between the head of neighboring “k” people can be used to control the radial range of the Gaussian Kernel function.

If x_i be the position of each person's head annotation with total N heads then H(X) can be defined as:

$$H(x) = \sum_{i=1}^{N} \delta(x - x_i) \qquad (3)$$

X is 2D coordinate position of pixel in the image and x_i is the targeted object.

Density function is then computed by convolving H(x) function with Gaussian Kernel Function G(x).

$$F(x) = \sum\nolimits_{i=1}^{N} \delta(x - x_i) * G_{\sigma_i}(x), \quad with\, \sigma_i = \beta \bar{d}^i \qquad (4)$$

Here, the input is the two-dimensional coordinate (x, y) and σ parameter controls the radial range of the function. $\bar{\boldsymbol{d}}^{\boldsymbol{i}}$ is the average distance of k-nearest neighbors.

3.10 Radius Nearest Algorithm and Ball Tree

Radius Nearest Algorithm finds all the neighbor within a distance from the query vector. It is useful for range searches and nearest neighbor searches. We can first create the tree and later the same tree can be used to query the nearest points. In this way Nearest Neighbors from any data points can be determined. Specifically, from the root node the algorithm will move down recursively computing whether the point is lesser or greater than the current node. Storing partial results, pruning and traversal order (visit the most promising subtree first) are the techniques for better search.

Like KD-Tree, Ball tree (metric tree) is another binary algorithm of building tree that partitions the data points to two clusters which is contained by a circle (in case of 2D) or sphere (in case of 3D). It partitions data recursively into nodes described by a centroid 'p' and radius 'r', with each node's point falling well within hyper-sphere that is described by 'p' and 'r'. These hyperspheres are also called "balls" hence the name ball tree. Each tree's internal node divides the data points into two distinct groupings, each connected with a separate ball. Each point in the partition is assigned to one of the two balls and the assignment is solely based on its distance from the ball's center.

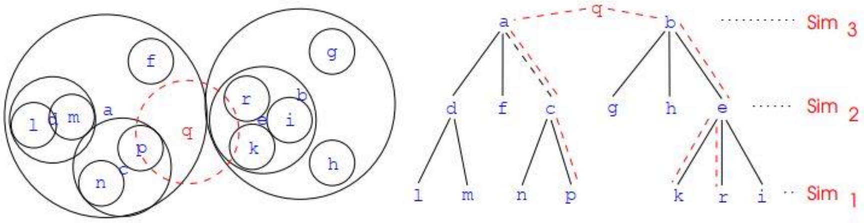


Figure 16: Ball Tree Data Structure

Figure 16 left side shows the ball tree data structure and right shows the ball tree. The range search/query procedure is a recursive one that uses the triangle inequality property. For a query point q, using triangle inequality, we first compute the similarity with top nodes a and b. If the distance to the node center minus the node radius is less than the query radius, i.e., if the query ball meets with the node ball, the search is resumed. The search is extended to include all children nodes that cross the query ball. The number of tree levels, node radii, and query radius all influence the query speed. Compared to KD-tree, computation time is larger since the dimensionality is increased, however with improved performance. [27]

3.11 Loss Function

A loss function is used by machines to learn. It's a way of determining how well a certain algorithm models the data. If the predictions are too far off from the actual results, the loss function will return a very large number. Loss function learns to lower prediction error over time with the help of some optimization function. Broadly, loss function is divided into:

- Regression Loss and
- Classification Loss

To account for the local correlation in the density map, model uses a combination of Euclidean Loss and SSIM Loss function.

3.11.1 Mean Square Loss/Quadratic Loss/L2 Loss

It is the average squared distance computed between model predictions and Ground Truth observations. Because it is squared, far off predictions from actual values are penalized heavily than the less deviated ones.

This particular problem can be thought of as a regression problem. We need to measure the difference of ground-truth with estimated density map that is output by the model. For this measurement of differences, we employ Euclidean distance like the previous papers [20, 21]. The Euclidean Loss Function is defined as:

$$Lq(\Theta) = \sqrt{\frac{1}{2N} \sum_{i=1}^{N} \left|Z(X_i, \Theta) - Z_i^{GT}\right|^{\;2}} \qquad (5)$$

Where N = size of training batch

$Z(X_i, \Theta)$ = output by the model with parameters Θ.

$X_{i\,=}$ input image and

Z_i^{GT} = ground-truth of input image X_i.

3.11.2 Local Pattern Consistency Loss

Along with the Euclidean Loss, SSIM index has been employed to include local correlation in the density map. The SSIM index is commonly used to assess image quality. It uses three local statistics, namely mean, variance, and covariance, to calculate similarity between two images. With respect to [30], a normalized Gaussian Kernel of size 11 x 11 having standard deviation 1.5 is employed to compute the local statistics. If W represent the weight in D containing all positions of kernel, d represent offset from the center, then W is given by:

$$W = \{W(d) \mid d \,\epsilon\, D, D = \{(-5,-5), \ldots\ldots\ldots\ldots, (5,5)\}\} \qquad (6)$$

If F be the predicted density map of the model with its corresponding Ground Truth Y, we can calculate the local statistics as follows:

$$\mu_F(x) \;=\; \sum_{d\,\epsilon\,D} W(d).F(x+d), \qquad (7)$$

$$\sigma_F^2(x) \;=\; \sum_{d\,\epsilon\,D} W(d)\,.\,[F(x+d) - \mu_F(x)]^2, \qquad (8)$$

$$\sigma_{FY}(x) \;=\; \sum_{d\,\epsilon\,D} W(d)\,.\,[F(x+d) - \mu_F(x)]\,.\,[Y(x+d) - \mu_Y(x)], \qquad (9)$$

$$\mu_Y(x) \;=\; \sum_{d\,\epsilon\,D} W(d).F(x+d), \qquad (10)$$

$$\sigma_Y^2(x) \;=\; \sum_{d \,\epsilon\, D} W(d)\,.\,[F(x+d) - \mu_Y(x)]^2, \tag{11}$$

Where:

μ_F = Local Mean estimation of F

σ_F^2 = Local Variance estimation of F

σ_{FY} = Local Covariance estimation

μ_Y = Local Mean estimation of Y

σ_Y^2 = Local Variance estimation of Y

Finally, SSIM index is calculated as in Equation (12).

$$SSIM = \frac{(2\mu_F\mu_Y + C_1)(2\sigma_{FY} + C_2)}{(\mu_F^2 + \mu_Y^2 + C_1)(\sigma_F^2 + \sigma_Y^2 + C_2)} \tag{12}$$

Small constant values C_1 and C_2 prevent division by zero and are assigned as in [28].

Equation (13) defines the local pattern consistency loss.

$$L_{SSIM} = \; 1 - \frac{1}{N}\sum_{x} SSIM(x), \tag{13}$$

Where:

N = No. of pixels

L_{SSIM} = Local pattern consistency loss

3.11.3 Compositional Loss

With respective to Equation (5) and Equation (13), the objective function is defined as:

$$\mathrm{L} = \; \mathrm{L_q} + \alpha_c \mathrm{L_{SSIM}} \tag{14}$$

α_c controls the pixel-wise loss and local pattern consistency loss. It is set to 0.001 as in [30]

3.12 Evaluation Metric

MAE and MSE are used as the metric for performance comparison. MAE is a measure of how accurate the predicted crowd count is throughout the test sequence.

[13] For a test sequence with N images, MAE is computed as in Equation (15).

$$MAE = \frac{1}{N}\sum_{i=1}^{N} | C_i - C_i^{GT} | \qquad (15)$$

Where:

C_i is crowd count estimate from the model

C_i^{GT} is the crowd count from Ground Truth.

The MSE represents the predictability of the count. MSE is defined as in Equation (16), for a test sequence with N images.

$$MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} | C_i - C_i^{GT} |^2} \qquad (16)$$

The crowd count estimate as predicted by the model, C_i is given by Equation (17).

$$C_i = \sum_{l=1}^{L}\sum_{w=1}^{W} z_{l,w} \qquad (17)$$

Where:

L = length of the density-map

W = width of the density-map

Such that in the so obtained density-map, $z_{l,w}$ is the pixel at position (l, w).

3.13 Tools

Programming is done in Python using PyTorch library and executed in Google Colaboratory GPU. ShanghaiTech dataset is used during this work wherein both Part_A and Part_B are used to develop separate models.

4. RESULT ANALYSIS AND COMPARISON

4.1 Setup

ShanghaiTech dataset has been used in this work, which is one of the mostly used crowd counting dataset. It is divided into two parts namely A and B whereby Part_A consists of dense crowd and Part_B consists of sparse crowd. Since model working on a dense crowded prediction might not accurately model sparse crowd so two parts are provided. All the images have been used in the experiment. The crowd count distribution for Part_A and Part_B is shown in Figure 17.

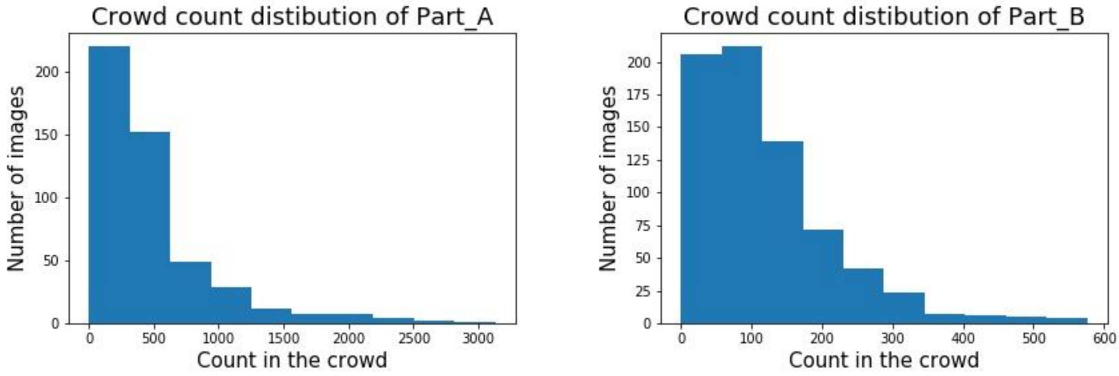


Figure 17: Crowd count distribution

Observation: The dataset is severely unbalanced.

Dataset contains head annotation in MATLAB file. Figure 18 is a sample of head annotation and in the zoomed portion on the right, we can see red colored dot marking in the head of each person, this is achieved from the annotation file.

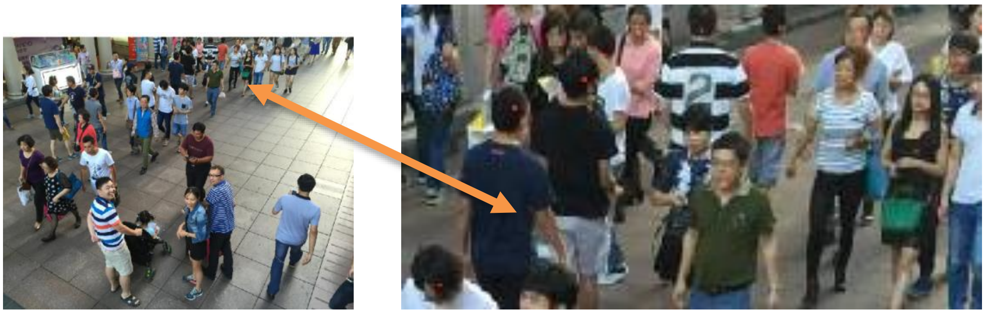
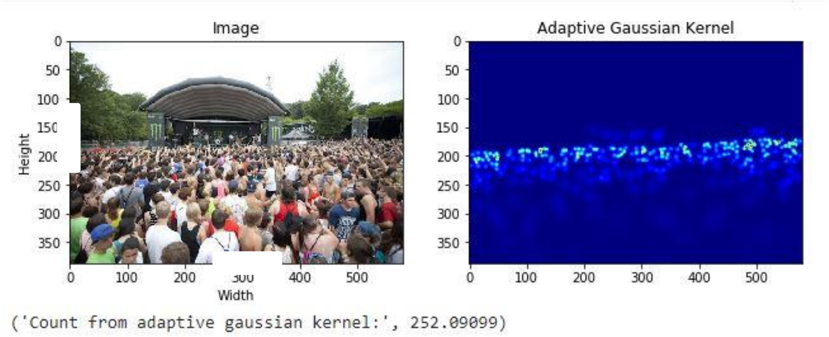


Figure 18: Head annotation for head of each person in the dataset

Regarding density map generation, for Part_A adaptive Gaussian Kernel is employed since the crowd density is large. This means low degree of burring for the region of dense crowd and high degree for the region of sparse crowd. For adaptive Gaussian Kernel, Ball Tree with dimensionality 4 and leaf-size 2048 has been used because there are many head counts in Part_A dataset. 3

neighbors are computed in order to compute the location information from the current node. β value is set to 0.3 [20]. Gaussian Kernel is normalized with sum 1 to generate density map.

