A robust approach for people counting in dense crowd images using deep neural networks

K Sri Harsha¹, C Shoba Bindu² and E Sudheer Kumar³

¹ PG Scholar, Department of CSE, JNTUacea, Anantapuram.
² Professor, Department of CSE, JNTUA, Anantapuram
³ Research Scholar, Department of CSE, JNTUA, Anantapuram
Email: Harsha070@Outlook.com

Abstract. People counting in dense crowd images with deep neural networks are proved to be effective. The models like RCNN are able to predict the crowd count by head detection, using the CNN and selective search algorithm, but these approaches are very slow, as they involve computing convolutional operation for 2k regional proposals, involving no shared computations, besides the selective search algorithm itself is slow. In this approach a Faster R-CNN for head detection which uses a Regional Proposal Network (RPN) has been used. The region proposal network is a Fully Convolutional Neural Networks that generates region proposals, these regional proposal were fed in to RoI pooling layer and subsequently classified and localized, Thus Faster-RCNN reduces computation cost of convolutional operations by passing a image only once, sharing the convolutional operations and also using RPN for regional Proposals. Using the Faster R-CNN MAE and MSE are reduced compared to R-CNN.

1. Introduction
As with the day by day progress in the world, people from different communities collaborate and participate in conferences, debates where a large gathering of crowd takes place, a new innovation in technologies would also attract people in mass gatherings, which represent the frequent gathering of crowds, apart from that people gather in huge numbers for religious events, parties and form a crowd. Hence providing security becomes very important at this places, keeping the surveillance on the number of people attending would help for several statistical purposes and also helps in disaster management and better safety to the crowd can be provided. Estimating the number of people present in the crowd from an image, using a feature extractor machine would automate the work and make easy for humans to maintain the statistics and log records, although detecting the number of people through the features of an image as shown in figure 1 would become difficult as they work like a blind rule, because the hand engineered features like texture features of background may resemble with the person texture, hence estimating the count of a person may be more or the method may consider a person as a background when occlusion were there and ignore them.

Considering the features that can be extracted from an image of low resolution, these methods will make a huge error on predicting the count, on the other hand end-end regression techniques will be used by the models like RCNN [1] and other detection models in order to predict the count, these models make use of the neural networks to learn and predict the count from an image, although these deep neural networks make use of the regression methods they take a large amount.
of time to predict the count of people from image with almost extracting nearly 2k proposals for an image and processing the regions of the image and making recursive computation in order to estimate the count, which will be highly computational cost in-efficient and these convolutional operations were not shared. To avoid this inefficiency this approach uses a Faster R-CNN [3] model that make use of regional proposal network in order to get the regional proposals from a image passed single time through the convolutional neural network which will be more faster than the selective search algorithm. The Faster RCNN method also uses the shared convolutional operations for region proposals and detections reducing the computational cost. They nearly produce 300 regions, during the test time it is very handy as they were lesser than that of models [2] with selective search algorithm with 2k proposals having repeated computations.

The rest of the content is organized as section 2 contains the literature survey for Faster RCNN, section 3 describes the dataset preparation, section 4 describes the experimental setup, section 5 describes experimental methodology, section 6 describes results, and draw some conclusions in section 7.

2. Literature survey
Since the approach is using region proposals in order to detect the head, there were lot of hand crafted feature and learning techniques for regional proposals. So studying them will be helpful in understanding the regional proposal algorithms.

2.1. Expert and learning based methods
Detecting the object requires the features to be extracted and classified. But these hand crafted features extracted are not sufficient to predict the count in dense crowd image. There are other detection methods that consider the movement of objects in order to count but these methods fails in a large gatherings as the movement of objects will be negligible. So to detect the crowd people came up with the regression methods where the image is divided in to patches and extract the low level features. How ever these methods were successful for a less number of people in an image. The learning methods make use of end-end regression analysis like cnn to predict the count. The R-CNN [1] model with regional proposal algorithm selective search [7] is an learning problem which predicts the relation between the input and output through the regression analysis. Context aware cnn for person head detection [4] is another approach which make us of three different models based on CNN for detecting the head by using endend regression approach. Let us the look at some of the object detection and region proposal algorithm and understand their methods for better analysis.

