International Journal of Engineering & Technology, 7 (3.6) (2018) 74-80

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



Vision Based Algorithm for People Counting Using Deep Learning

M. Padmashini[1*], R. Manjusha[2], Latha Parameswaran[3]

[1]*Department Of Computer Science And Engineering, Amrita School Of Engineering, Coimbatore. Amrita Vishwa Vidyapeetham, India.*
[2]*Department Of Computer Science And Engineering, Amrita School Of Engineering, Coimbatore. Amrita Vishwa Vidyapeetham, India.*
E-Mail:R_Manjusha@Cb.Amrita.Edu
[3]*Department Of Computer Science And Engineering, Amrita School Of Engineering, Coimbatore. Amrita Vishwa Vidyapeetham, India.*
E-Mail:P_Latha@Cb.Amrita.Edu
**Corresponding Author E-Mail:Cb.En.P2cvi16010@Cb.Students.Amrita.Edu*

Abstract

Estimating the number of people in a particular scene has always been an important topic of research in computer vision and digital image processing. People counting has wide applications in scenario ranging from analyzing the customer's choice and improving the quality of service in retail stores, supermarkets and shopping malls to managing human resources and optimizing the energy usage in office buildings. While there exists algorithms for counting people in a scene, some algorithm have set their benchmark in performance with respect to efficiency, flexibility and accuracy. In this paper, an attempt has been made to perform people counting using Deep Neural Networks (DNN) on comparison with existing image processing based algorithms like Histogram of Oriented Gradients with Support Vector Machine (HoG with SVM), Local Binary Pattern (LBP) based Adaboost classifier and contour based people detection. The proposed DNN based approach has higher accuracy at 90% and less false negatives.

Keywords: *People counting, deep neural network, HoG, LBP.*

1. Introduction

There are many ways for people detection and counting in a scene. Manual counting is not a good approach as most of the public places like malls and shopping centres have dense crowd. People counting systems mostly used today in public places are based on thermal sensors and tally counters. These systems can count only people passing through a passage and cannot be extended to wider area. Now-a-days these systems are also becoming unreliable due to occlusion. When two people walk through the tally counter at the same time, they produce inaccurate results. Thus research has focus towards vision based counting as it is relatively inexpensive and provides more information, does not have a direct contact with the people who move.

The main idea behind video based people counting is detection and tracking. Detection involves locating the people in the scene and tracking is done after detection to determine the location of people along the video sequence. The existing methods on video based people counting are background modelling, body part detection and body shape detection. In background modelling, the foreground is obtained by subtracting the background from the particular frame, but this method fails due to occlusion. Body part detection uses the method of dividing the body of the person into separate parts as head, arms, upper body and legs. Each part is detected separately and using reconstruction it is determined whether the detection is a person or not. This method does not work when body parts of two people are connected together or behind each other. In body shape detection, the upper body, arms and legs are approximated into rectangular or ellipse of different sizes. The rectangular or ellipse is used to match with the body parts in the frame. A relationship is then modelled with these shapes to detect the person in the frame.

2. Literature Survey

In this section various existing people counting techniques have been discussed.

The indirect feature based approach adopted by Riachi et al. [1] for people detection uses a Hessian matrix based SURF algorithm. After acquiring the frames from the video the computation was made faster by converting the original images into integral ones using box type convolution filters for pre-processing. Blobs or interest points on the image were then detected. Assuming that the people in the scene were moving, static points were extracted and the irrelevant interest points were removed. Adaptive Rood Pattern Search block technique had been used for estimating motion vectors as it performs better than all other block matching algorithm in terms of computation time and accuracy. A linear regression has been used to estimate the count of people in the frame.

Li et al. [2] utilizes head as a feature for counting the people in the scene. A texture based approach that works only for dataset obtained from camera placed over head was employed. After acquiring frames from the video dataset, the foreground had been extracted from the image so that further processing was done only

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

on the moving objects. The foreground regions were then optimized using morphological operations like closing and opening so that the noises were removed and holes were filled. The foreground regions were used as a mask for obtaining the candidate region (region of interest) from the frame. For detection, features were extracted using Local Binary Pattern descriptor and the head regions were classified using Adaboost classifier that has been trained offline. A counting line was defined by the user to decide upon whether the head should be counted or not.

The method adopted by Wang et al. [3] uses a Mixed Gaussian Model (MGM) to eliminate the background and extract only the foreground region to distinguish the moving objects in the image. Morphological processing like dilation and erosion were employed to reduce the noise. Various features like edge, segment, and texture were extracted from the foreground. For improving the performance, HoG features were added to the system to extract more meaningful features of the people. All these features were given as input to Support Vector regression. Regression function aids in finding the relationship between the foreground pixels and people count.