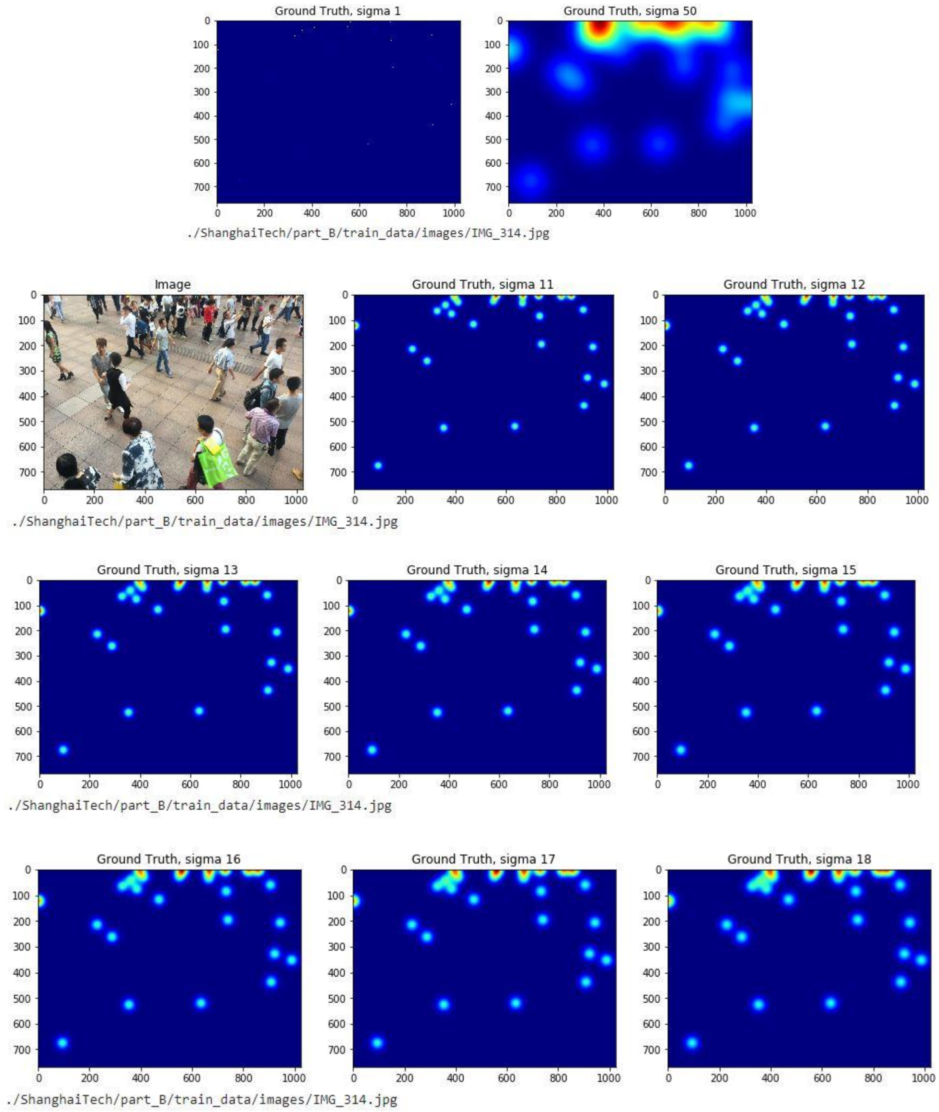
Following Figure 19 is the Ground Truth and count generated for the Part_A dataset so that we can begin the training process.



```
('Count from adaptive gaussian kernel:', 252.09099)
```

Figure 19: Ground Truth and count for Part_A

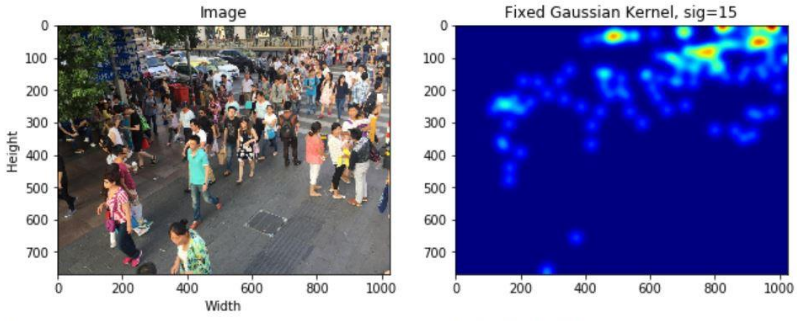
Apart from that, for Part_B, dataset that contains sparse crowd, fixed size kernel (σ = 15) is employed for the average head size in order to blur all the annotations. σ is the spread parameter of the Gaussian Kernel. It signifies the extent of blurring in the so obtained density map. Since we are using Gaussian distribution to incorporate the size of the human head and it can't be set to 1 because the head in the image does not contain one pixel for head size. It undoubtedly covers a certain pixel in terms of area. The geometry of the image plane is not provided in the dataset. So, an assumption needs to be made on the spread parameter to proceed for the Ground Truth generation. Following Figure 20 shows the Ground Truth generation of sample image in Part_B using various σ values. It can be seen that if σ is set to 1 then it represents a very tiny dot for head position. However, the head size is bigger and it will be further difficult to visualize the density map. Similar is the case for σ value set to 50 where every head is represented by a huge spread Gaussian Kernel that leads to insignificant density map. Apart from arbitrary setting, if the spread parameter is set in the range [13,18], clear visualization of density map can be seen. Since the motive of this research work is to analyze and compare whether use of Smooth Dilated CNN will improve the dense prediction compared to Dilated CNN as in Li et. al. [13], same environmental value leads to the better comparison than tweaking everything. Keeping this in mind σ is set to 15 as in [13] to analyze better in terms of varied model employing Dilated CNN and Smooth Dilated CNN.



```
./ShanghaiTech/part_B/train_data/images/IMG_314.jpg
./ShanghaiTech/part_B/train_data/images/IMG_314.jpg
./ShanghaiTech/part_B/train_data/images/IMG_314.jpg
./ShanghaiTech/part_B/train_data/images/IMG_314.jpg
```

Figure 20: Analysis for varied σ values

Following Figure 21 is the Ground Truth and count generated for the Part_B dataset so that we can begin the training process.



```
('Count from fixed gaussian kernel:', 130.0000000000001)
```

Figure 21: Ground Truth and count for Part_B

4.2 Comparison of adaptive vs fixed Gaussian Kernel

4.2.1 Comparison of adaptive vs fixed Gaussian Kernel for Part_A

The original image and the generated Ground Truth from Part_A are depicted in Figure 22. Left column represents original image, middle column represents Ground Truth obtained using adaptive Gaussian Kernel and right column represents the Ground Truth obtained using fixed sized Gaussian Kernel.

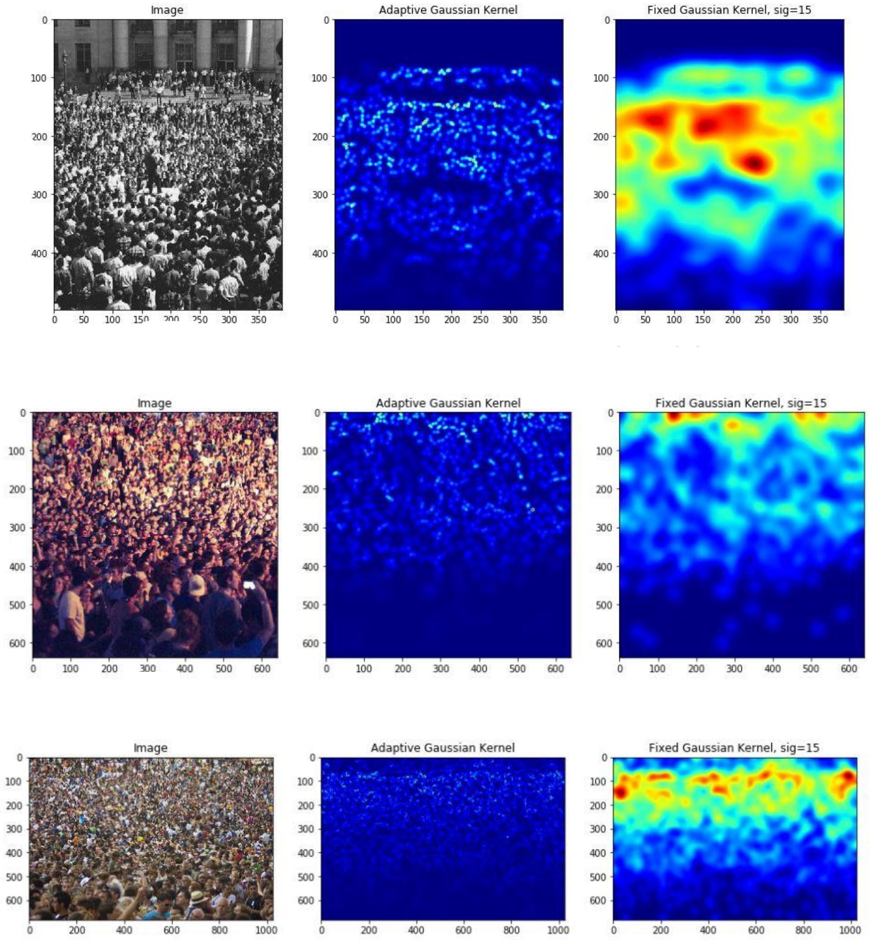
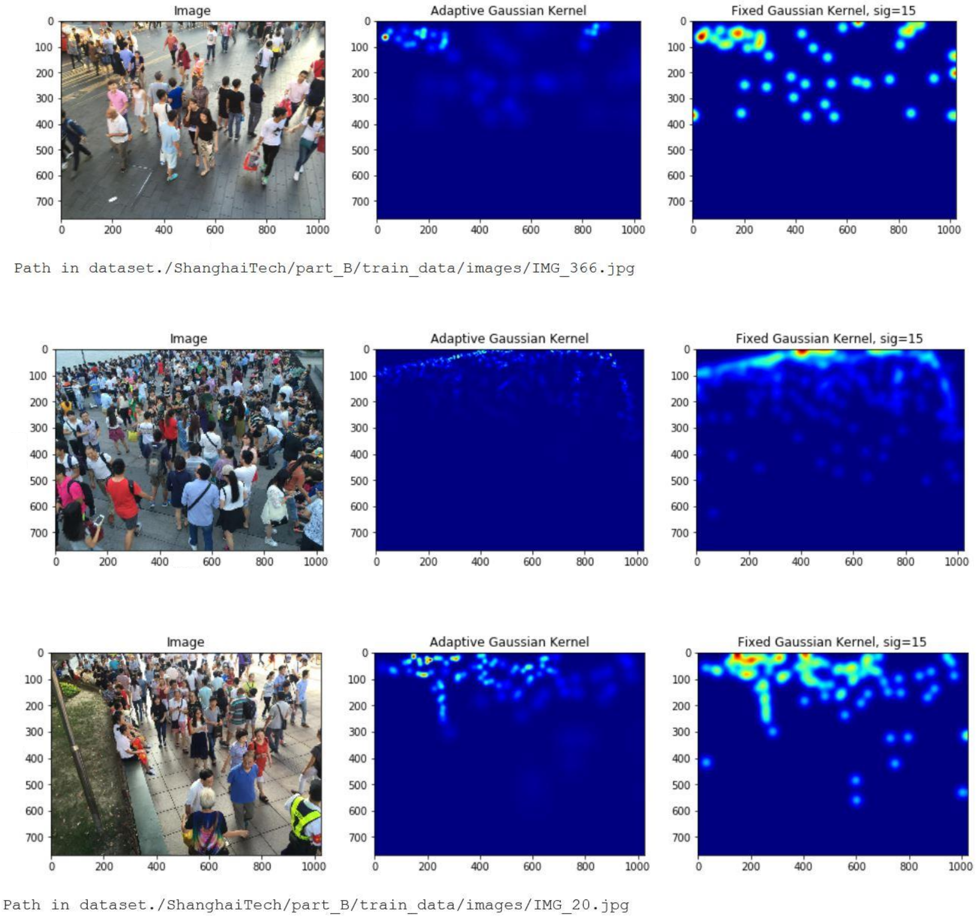


Figure 22: Comparison of adaptive vs fixed Gaussian Kernel – Part_A

Observation: In this Part_A dataset density map is more suitable if we adopt adaptive Gaussian Kernel due to blur adaptation for denser and sparse region.

4.2.2 Comparison of adaptive vs fixed Gaussian Kernel for Part_B

The original image and the generated Ground Truth from Part_A are depicted in Figure 23. Left column represents original image, middle column represents Ground Truth obtained using adaptive Gaussian Kernel and right column represents the Ground Truth obtained using fixed-sized Gaussian Kernel.



```
Path in dataset./ShanghaiTech/part_B/train_data/images/IMG_366.jpg
```

```
Path in dataset./ShanghaiTech/part_B/train_data/images/IMG_20.jpg
```

Figure 23: Comparison of adaptive vs fixed Gaussian Kernel – Part_B

Observation: If we use adaptive Gaussian Kernel for sparse crowd then density map can't address well in the region of sparse crowd. However, if we use fixed-sized Gaussian Kernel then we can see head position on the sparse region as well.