2.2. Region proposals
There were many popular region proposal algorithms like selective search [7], edge boxes [11] and some other methods for regional proposals, these algorithms make use of learning methods like neural networks sliding window approach and hand crafted features. The selective search algorithm is a region proposal algorithm that make use of HOG feature descriptor [12] and segmentation techniques like efficient graph based image segmentation which forms a basis for selective search algorithms to merge the similar features and propose the regions. These algorithms greedily merge the pixels of image at
different scales and aspect ratios producing nearly 2k proposals, and most of proposals will overlap, which make recursive computation to be done for these region proposals which are not shared and also takes a lot of time.

2.3. Object detection
For object detection different algorithms exist in the literature that can be studied. DPM [8] a discriminatively trained part based model is an detection approach that make use of the individual parts of an object and their deformed shapes to detect an object. A region based convolutional neural networks uses an region proposal algorithm to get the regions and take these regions to classify the objects in an image and also bounding box regressor in order to localize the objects. Fast-RCNN [5] an other approach for reducing the time complexity of RCNN [1] by using the region of interest (ROI) extraction and ROI pooling methods to fast up the computation. SPP [10] which provides an architecture to speed up the R-CNN forms as a support to the R-CNN. The Fast RCNN uses the convolutional operation shared to speed up the detection process. There were some other methods like overfeat [9] which were used for object detection. Regarding the crowd counting there were methods like Multisource multiscale counting in extremely dense crowd images [6] which predicts the count based on head detection. Composition Loss for Counting, Density Map Estimation and Localization is other object detection algorithms that make use of the feature descriptors like HOG, SIFT for object detection.

3. Dataset preparation
The present approach make use of the Hollywood Head dataset [4] and Shanghai Tech [2] and UCF _CC_ 50 [2] crowd dataset, for training the proposed approach is going to use Hollywood Head Dataset and only few images from the shanghai tech Dataset, and la some of the images in the Shanghai tech are labelled using labeling tool in order to get the bounding boxes and train the network. table 1 shows the dataset information used for training.

![Figure 2. (a) represent image from Hollywood heads dataset with bounding box drawn to get coordinates shown.(b) image from Shanghai Tech dataset.](image)

| S.no | Dataset            | No of Images | Bounding Coordinates | Class Labels |
|------|--------------------|--------------|----------------------|--------------|
| 1    | Hollywood Heads    | 2,16,719     | x, y, w, h           | Head, rest   |
| 2    | Shanghai Tech      | 300          | x, y, w,h            | Head, rest   |
4. Experimental setup

**Hardware and Software:** python 3.6 with tensorflow 1.13 has been used for object detection.

**Data Preparation:** The dataset has been divided into the 80% training and 10% for validation and 10% for the test sets. The Dataset is from Hollywood dataset and Some of images from Shanghai tech.

**Optimizers:** For fine tuning of RPN stochastic gradient descent with backpropagation and learning rate of 0.0001 were used, and tuning of RCNN uses tuning end–end initially and then only the final layers. the RCNN ROI pooling layers were fine tuned with Stochastic Gradient Descent with backpropagation [5].

5. Experimental methodology

Deep Neural Networks gained a lot of attention rapidly as they were used for feature extraction and classification both. Now the same convolutional operations are used for generating regional proposals too. This paper uses the transfer learning approach which uses the.

1) Pre-trained network to extract the features.
2) And these features are used to generate the region proposals by using region proposal networks.
3) Obtained regional proposals are fed in to the Fast RCNN model RoI pooling to classify the regions and predict the bounding box.

In the further section there is detailed discussion of Faster RCNN architecture.