4.3 Comparison of the developed models for Part_A

4.3.1 Training Loss

Model is end-to-end trainable, initial values for the layers are assigned by Gaussian initialization with standard deviation 0.01. Model has been trained with starting learning rate of 1e-6, SGD optimizer with momentum 0.9 subject to combinational loss, up to 435 epochs. Hyper parameter tuning and learning rate decrement was done in accordance with the learning graph. Batch size of the training is set to 1 because pre-processing the images (crop, resize) has been avoided so as to capture the essence of image and thereby generate high quality ground truth for training. In order to measure the Ground Truth and predicted density map, Euclidean distance metric is used along with SSIM. To compute the SSIM, Ground Truth is resized to 1/8 so as to match the size with the predicted density map. Finally, objective function is computed for the optimizer. Two different models are created with dilation rate 2 and 4 as shown in Table 1. Figure 24 shows the comparison between the two models.

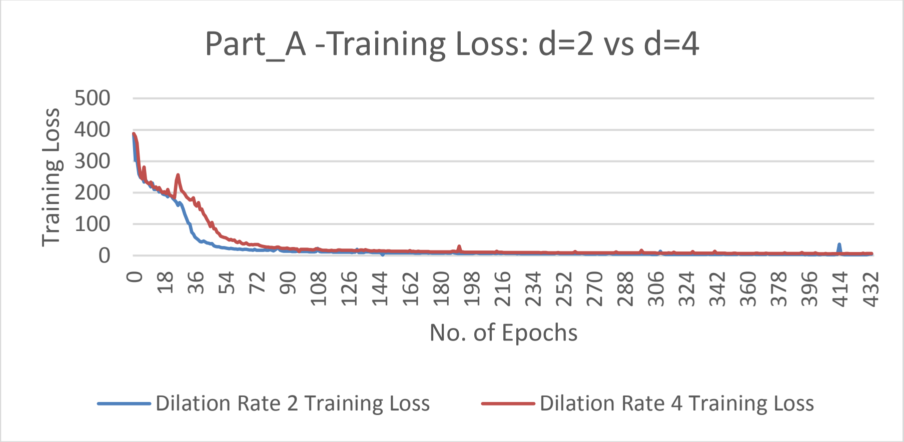


Figure 24: Training Loss for Part_A

Observation: Training Loss is decreasing with the number of epochs. Few spikes are seen because batch size is set to 1.

4.3.2 Validation Loss

Figure 25 shows the validation loss comparison for dilation rate 2 and 4.

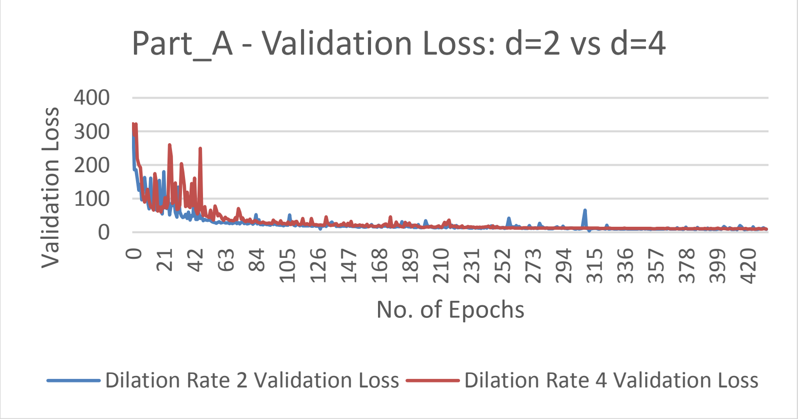


Figure 25: Validation Loss for Part_A

Observation: It can be observed that the loss is decreasing with the number of epochs thereby increased accuracy.

4.4 Testing Accuracy - Part_A

Ground Truth vs the count estimated by the model for Part_A is shown in Figure 26. In the dataset there are 182 test images, index of which are shown in x-axis of the chart. Y-axis represents the total count of people. Orange color represents "Ground Truth" and blue color represents "Predicted Count".

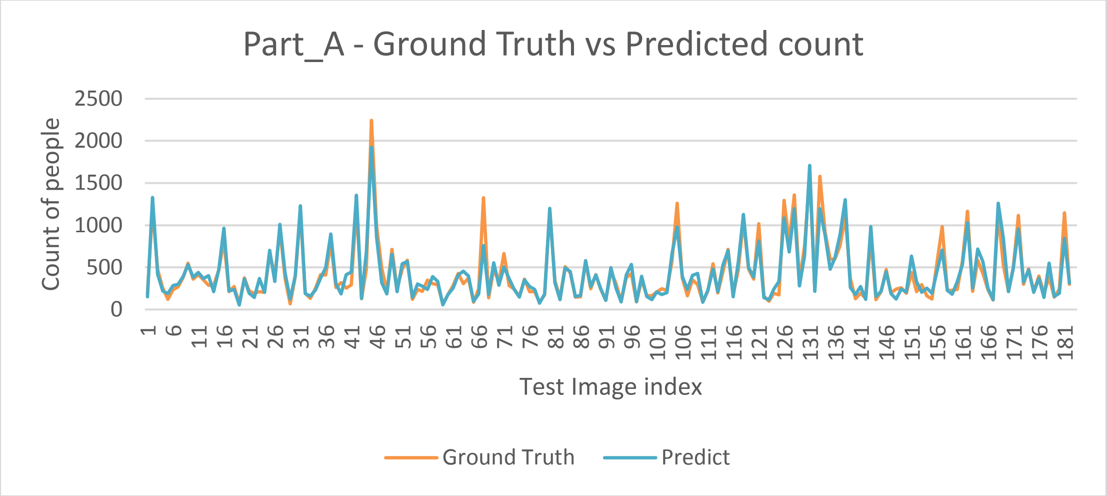


Figure 26: Ground Truth vs Predicted Count Part_A

Observation: The overall MAE is 67.64 for the test set of Part_A.

4.5 Evaluation of Part_A

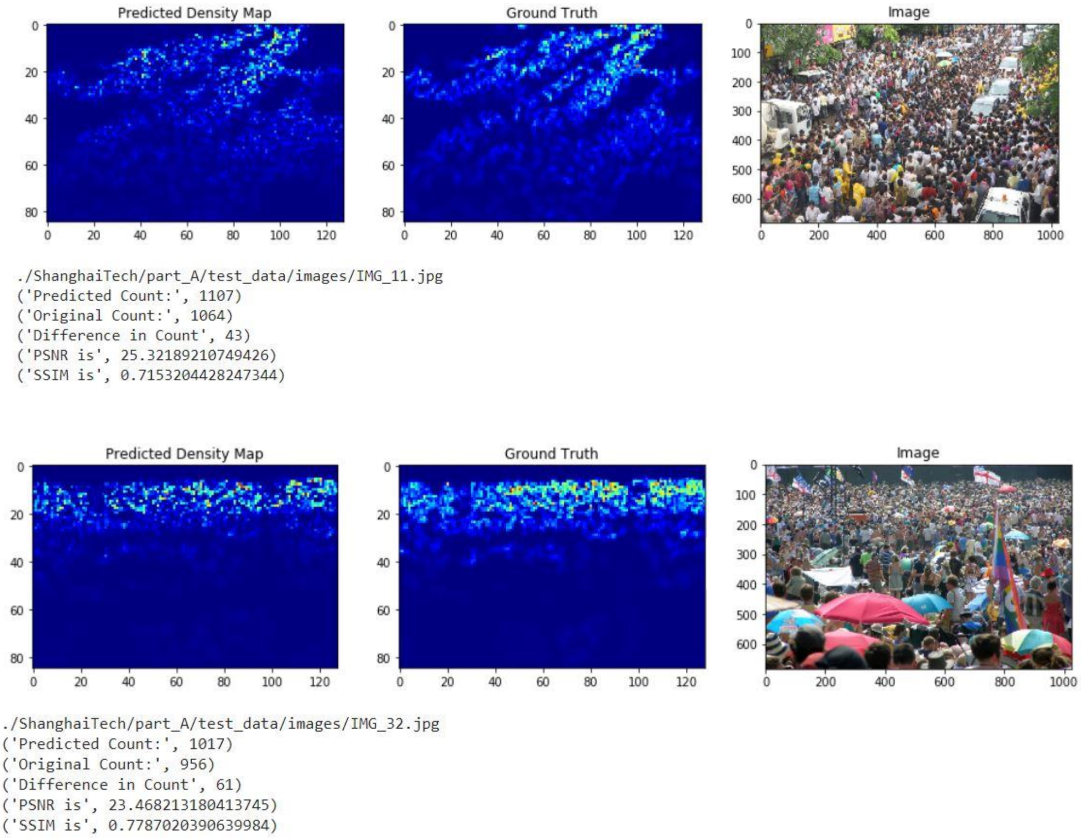
MAE, MSE, PSNR and SSIM has been used as evaluation metric for the model. Table 1 shows the performance of different architecture with respect to base paper. The main base paper is CSRNET which use CNN for feature extraction and dilated CNN in the backend for the density map generation. MCNN and CP-CNN also does dense prediction but by employing multi-column and multi-level architecture. Basically, they use the same density map generation process using adaptive and fixed Gaussian Kernel.

Table 2: Evaluation of Part_A

ShanghaiTech Part_A					
Metric	**Dilation = 2**	**Dilation = 4**	**CSRNET**	**MCNN**	**CP-CNN**
MAE	67.64	67.89	68.2	110.2	73.6
MSE	103.39	103.48	115	173.2	106.4
PSNR	24.1	24	23.79	N/A	N/A
SSIM	0.77	0.71	0.76	N/A	N/A

Observation: Model has increased both the counting accuracy and the quality of density map. When dilation rate = 2 was applied MAE and MSE increased by 0.82% and 10.09% respectively. Similarly, the PSNR and SSIM increased by 1.3%. when compared to the approach in Li et. al. [13] that used dilated convolution in their network. When compared to Zhang et. al. [20], that used multi column CNN for the same but with the same Ground Truth generation process, we can see that the counting accuracy is vastly improved. Since we avoid the multi column, the model is very lighter and computationally cheaper than [20]. Apart from that Sindagi et. al. [31], used contextual pyramid that can capture local and global information for generation of density map. Counting accuracy is significantly increased by this approach as compared to [31].

Some of the results for Part_A is shown in Figure 27. The image index is shown along with the predicted count, the original count, the difference in the count. Similarly, image quality metrics SSIM and PSNR of the image is also shown.



```
./ShanghaiTech/part_A/test_data/images/IMG_11.jpg
('Predicted Count:', 1107)
('Original Count:', 1064)
('Difference in Count', 43)
('PSNR is', 25.32189210749426)
('SSIM is', 0.7153204428247344)
```

```
./ShanghaiTech/part_A/test_data/images/IMG_32.jpg
('Predicted Count:', 1017)
('Original Count:', 956)
('Difference in Count', 61)
('PSNR is', 23.468213180413745)
('SSIM is', 0.7787020390639984)
```

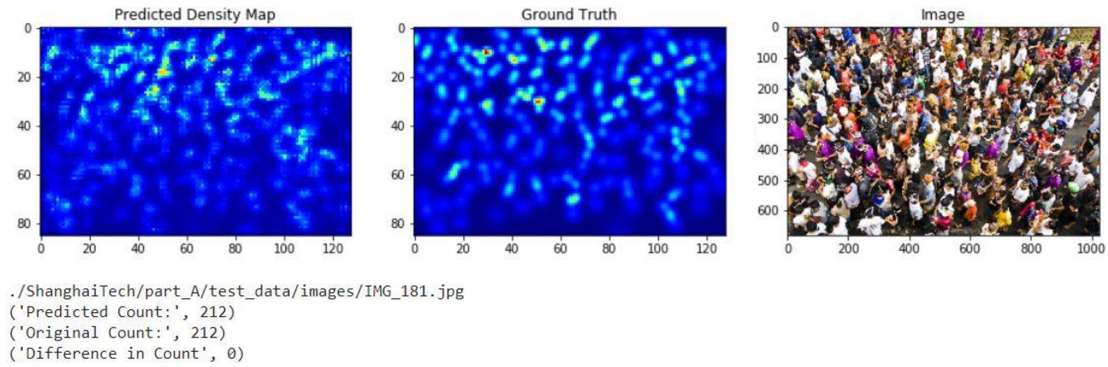
Figure 27: Samples of Part_A

4.6 Results from Part_A

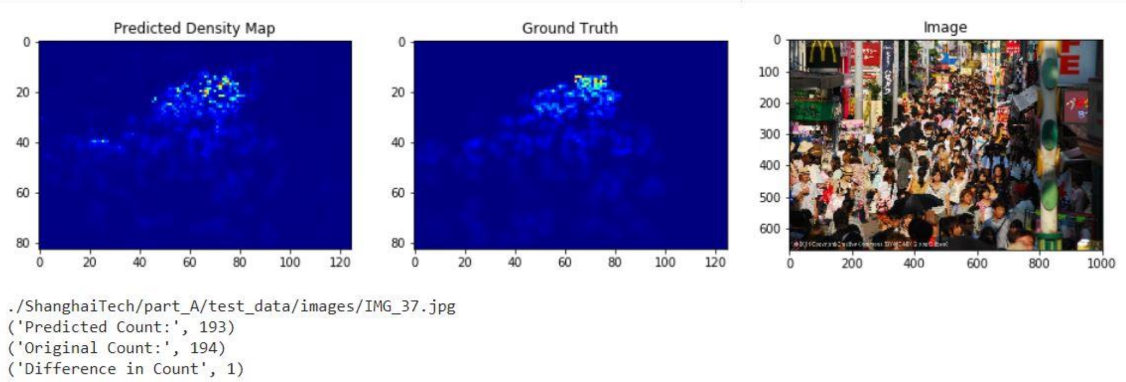
Following are results obtained using dilation rate = 2.

4.6.1 Some high accuracy result for Part_A dataset

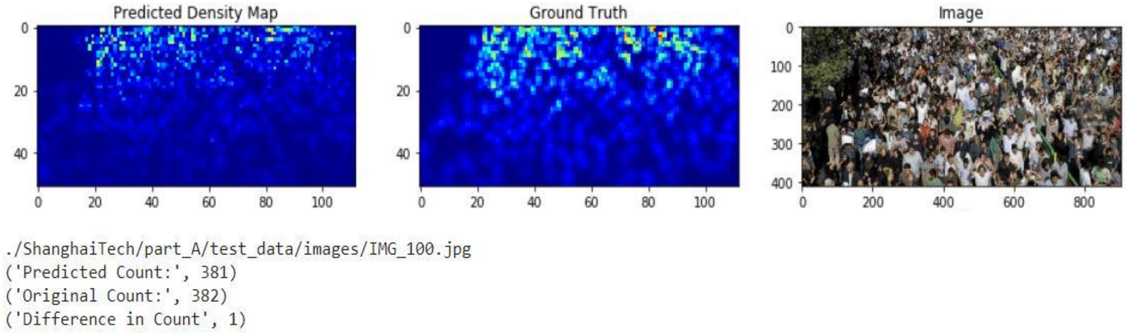
Some of the good predictions in terms of counting accuracy are shown in Figure 28.



```
./ShanghaiTech/part_A/test_data/images/IMG_181.jpg
('Predicted Count:', 212)
('Original Count:', 212)
('Difference in Count', 0)
```



```
./ShanghaiTech/part_A/test_data/images/IMG_37.jpg
('Predicted Count:', 193)
('Original Count:', 194)
('Difference in Count', 1)
```

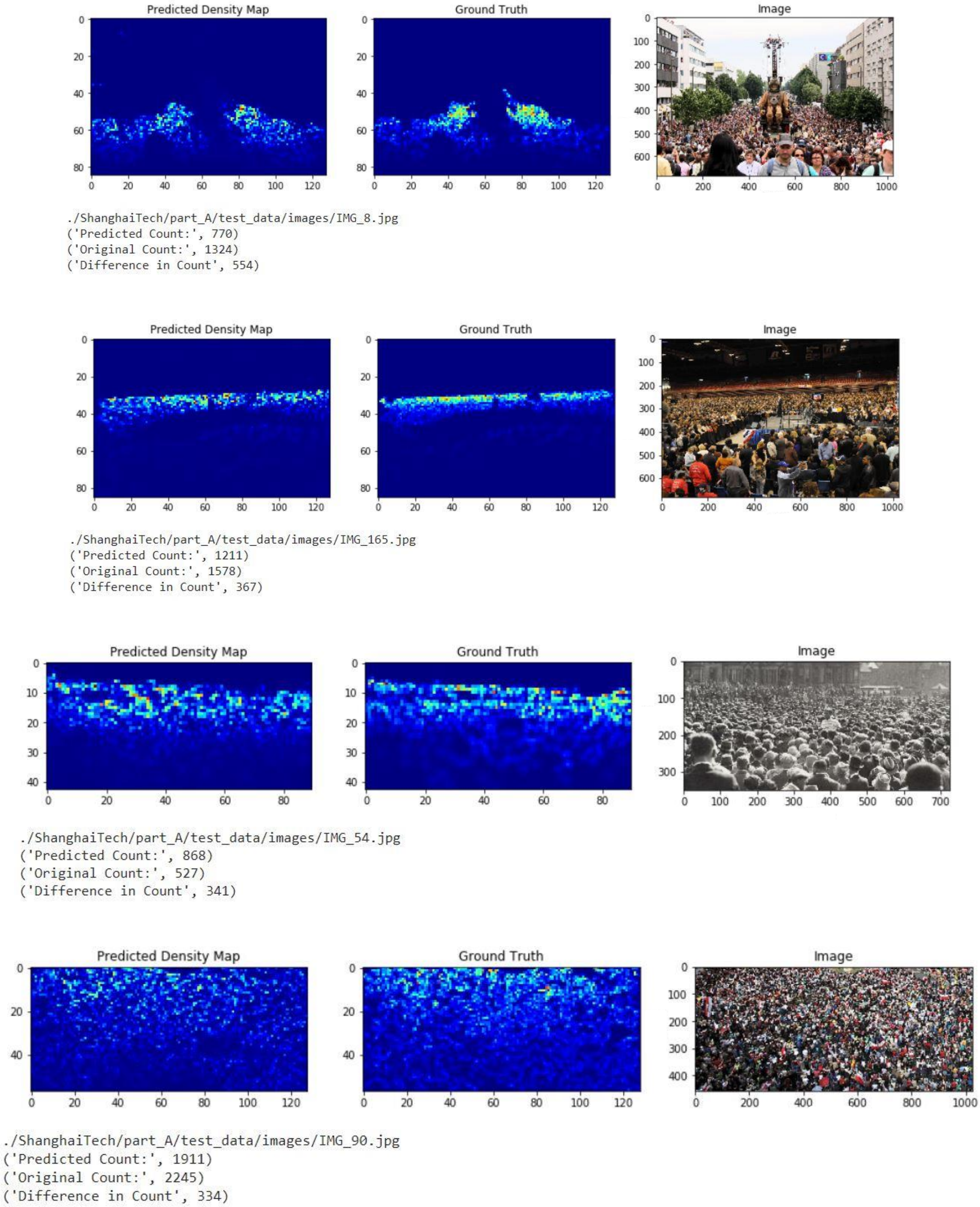


```
./ShanghaiTech/part_A/test_data/images/IMG_100.jpg
('Predicted Count:', 381)
('Original Count:', 382)
('Difference in Count', 1)
```

Figure 28: Some good prediction from Part_A

4.6.2 Some low accuracy result for Part_A dataset

Predictions have comparatively lower counting accuracy are shown in Figure 29.



```
./ShanghaiTech/part_A/test_data/images/IMG_8.jpg
('Predicted Count:', 770)
('Original Count:', 1324)
('Difference in Count', 554)
```

```
./ShanghaiTech/part_A/test_data/images/IMG_165.jpg
('Predicted Count:', 1211)
('Original Count:', 1578)
('Difference in Count', 367)
```

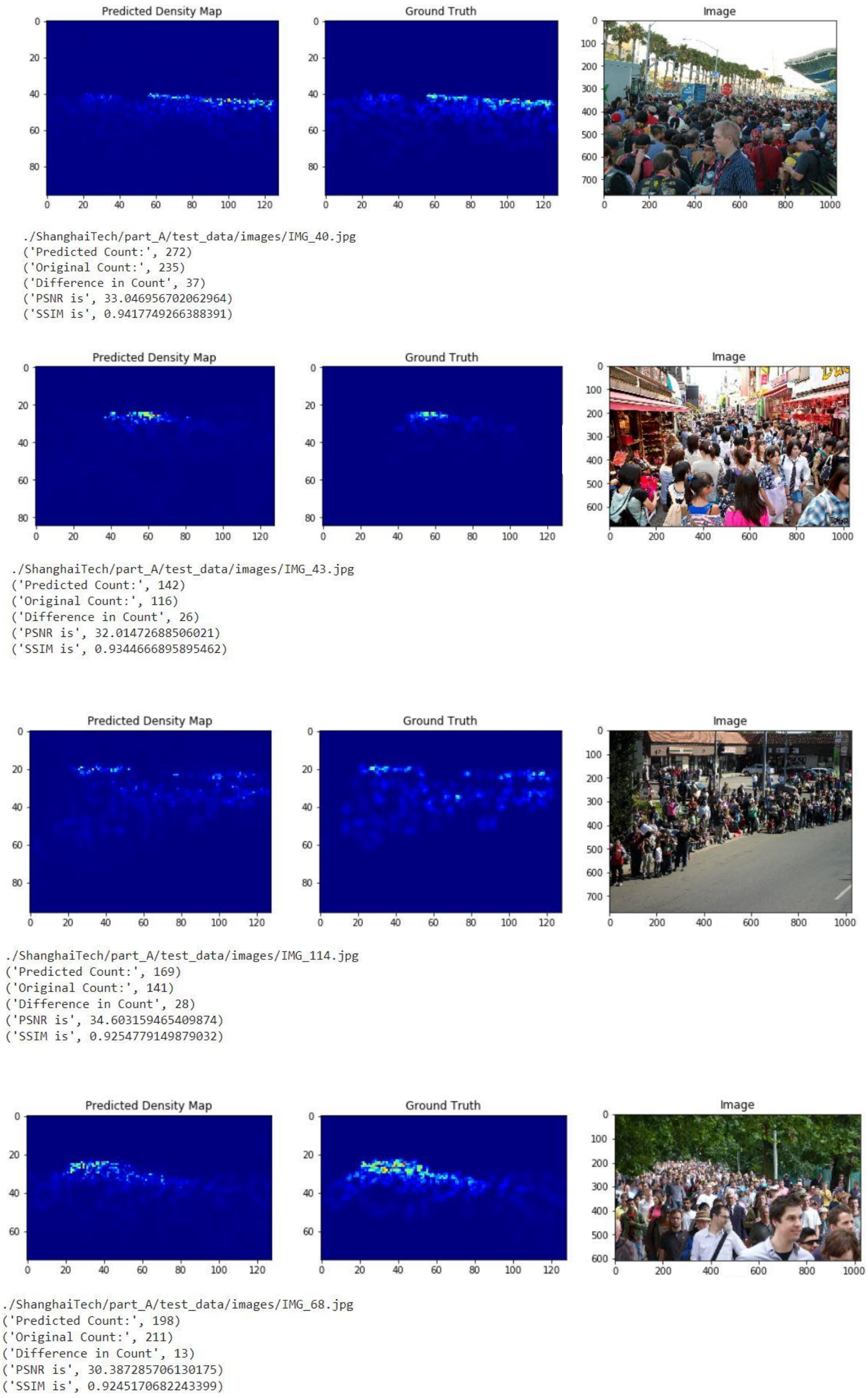
```
./ShanghaiTech/part_A/test_data/images/IMG_54.jpg
('Predicted Count:', 868)
('Original Count:', 527)
('Difference in Count', 341)
```

```
./ShanghaiTech/part_A/test_data/images/IMG_90.jpg
('Predicted Count:', 1911)
('Original Count:', 2245)
('Difference in Count', 334)
```

Figure 29: Less accurate crowd estimation from Part_A

4.6.3 Some good SSIM result for Part_A

Some of the prediction with best SSIM are shown in Figure 30.



```
./ShanghaiTech/part_A/test_data/images/IMG_40.jpg
('Predicted Count:', 272)
('Original Count:', 235)
('Difference in Count', 37)
('PSNR is', 33.046956702062964)
('SSIM is', 0.9417749266388391)
```

```
./ShanghaiTech/part_A/test_data/images/IMG_43.jpg
('Predicted Count:', 142)
('Original Count:', 116)
('Difference in Count', 26)
('PSNR is', 32.01472688506021)
('SSIM is', 0.9344666895895462)
```