**Figure 3.** Faster R-CNN architecture [3].

5.1. Pre-trained cnn

The first part is going to use a pre-trained network that was trained on the image-net dataset. The network it uses is the VGG-16 network as in Simonyan and Zisserman model [14]. At first images are fed in to the vgg-16 to extract the features. Here the features are extracted from the middle layers. Since the approach is going to extract the features from middle layer of the network there is no restriction on the input size of the image. These extracted convolutional features are shared among the region proposal networks and the Fast R-CNN. During the training process convolutional features were trained corresponding to RPN and Fast R-CNN and they are fixed after that. The input to the network are images which are warped using the OpenCV library, these images propagate through 13x13 convolutional layers to produce the features.

5.2. Region proposal network

This method has been claimed as the faster region proposal method compared to the selective search. The RPN in Faster R-CNN [3] is based on Fully Convolutional Network [13]. With help of RPN the images are passed single time through a convolutional network and region proposal are generated from them which are used for detection, rather than processing multiple proposals through convolutional
layers, which are not shareable as in the R-CNN. The regional proposal network takes the n x n spatial window of shared features extracted from the last layer that is convolutional features, that mean it applies a sliding window over shared convolutional features to get the region proposals along with the class object score. The regional proposal that were proposed are going to be of different scales and ratios which were defined as anchor boxes. The approach defines that a maximum number of regional proposals for each sliding windows position is going to be k that is (k anchor boxes), these anchor boxes are considered to have the center of object falls in to them. The number of anchor boxes k is defined on the different scales and aspect ratios that were considered. These sliding window are mapped to lower dimensional (512-d features). These lower dimensional features are passed in to two fully connected layer for classification and regression. The RPN is going to classify the proposals as foreground class or background class that is foreground means whether a object is there in the proposals or background that is proposal is not going to have the object. The proposal are classified as foreground or background based on the IoU over the Ground truths and the proposals. we set the proposals as foreground if IoU over ground truths and proposal is greater than 0.7 and if IoU is less than 0.3 then it is going to be considered a back ground class. The proposal anchor boxes that were generated may be more and overlap with each other for this we are going to use Non Max Suppression (NMS) where proposals with IoU greater than some threshold are going to be eliminated and finally you get the proposals. The bounding box regressor is going to adjust the proposals, during the training the bounding coordinates generated for background class are going to be ignored. The loss function that has been used for classification is log loss and for regression is robust loss function (smooth L1). For each scale and aspect ratio we are going to have k regressor boxes. The RPN is going to use the pre-trained image-net model and uses back propagation and stochastic gradient descent to reduce the loss. The general architecture that used for RPN is a n x n convolutional layer followed by two sibling 1x1 convolutional layers as shown in figure 3.

5.3. Fast R-CNN
The region proposals that were generated were fed to the Fast-RCNN network that uses the RoI pooling. The proposals were converted into the fixed size by using RoI pooling and are classified by the softmax layer and the network follows the bounding box regression to solve the localization errors. The fine tuning methods are applied at first end-end and then only to the unique layers of the RCNN. To understand the tuning of fast RCNN look at the Fast RCNN [5,3,10].

5.4. Training process
For the training Process Hollywood Head Dataset is going to be used. The input image is fed in to an convolutional network (vgg-16) and get the features extracted. The features along with the Ground Truth (Bounding Box Co-ordinates) are then passed to the region proposal network to get the anchor boxes and the bounding box coordinates. These anchor boxes along the with generated convolutional features are fed in to the Fast-RCNN model to get the classification of the object and generate the bounding boxes. Fine tuning of the network goes in an alternate way first fine tuning the RPN in an end-end manner is done and then the Fast R-CNN network. Later the Fast R-CNN is used as initialization of RPN network and then fine tune unique layers of RPN and Fast R-CNN, until predictions were correct.