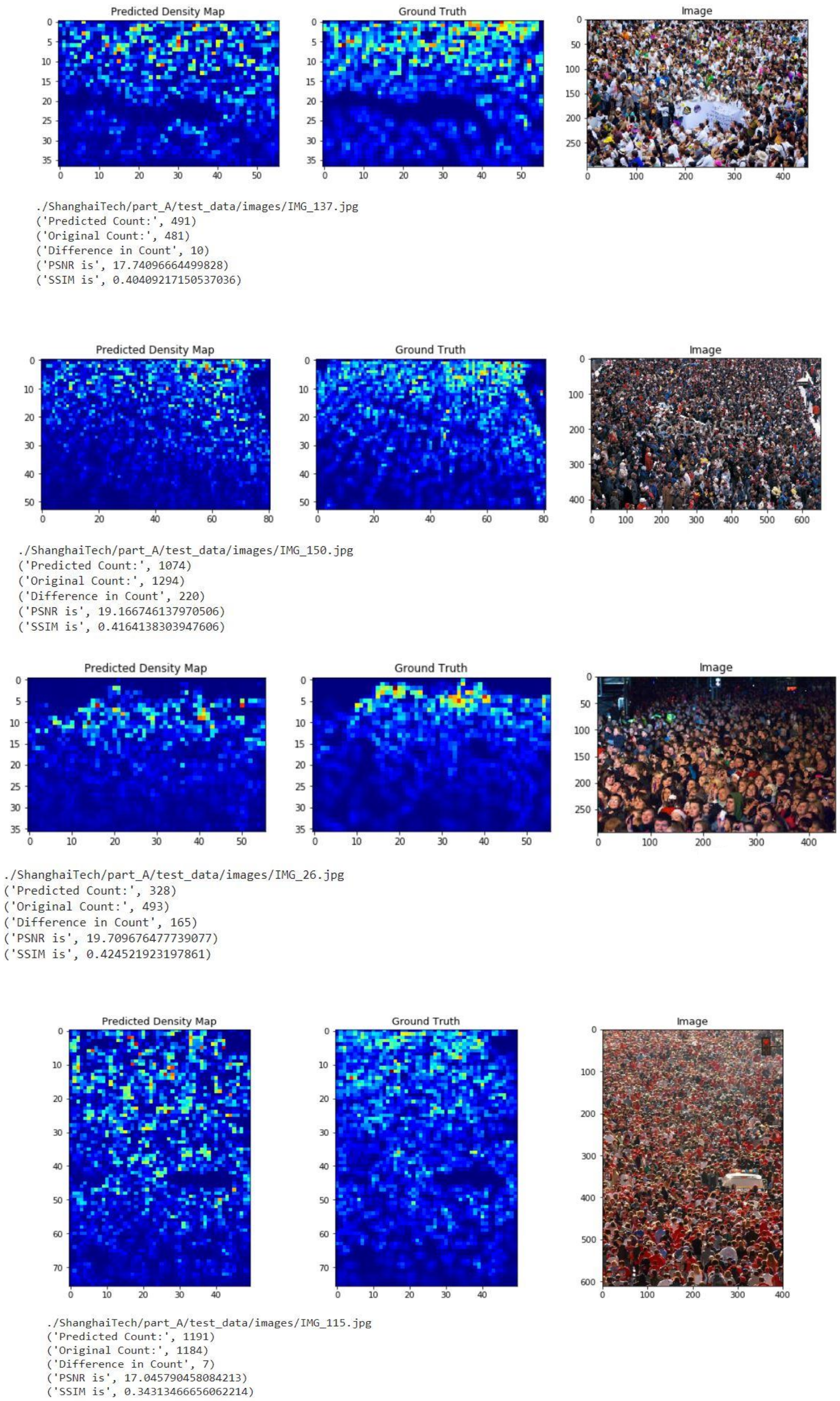
```
./ShanghaiTech/part_A/test_data/images/IMG_114.jpg
('Predicted Count:', 169)
('Original Count:', 141)
('Difference in Count', 28)
('PSNR is', 34.603159465409874)
('SSIM is', 0.9254779149879032)
```

```
./ShanghaiTech/part_A/test_data/images/IMG_68.jpg
('Predicted Count:', 198)
('Original Count:', 211)
('Difference in Count', 13)
('PSNR is', 30.387285706130175)
('SSIM is', 0.9245170682243399)
```

Figure 30: Good SSIM prediction for Part_A

4.6.4 Some low SSIM results for Part_A

Some of the prediction with comparatively low SSIM are shown in Figure 31.



```
./ShanghaiTech/part_A/test_data/images/IMG_137.jpg
('Predicted Count:', 491)
('Original Count:', 481)
('Difference in Count', 10)
('PSNR is', 17.74096664499828)
('SSIM is', 0.40409217150537036)
```

```
./ShanghaiTech/part_A/test_data/images/IMG_150.jpg
('Predicted Count:', 1074)
('Original Count:', 1294)
('Difference in Count', 220)
('PSNR is', 19.166746137970506)
('SSIM is', 0.4164138303947606)
```

```
./ShanghaiTech/part_A/test_data/images/IMG_26.jpg
('Predicted Count:', 328)
('Original Count:', 493)
('Difference in Count', 165)
('PSNR is', 19.709676477739077)
('SSIM is', 0.424521923197861)
```

```
./ShanghaiTech/part_A/test_data/images/IMG_115.jpg
('Predicted Count:', 1191)
('Original Count:', 1184)
('Difference in Count', 7)
('PSNR is', 17.045790458084213)
('SSIM is', 0.34313466656062214)
```

Figure 31: Some low SSIM prediction for Part_A

4.7 Comparison of the developed models for Part_B

4.7.1 Training Loss

Here also, initial values for the layers are assigned by Gaussian initialization with standard deviation 0.01. Model has been trained with starting learning rate of 1e-6, SGD optimizer with momentum 0.85 subject to combinational loss, up to 230 epochs. Hyper parameter tuning and learning rate decrement was done in accordance with the learning graph. Batch size of the training is set to 1. Two different models are created with dilation rate 2 and 4 as shown in Table 1. Figure 32 shows the comparison between the two models.

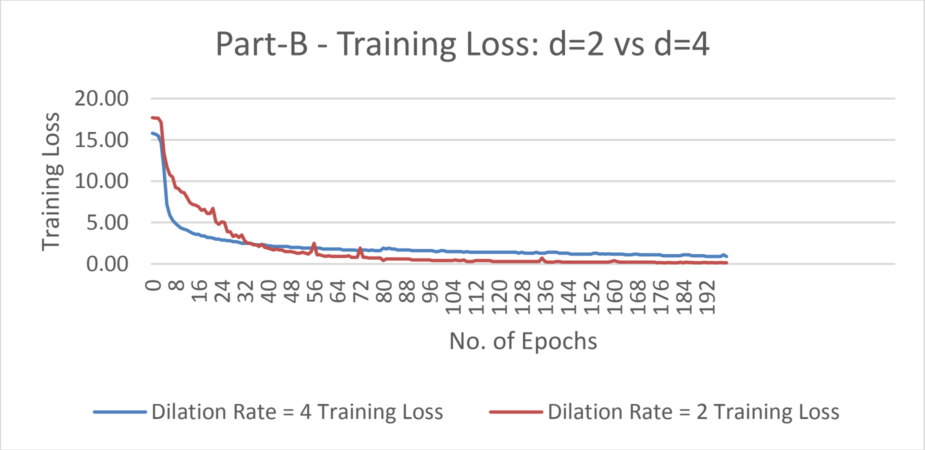


Figure 32: Training loss for Part_B

Observation: Training Loss is decreasing with the number of epochs. Few spikes are seen because batch size is set to 1.

4.7.2 Validation Loss

The validation loss chart is shown in Figure 33.

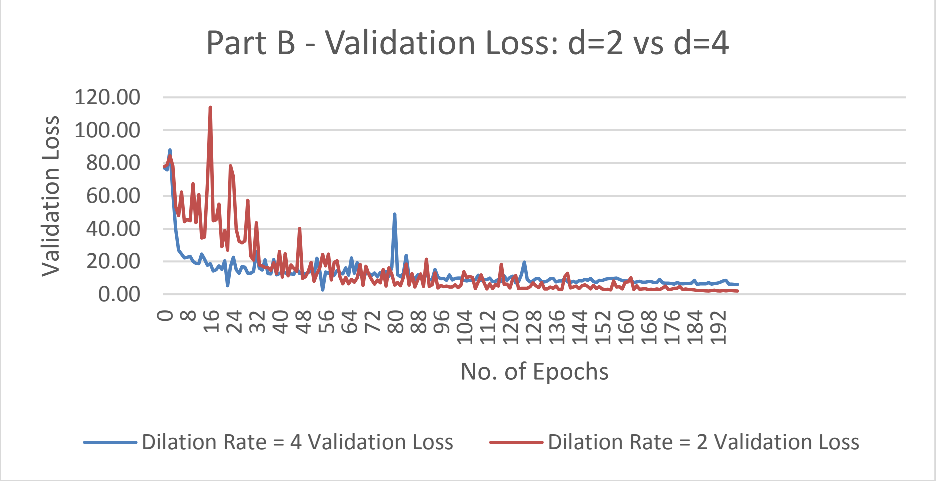


Figure 33: Validation Loss for Part_B

Observation: It can be observed that the loss is decreasing with the number of epochs thereby increased accuracy.

4.8 Testing Accuracy Part_B

Ground Truth vs the count estimated by the model for Part_B is shown in Figure 34. In the dataset there are 316 test images which is shown in x-axis of the chart. Y-axis represents the total count of people. Orange color represents "Ground Truth" and blue color represents "Predicted Count".

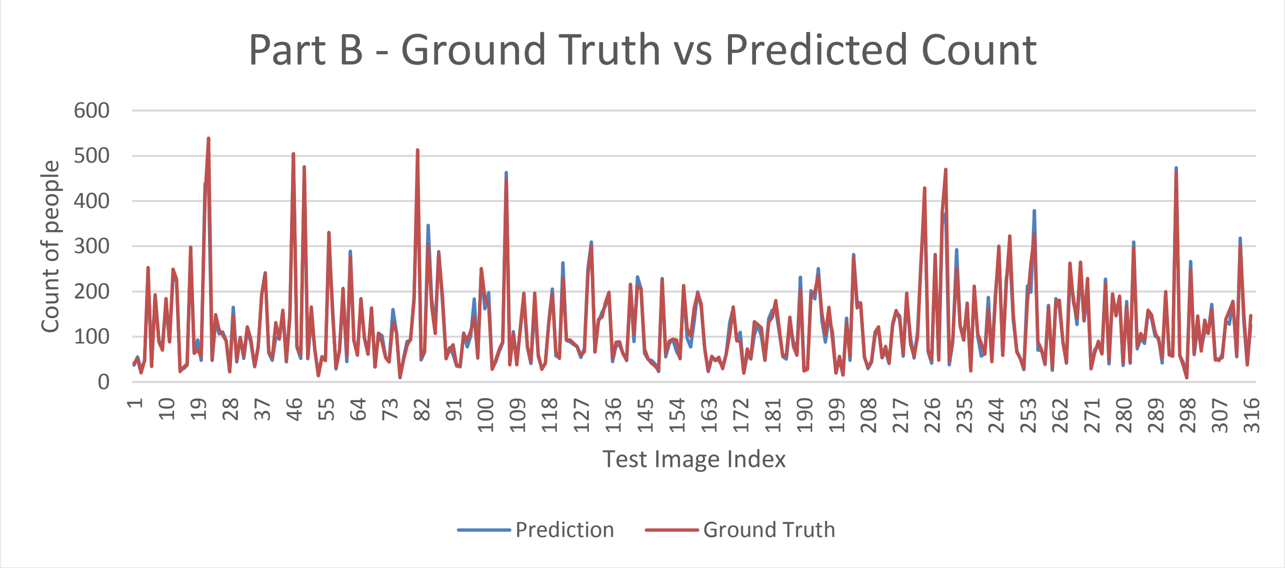


Figure 34: Ground Truth vs Predicted Count Part_B

Observation: The overall MAE is 9.6 for the test set of Part_B.

4.9 Evaluation of Part_B

MAE, MSE, PSNR and SSIM has been used as evaluation metric for the model. Following Table 3 shows the performance of different architecture with respect to base paper.

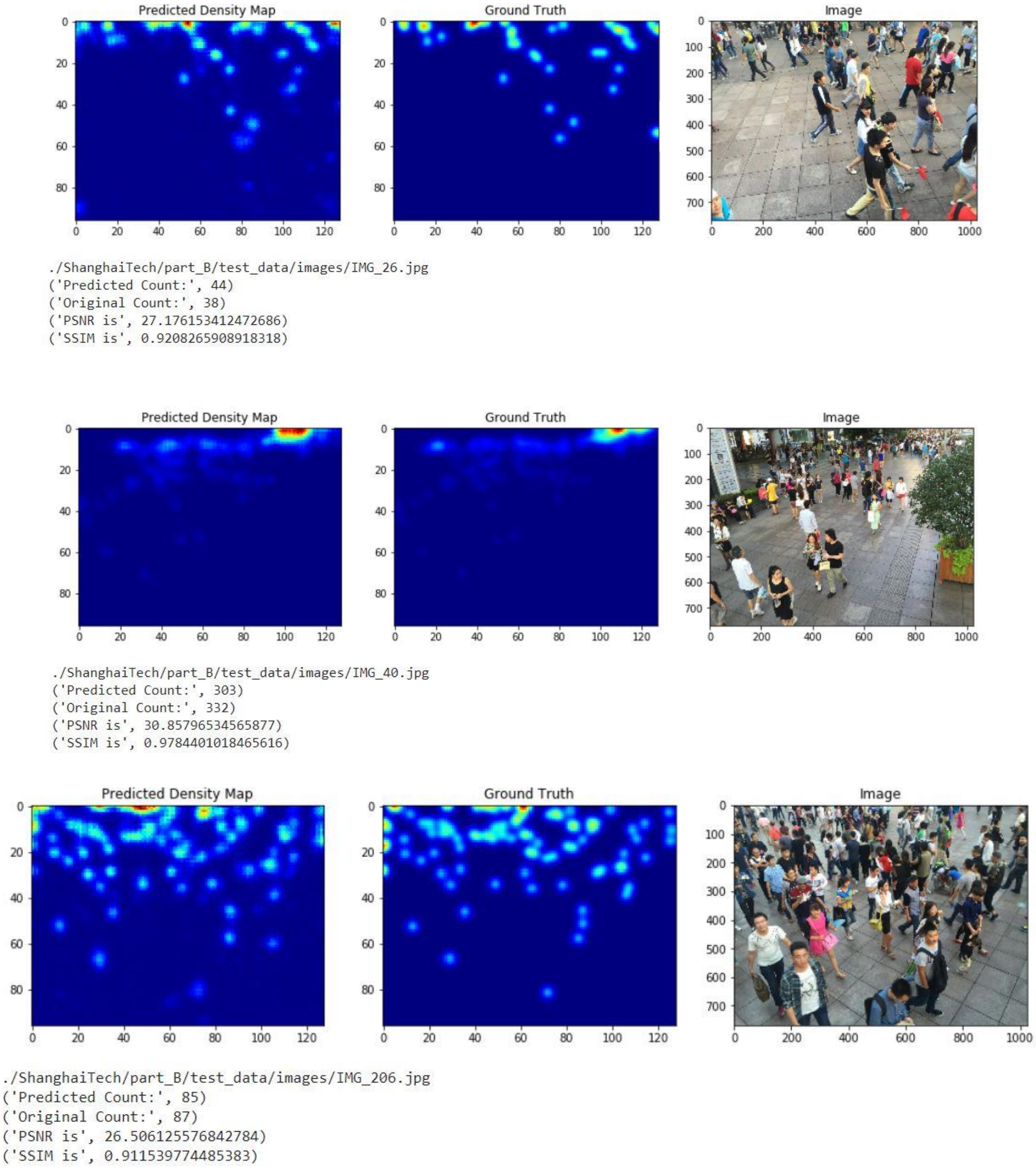
Table 3: Evaluation of Part_B

ShanghaiTech Part_B					
Metric	**Dilation = 2**	**Dilation = 4**	**CSRNET**	**MCNN**	**CP-CNN**
MAE	9.6	9.87	10.6	26.4	20.1
MSE	15.41	15.48	16	41.3	30
PSNR	27.88	27.86	27.02	N/A	N/A
SSIM	0.927	0.926	0.89	N/A	N/A

Observation: Model has increased both the counting accuracy and the quality of density map. When dilation rate = 2 was applied, MAE and MSE increased by 9.4% and 3.6% respectively.

Similarly, the PSNR and SSIM increased by 3.18% and 4.16% when compared to CSRNET. Moreover, the accuracy is significantly increased compared to MCNN and CP-CNN.

Some of the results are illustrated in Figure 35. Predicted count, Ground Truth count, difference in count along with the image quality metrics SSIM and PSNR is shown in figure.



```
./ShanghaiTech/part_B/test_data/images/IMG_26.jpg
('Predicted Count:', 44)
('Original Count:', 38)
('PSNR is', 27.176153412472686)
('SSIM is', 0.9208265908918318)
```

```
./ShanghaiTech/part_B/test_data/images/IMG_40.jpg
('Predicted Count:', 303)
('Original Count:', 332)
('PSNR is', 30.85796534565877)
('SSIM is', 0.9784401018465616)
```

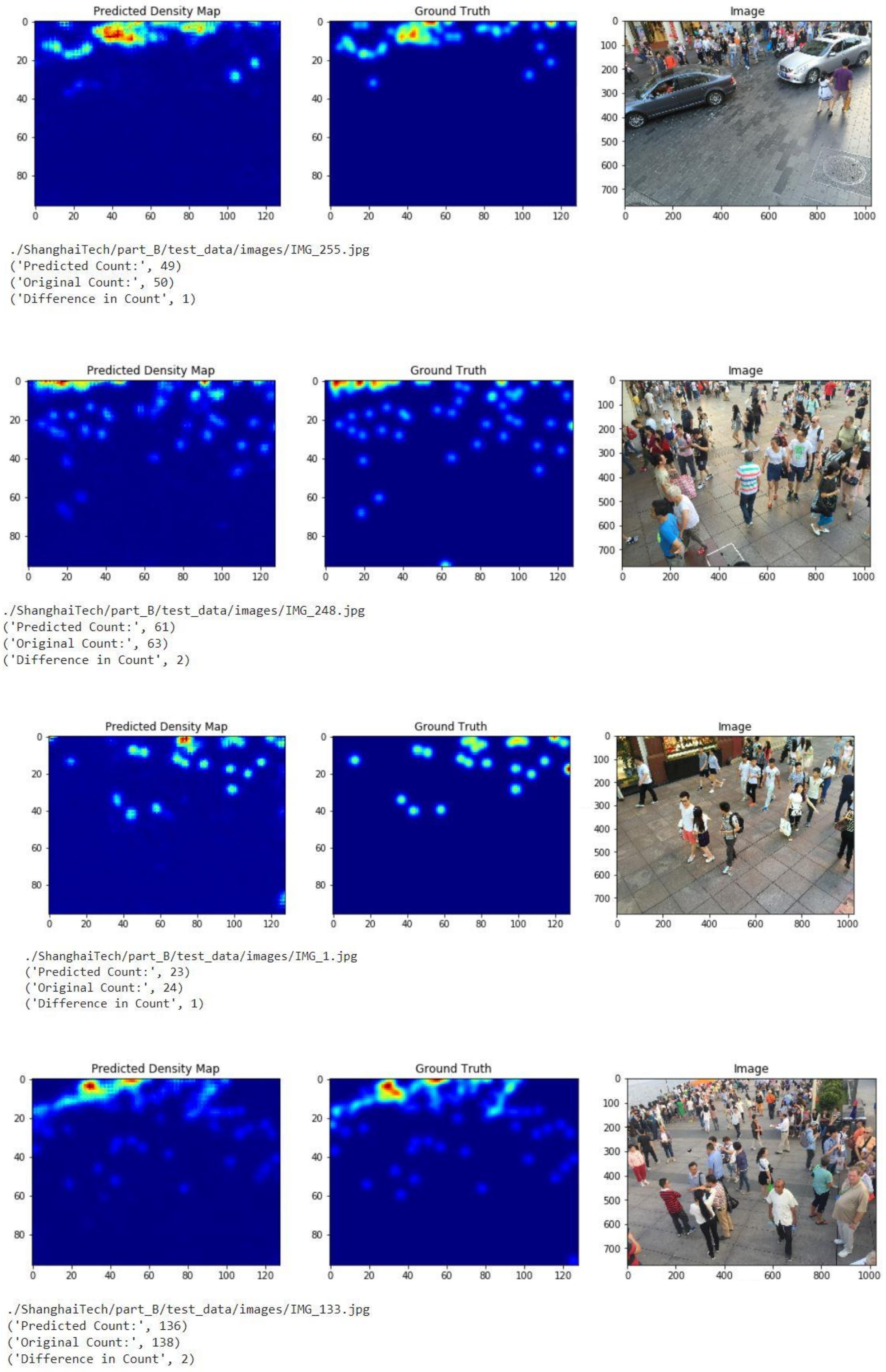
```
./ShanghaiTech/part_B/test_data/images/IMG_206.jpg
('Predicted Count:', 85)
('Original Count:', 87)
('PSNR is', 26.506125576842784)
('SSIM is', 0.911539774485383)
```

Figure 35: Samples of Part_B

4.10 Results from Part_B

4.10.1 Some high accuracy result for Part_B dataset

Some of the good predictions in terms of counting accuracy are shown in Figure 36.



```
./ShanghaiTech/part_B/test_data/images/IMG_255.jpg
('Predicted Count:', 49)
('Original Count:', 50)
('Difference in Count', 1)
```

```
./ShanghaiTech/part_B/test_data/images/IMG_248.jpg
('Predicted Count:', 61)
('Original Count:', 63)
('Difference in Count', 2)
```

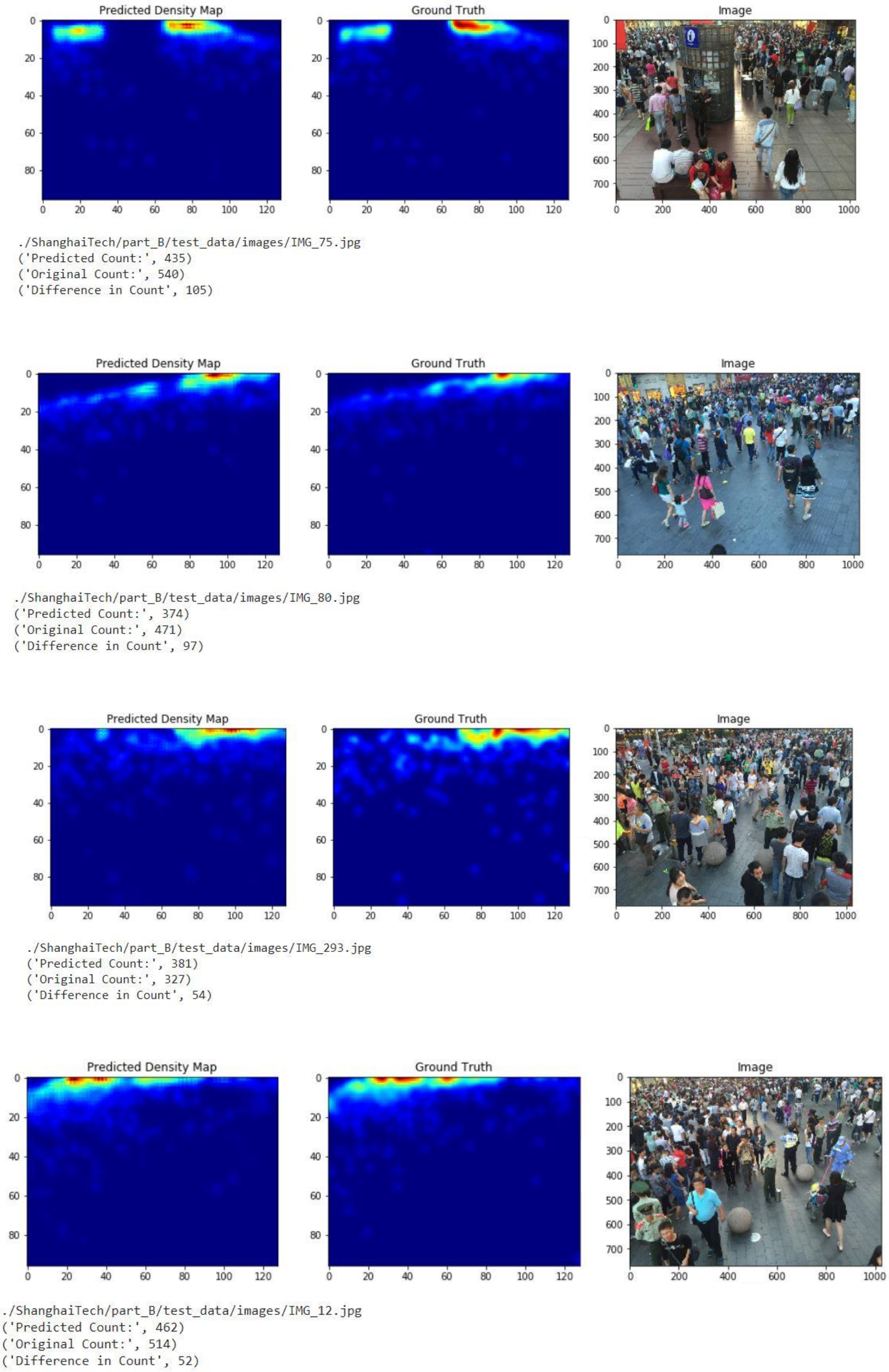
```
./ShanghaiTech/part_B/test_data/images/IMG_1.jpg
('Predicted Count:', 23)
('Original Count:', 24)
('Difference in Count', 1)
```

```
./ShanghaiTech/part_B/test_data/images/IMG_133.jpg
('Predicted Count:', 136)
('Original Count:', 138)
('Difference in Count', 2)
```

Figure 36: Some good prediction from Part_B

4.10.2 Some low accuracy result for Part_B dataset

Predictions have comparatively lower counting accuracy are shown in Figure 37.



```
./ShanghaiTech/part_B/test_data/images/IMG_75.jpg
('Predicted Count:', 435)
('Original Count:', 540)
('Difference in Count', 105)
```

```
./ShanghaiTech/part_B/test_data/images/IMG_80.jpg
('Predicted Count:', 374)
('Original Count:', 471)
('Difference in Count', 97)
```