5.5. Implementation details
The approach as said uses a VGG-16 to extract features, after passing through the convolutional layers. These backbone feature are given to the regional proposal network, a 2x2 sliding window is applied over the feature map in RPN, so that details of image were considered more during the testing, where the dense crowd images were passed. The scales the approach considers is 64,128 and 256 and with the aspect ratios of 1:1,1:2 and 2:1. The IoU for the non max suppression is going to be high, since the dense crowd have small heads where keeping less IoU would eliminate some region proposals. The network is trained for 40 epoch for RPN and the Fast R-CNN and later the convolutional layers were fixed and unique layer of RPN and Fast R-CNN were trained. During the testing the image is passed through the
network and region proposal were given by RPN and head were detected and classified by Fast R-CNN. The figure 4 shows the block diagram during the testing where ground truth for bounding box were disabled.

![Block Diagram](image)

**Figure 4.** The Faster R-CNN [3] during the testing with ground truths disabled.

### 6. Results

The results for some of the images by Faster R-CNN model are shown in the figure 5.

![Images](image)

**Figure 5.** (a),(b) are results of head detection on some of the images of test set.

The approach have evaluated the Mean Absolute Error (MAE) and the Mean Squared Error (MSE) on Shanghai Tech and UCF CC 50 datsets. The methods corresponds to the robustness and accuracy of the algorithm you can see in table 2 and table 3 that the proposed model has performed better than the CNN and R-CNN model in the case of Shanghai Tech dataset and UCF cc 50 even though the errors were closed to the existing system the predictions of the count is much more faster and the model performed well over the R-CNN.

The below equations provide the calculations for MAE and MSE here $y_i$ and $y_o$ are the predicted and ground truth of the respective outputs, in this context of the paper $y_i$ and $y_o$ are the predicted count of the heads and ground truth count of the heads.

\[
MAE = \frac{\sum_{i=0}^{n} (y_i - y_o)}{n} \tag{1}
\]

\[
MSE = \frac{\sum_{i=0}^{n} (y_i - y_o)^2}{n} \tag{2}
\]
Table 2. Comparisons on Shanghai tech dataset.

| Method                          | Part A |     | Part B |     |
|---------------------------------|--------|-----|--------|-----|
|                                 | MAE    | MSE | MAE    | MSE |
| Zhang et al., 2015 [15]         | 181.8  | 277.7 | 2.0    | 49.8|
| Zhang et al., 2016 [17]         | 110.2  | 173.2 | 26.4   | 41.3|
| Marsden et al., 2016 [18]       | 126.5  | 173.5 | 23.76  | 33.12|
| CNN                             | 354    | 498  | 81     | 118 |
| RCNN (Without Weighted Average) | 254.32 | 395.5 | 60.4   | 79.8|
| RCNN (With Weighted Average)    | 241.6  | 369.6 | 43.2   | 61.4|
| Faster R-CNN                    | 235.7  | 350.4 | 39.7   | 58.1|

Table 3. Comparision on UCF CC 50 dataset.

| Method                                           | MAE   | MSE   |
|--------------------------------------------------|-------|-------|
| Rodriguez et al., 2011 [19]                     | 655.7 | 697.8 |
| Lempitsky and Zissermann, 2010 [21]              | 493.4 | 487.1 |
| Idrees et al., 2013 [6]                          | 419.5 | 541.6 |
| Zhang et al., 2015 [15]                          | 467.0 | 498.6 |
| Hu et al., 2016 [16]                             | 431.5 | 438.5 |
| Zhang et al., 2016 [17]                          | 377.6 | 509.1 |
| Marsden et al., 2016 [18]                        | 338.6 | 424.5 |
| Chang et al., 2016 [20]                          | 270.3 | -     |
| CNN                                              | 1242  | 1568  |
| R-CNN (Without Weighted Average)                 | 1161.9| 1497.8|
| R-CNN (With Weighted Average)                    | 1085.6| 1427.1|
| Faster R-CNN                                     | 1069.7| 1362.9|

7. Conclusion
The object detection algorithms like R-CNN uses a Region proposals like selective search which are very slow and involving non shared convolutional operations. This paper has used a Faster RCNN which proved to have less error (MAE,MSE) in context of crowd counting and also faster than the existing approaches by sharing convolutional operations. How ever the Faster RCNN are not that much fast we can go for some other methods which may be reliable in real time.