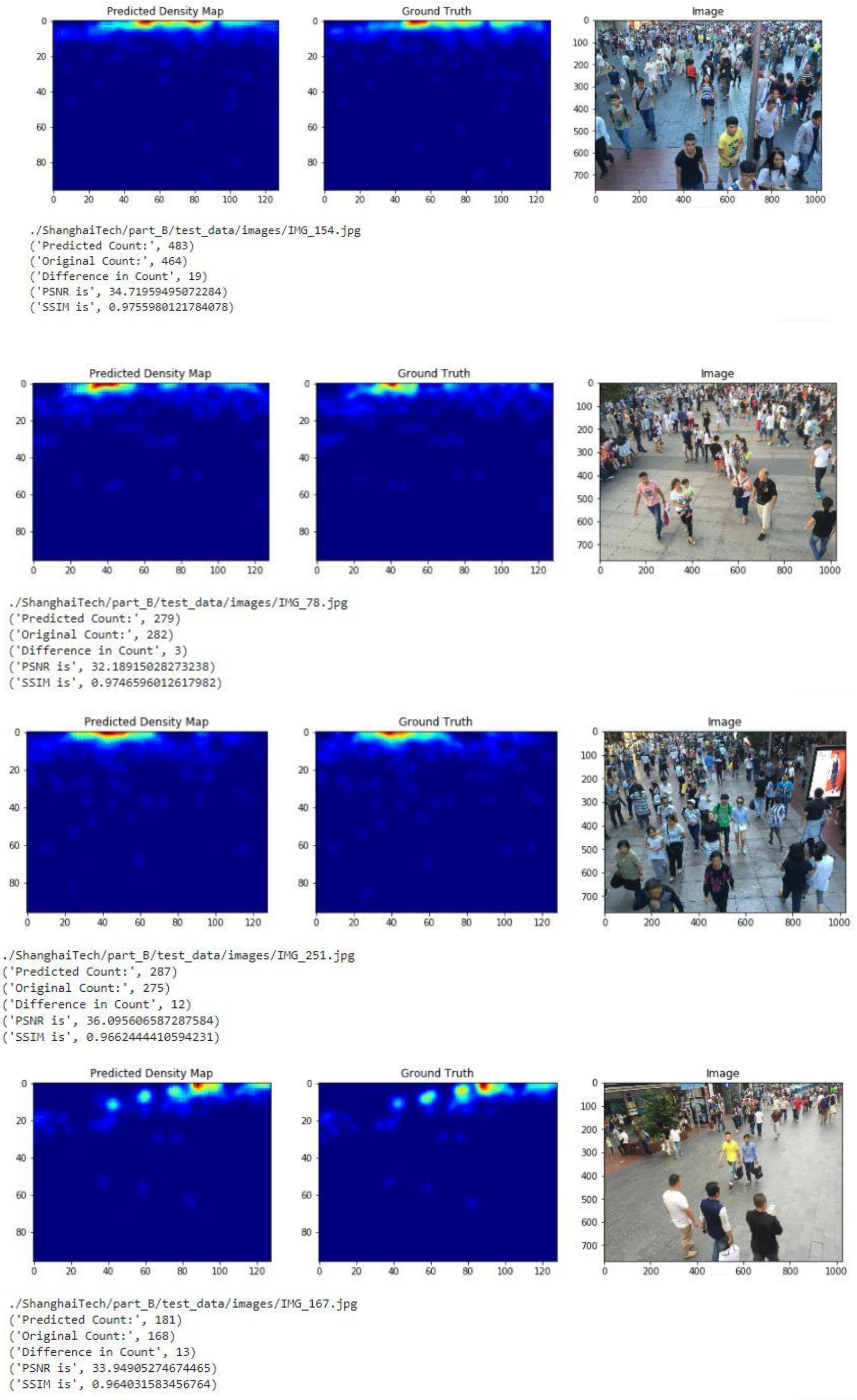
```
./ShanghaiTech/part_B/test_data/images/IMG_293.jpg
('Predicted Count:', 381)
('Original Count:', 327)
('Difference in Count', 54)
```

```
./ShanghaiTech/part_B/test_data/images/IMG_12.jpg
('Predicted Count:', 462)
('Original Count:', 514)
('Difference in Count', 52)
```

Figure 37: Less accurate crowd estimation from Part_B

4.10.3 Some good SSIM result for Part_B

Some of the prediction with best SSIM are shown in Figure 38.



```
./ShanghaiTech/part_B/test_data/images/IMG_154.jpg
('Predicted Count:', 483)
('Original Count:', 464)
('Difference in Count', 19)
('PSNR is', 34.71959495072284)
('SSIM is', 0.9755980121784078)
```

```
./ShanghaiTech/part_B/test_data/images/IMG_78.jpg
('Predicted Count:', 279)
('Original Count:', 282)
('Difference in Count', 3)
('PSNR is', 32.18915028273238)
('SSIM is', 0.9746596012617982)
```

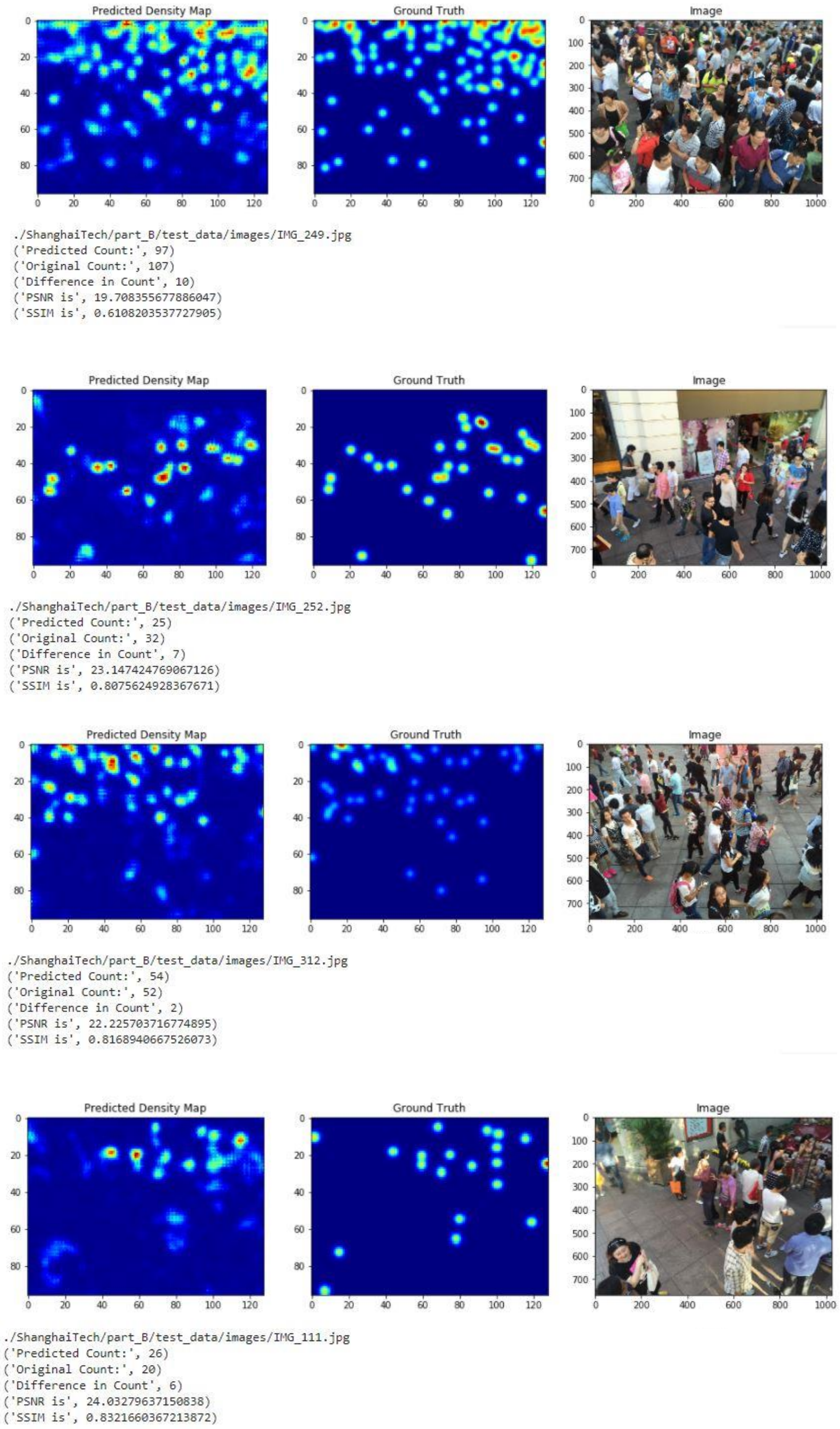
```
./ShanghaiTech/part_B/test_data/images/IMG_251.jpg
('Predicted Count:', 287)
('Original Count:', 275)
('Difference in Count', 12)
('PSNR is', 36.095606587287584)
('SSIM is', 0.9662444410594231)
```

```
./ShanghaiTech/part_B/test_data/images/IMG_167.jpg
('Predicted Count:', 181)
('Original Count:', 168)
('Difference in Count', 13)
('PSNR is', 33.94905274674465)
('SSIM is', 0.964031583456764)
```

Figure 38: Good SSIM prediction for Part_B

4.10.4 Some low SSIM results for Part_B dataset

Some of the prediction with comparatively low SSIM are shown in Figure 39.



```
./ShanghaiTech/part_B/test_data/images/IMG_249.jpg
('Predicted Count:', 97)
('Original Count:', 107)
('Difference in Count', 10)
('PSNR is', 19.708355677886047)
('SSIM is', 0.6108203537727905)
```

```
./ShanghaiTech/part_B/test_data/images/IMG_252.jpg
('Predicted Count:', 25)
('Original Count:', 32)
('Difference in Count', 7)
('PSNR is', 23.147424769067126)
('SSIM is', 0.8075624928367671)
```

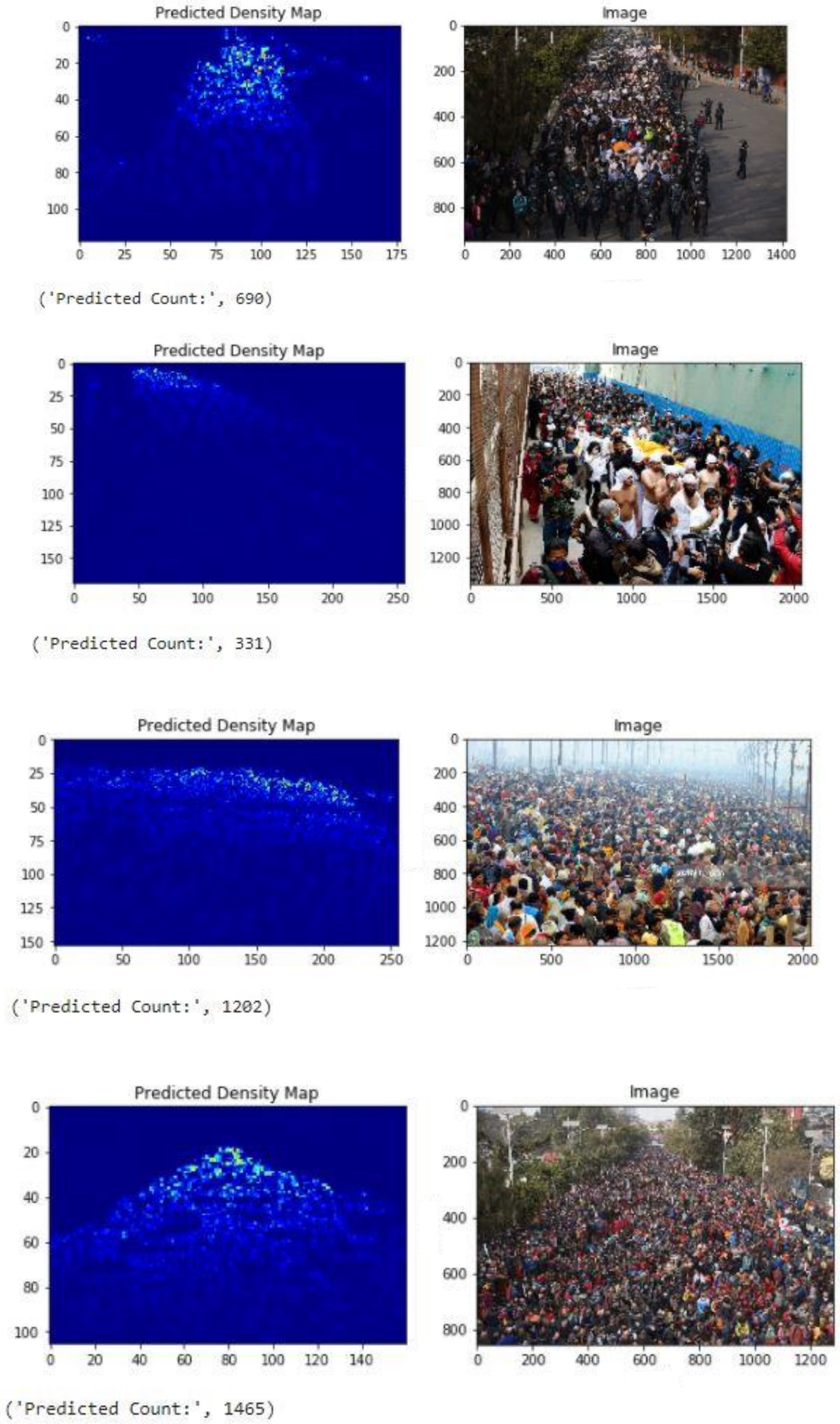
```
./ShanghaiTech/part_B/test_data/images/IMG_312.jpg
('Predicted Count:', 54)
('Original Count:', 52)
('Difference in Count', 2)
('PSNR is', 22.225703716774895)
('SSIM is', 0.8168940667526073)
```

```
./ShanghaiTech/part_B/test_data/images/IMG_111.jpg
('Predicted Count:', 26)
('Original Count:', 20)
('Difference in Count', 6)
('PSNR is', 24.03279637150838)
('SSIM is', 0.8321660367213872)
```

Figure 39: Some low SSIM prediction for Part_B

4.11 Test of model in local data

Following Figure 40 shows the model estimate in context of crowd data in Nepal.



```
('Predicted Count:', 690)
```

```
('Predicted Count:', 331)
```

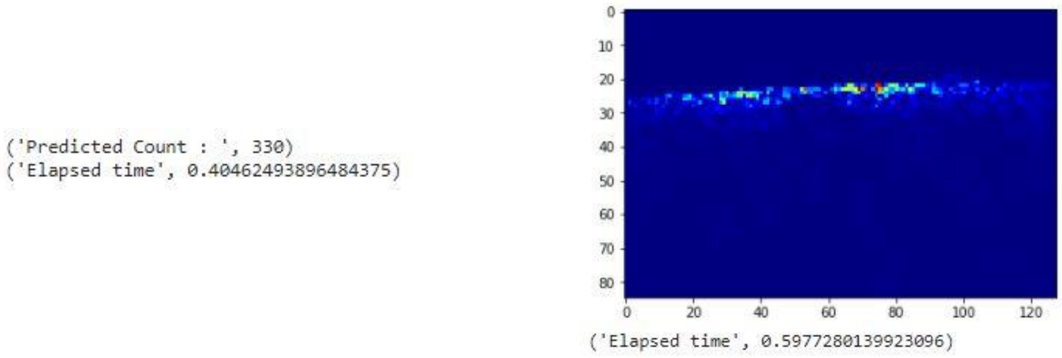
```
('Predicted Count:', 1202)
```

```
('Predicted Count:', 1465)
```

Figure 40: Crowd estimate on local data

4.12 Computation time

The computation time for two instances is shown in Figure 41. Left figure represents the computation time for crowd estimation which is 0.405 seconds and the right figure represents the time for obtaining the density map which is 0.598 seconds.



```
('Predicted Count : ', 330)
('Elapsed time', 0.40462493896484375)
```

```
('Elapsed time', 0.5977280139923096)
```

Figure 41: Computation Time

5. CONCLUSION AND FUTURE ENHANCEMENT

In this work, smoothed dilated CNN is used for generation of the density map form single image and estimate the count of people. Compared to the base paper, smoothing operation using Separable and Shared convolution has increased both the counting accuracy and the quality of density map when dilation rate 2 was applied for both Part_A and Part_B of ShanghaiTech dataset. MAE and MSE of Part_A dataset have increased by 0.82% and 10.09% respectively. Similarly, the PSNR and SSIM of Part_A dataset both has increased by 1.3%. Apart from that, MAE and MSE of Part_B dataset has increased by 9.4% and 3.6% respectively. Similarly, the PSNR and SSIM of Part_B dataset has increased by 3.18% and 4.16%. We can see a significant increase in all the evaluation metrics.

This work can be extended to crowd counting and visualization in videos. Moreover, if vehicle detection and counting is employed then it can be used to develop smart traffic control system and to monitor accidents in highways. It can be extended to study animal migrations as well. Wan et. al [32] suggests the use of attention mechanism using refinement network.

REFERENCES

[1] Bo Wu and Ramakant Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1,* volume 1, pages 90–97. IEEE, 2005.

[2] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE transactions on pattern analysis and machine intelligence,* 34(4):743–761, 2011.

[3] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision,* 57(2):137–154, 2004.

[4] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05),* volume 1, pages 886– 893. IEEE, 2005.

[5] Haroon Idrees, Imran Saleemi, Cody Seibert, and Mubarak Shah. Multi-source multi-scale counting in extremely dense crowd images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pages 2547– 2554, 2013.

[6] Viet-Quoc Pham, Tatsuo Kozakaya, Osamu Yamaguchi, and Ryuzo Okada. Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. In *Computer Vision (ICCV), 2015 IEEE International Conference on,* pages 3253–3261. IEEE, 2015.

[7] Jia Wan, Nikil Senthil Kumar, and Antoni B. Chan Fine-Grained Crowd Counting, *IEEE transactions on image processing,* 30:2114–2126, 2021.

[8] V. Lempitsky and A. Zisserman. Learning to count objects in images. *Advances in neural information processing systems,* 23:1324–1332, 2010.

[9] Elad Walach and Lior Wolf. Learning to count with CNN boosting. In *European Conference on Computer Vision,* pages 660–676. Springer, 2016.

[10] Nikolaos Karianakis, Thomas J Fuchs, and Stefano Soatto. Boosting convolutional features for robust object proposals. arXiv preprint arXiv:1503.06350. 2015 Mar 21.

[11] Wang, C., Zhang, H., Yang, L., Liu, S., Cao, X. Deep people counting in extremely dense crowds. In *Proceedings of the 23rd ACM International Conference on Multimedia. MM '15, New York, NY, USA, ACM* pp 1299–1302, 2015

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In *European Conference on Computer Vision,* 2014.

[13] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition,* pages 1091–1100, 2018.

[14] Zhengyang Wang and Shuiwang Ji. Smoothed dilated convolutions for improved dense prediction. In *Proceeding of ACM International Conference on Knowledge Discovery Data Mining,* pages 2486– 2495, London, United Kingdom, Aug 19-23 2018

[15] Matthias Holschneider, Richard Kronland-Martinet, Jean Morlet, and Ph Tchamitchian. A real-time algorithm for signal analysis with the help of the wavelet transform, in *Wavelets*. Springer, 1990, pp. 286–297.

[16] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 472–480, 2017.

[17] Ryuhei Hamaguchi, Aito Fujita, Keisuke Nemoto, Tomoyuki Imaizumi, and Shuhei Hikosaka. Effective use of dilated convolutions for segmenting small object instances in remote sensing imagery. In *2018 IEEE winter conference on applications of computer vision (WACV),* pages 1442–1450. IEEE, 2018.

[18] https://developer.nvidia.com/discover/convolutionalneuralnetwork, [Online; accessed 23-September-2020].

[19] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[20] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pages 589–597, 2016.

[21] Deepak Babu Sam, Shuv Surya, R. Venkatesh Babu. Switching Convolution Neural Network for Crowd Counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pages 5744–5752, 2017

[22] Depth-wise Convolution Neural Network. https://www.geeksforgeeks.org/depth-wise-separable-convolutional-neural-networks/ [Online; accessed 1-July-2021].

[23] https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728 [Online; accessed 1-July-2021].

[24] Yu, Fisher, and Vladlen Koltun. Multi-scale Context Aggregation by Dilated Convolutions. ICLR 2016.

[25] Rajeev Yasarla, Federico Perazzi, and Vishal M Patel. Deblurring Face Images Using Uncertainty Guided Multi-Stream Semantic Networks. *IEEE Transactions on Image Processing*, 29:6251–6263, 2020.

[26] François Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[27] Hazarath Munanga, Venkata Jarugumalli. Performance Evaluation: Ball-Tree and KD-Tree in the context of MST. *In International Joint Conference on Advances in Signal Processing and Information Technology*. Springer, Berlin, Heidelberg, 2011.

[28] Hang Zhao, Orazio Gallo, Iuri Frosio, Jan Kautz. Loss Functions for Image Restoration with Neural Networks. *IEEE Transactions on Computational Imaging* Vol 3(1):47–57, 2016.

[29] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, Fei Su. Scale Aggregation Network for Accurate and Efficient Crowd Counting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018.

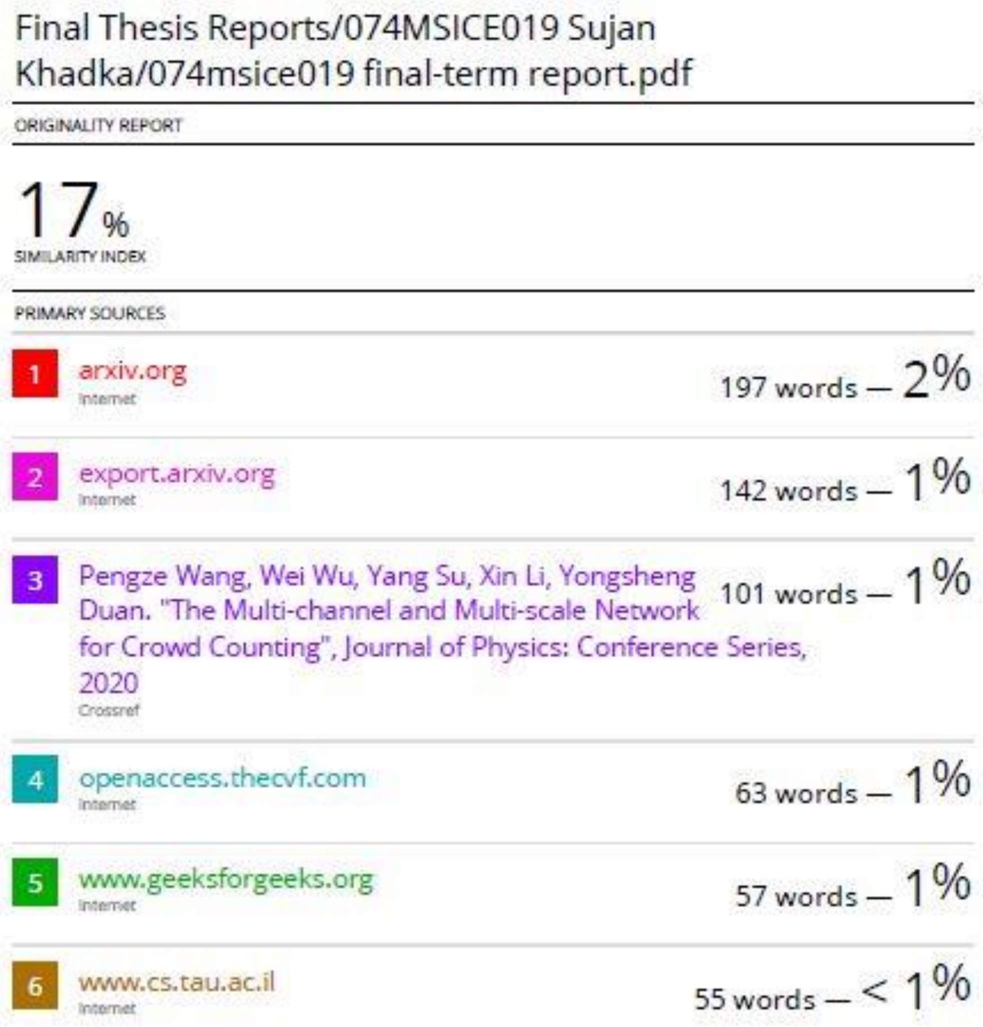
[30] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13(4) 600–612, 2004.

[31] Sindagi, Vishwanath A., and Vishal M. Patel. Generating high-quality crowd density maps using contextual pyramid cnns. *Proceedings of the IEEE international conference on computer vision*, pp. 1861 -1870. 2017.

[32] Wan, Jia, and Antoni Chan. Adaptive density map generation for crowd counting. *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 2019.

APPENDICES

Following is the snapshot of Turnitin Report.



Final Thesis Reports/074MSICE019 Sujan Khadka/074msice019 final-term report.pdf

ORIGINALITY REPORT

17%

SIMILARITY INDEX

PRIMARY SOURCES

1	arxiv.org Internet	197 words — 2%
2	export.arxiv.org Internet	142 words — 1%
3	Pengze Wang, Wei Wu, Yang Su, Xin Li, Yongsheng Duan. "The Multi-channel and Multi-scale Network for Crowd Counting", Journal of Physics: Conference Series, 2020 Crossref	101 words — 1%
4	openaccess.thecvf.com Internet	63 words — 1%
5	www.geeksforgeeks.org Internet	57 words — 1%
6	www.cs.tau.ac.il Internet	55 words — < 1%