References
[1] Girshick R, Donahue J, Darrell T and Malik J 2014 Rich feature hierarchies for accurate object detection and semantic segmentation IEEE Conference on Computer Vision and Pattern Recognition.
[2] Shami M B, Maqbool S, Sajid H, Ayaz Y and Cheung S C S 2018 People counting in dense crowd images using sparse head detections IEEE Transactions on Circuits and Systems for Video Technology 29(9) pp 2627-36.

[3] Ren S, He K, Girshick R and Sun J 2016 Faster r-cnn: Towards real-time object detection with region proposal networks IEEE transactions on pattern analysis and machine intelligence 39(6) pp 1137-49.

[4] Vu T H, Osokin A and Laptev I 2015 Context-aware CNNs for person head detection IEEE International Conference on Computer Vision pp 2893-901.

[5] Girshick R 2015 Fast r-cnn IEEE international conference on computer vision pp 1440-48.

[6] Idrees H, Saleemi I, Seibert C and Shah M 2013 Multisource multi-scale counting in extremely dense crowd images ICCV pp 2547–54.

[7] Uijlings J R, Van De Sande K E, Gevers T and Smeulders A W 2013 Selective search for object recognition International journal of computer vision 104(2) pp 154-71.

[8] Felzenszwalb P F, Girshick R B, McAllester D and Ramanan D 2009 Object detection with discriminatively trained part-based models IEEE transactions on pattern analysis and machine intelligence 32(9) pp 1627-45.

[9] Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R and LeCun Y 2013 Overfeat: Integrated recognition, localization and detection using convolutional networks. (arXiv preprint arXiv:1312.6229).

[10] He K, Zhang X, Ren S and Sun J 2015 Spatial pyramid pooling in deep convolutional networks for visual recognition IEEE transactions on pattern analysis and machine intelligence 37(9) pp 1904-16.

[11] Zitnick C L and Dollár P 2014 September Edge boxes: Locating object proposals from edges In European conference on computer vision pp 391-405.

[12] Dalal N and Triggs B 2005 June Histograms of oriented gradients for human detection 2005 IEEE computer society conference on computer vision and pattern recognition Vol 1 pp 886-93.

[13] Long J, Shelhamer E and Darrell T 2015 Fully convolutional networks for semantic segmentation IEEE Conference on Computer Vision and Pattern Recognition.

[14] Simonyan K and Zisserman A 2015 Very deep convolutional networks for large-scale image recognition in International Conference on Learning Representations.

[15] Zhang C, Li H, Wang X and Yang X 2015 Cross-scene crowd counting via deep convolutional neural networks In Proceedings of the IEEE conference on computer vision and pattern recognition pp 833-41.

[16] Hu Y, Chang H, Nian F, Wang Y and Li T 2016 Dense crowd counting from still images with convolutional neural networks. Journal of Visual Communication and Image Representation 38 pp 530-9.
[17] Zhang Y, Zhou D, Chen S, Gao S and Ma Y 2016. Single-image crowd counting via multi-column convolutional neural network. In Proceedings of the IEEE conference on computer vision and pattern recognition pp. 589-97.

[18] Marsden M., McGuinness K, Little S and O'Connor N E 2016. Fully convolutional crowd counting on highly congested scenes. arXiv (preprint arXiv:1612.00220).

[19] Rodriguez M, Laptev I, Sivic J and Audibert J Y 2011 Density-aware person detection and tracking in crowds IEEE International Conference on Computer Vision pp 2423-30.

[20] Shang C, Ai H and Bai B 2016 End-to-end crowd counting via joint learning local and global count ICIP IEEE pp 1215–19.

[21] Lempitsky V and Zisserman A 2010 Learning to count objects in images NIPS pp 1324–32.