In [4] Baozhu et al., have designed a method for counting people in an ATM. In this system the camera was fixed and background has been stable. The proposed method combines background subtraction and Gaussian model method to extract the foreground image. Then binarizing and morphological operations were done to get the contours of people. The obtained contour was taken as a person if number of pixels in the contour was more than a certain threshold, and the numbers of contours were the number of people.

Pongsakon and Suteera [5] have presented an automatic people counting system for indoor environment. It was a monocular vision system that uses an IP camera mounted in top view position to provide video stream of people walking through a door. The system composed of 3 parts: image capturing, people detection, people counting. The image data were transferred to detect and track individual person using background subtraction, morphological operation, and blob detection. The counting algorithm estimates the moving direction of the detected people using a region based approach.

The IoT based approach for counting and tracking people used by Dimitris et al. [6], in the context of a smart room, where several cameras have been installed, initializes the system by motion detection service where the existence of motion within the video stream was checked at regular intervals. When the motionless state was detected, the background subtraction service was triggered where mixture of Gaussian model was used to obtain the foreground image. Then the image patches (blobs) that corresponds to human were identified using blob detection service. Finally a tracking service was implemented by creating a separate list of ids that corresponds to the detected blobs. Whenever a blob appears or changes, the list was updated to reflect that change.

An innovative method for estimating the people count has been presented by Pizzo et al. [7] that operate best on cameras placed over head. Firstly, it detects the foreground by calculating the intensity difference between the frame and its background so that the motion pixels can be identified. The foreground pixels obtained were given to a sensor geometrically designed for counting people. The results were validated using dataset specially collected to account for factors like illumination and density flow.

In [8] Kuo et al., used camera with kinect sensors instead of traditional camera so that the depth information of the image was obtained for counting the people who pass through the supervised area. This method used thumbnail (nearest neighbour interpolation) pre-processing to reduce the image resolution which improved the computation speed. The foreground was extracted from the image using depth slicing filters. In pre-processing, thumbnail (nearest neighbour interpolation) reduced the resolution of the image to cut down the computation time and depth slicing filters out the foreground object from background. People were detected using canny edge detection and Hough circles identified the circles in the depth image. Shoulder detection was then used to decide whether the circle was people or not.

Zhuo et al. [9], employs a method for counting people in crowded scenes without performing individual detection. The feature vectors obtained by extracting various features like segment (area, perimeter, blob count), edge (length and orientation) and texture (energy, homogeneity, entropy) were given as input to gradient boosted trees. Firstly, the features of crowd moving in different directions were extracted from the corresponding segmentations using segment features (area, perimeter, blob count), internal edge features (edge length, edge orientation), and texture features (energy, homogeneity, entropy). After obtaining the feature vectors, it was given as an input for training the gradient boosted trees which was then used for classification. A non-linear function maps the feature vectors and people count.

The method adopted by Perng et al. [10], estimates the people count using a single camera that fixed overhead. The foreground was extracted using background subtraction so that only the moving objects were then processed. Morphological operations like opening and closing were used to eliminate the noise and fill the holes. Connected component method separates each object from the foreground. Features like area, aspect ratio, and velocity are used for people detection. An effective counting region and line was defined by the user for counting people.

In Cai et al. [11], a people counting system was designed, especially for crowded scenes. Firstly the data was taken as video from the camera installed directly on the roof. The head information was then extracted from the input video taken from the camera installed directly on the roof. Selecting the region of interest makes the computation faster and eliminates the uninterested areas. Local Binary Pattern features were extracted from the data and were fed into the Adaboost classifier. The resultant feature vector was used for target detection. The head detected was then tracked by a Haar matching model. The head detection and tracking combined together, was used to count the number of people in the frame.

Xu et al. [12] presented an efficient regression based people detection approach that makes any the video surveillance system much more efficient. Regression based methods detects people based on the features extracted from the scene. The method employed divides the foreground and background based on the dominant colours in the scene. This makes it unconstrained to static image and video. Clustering and morphological operations performed not only improves the results but also enhances the efficiency. Finally an association was formed between the number of foreground pixels and number of people from the trained Support Vector Regression (SVR).

3. Proposed Work

The first step towards the process of people counting is people detection. Although there are many algorithms existing for people detection, some of them often give high false positive rate while others give high false negative rate. The proposed work aims at using Deep Neural Network(DNN) for people detection and counting and the results has been compared with classic algorithms of people detection such as HoG with SVM, LBP based Adaboost classifier, and contour based people detection. The interest of this work lies in measuring the performance of DNN compared to other algorithms on various datasets and under different scenarios.

Contour based people detection

The contour based people detection method is much faster and easier in terms of implementation. As human body cannot be static and make even slighter movements, motion information is taken as criteria for this method. Once the dataset is fed to the algorithm, frames are extracted and resized to reduce the processing time. Adaptive background mixture model is then applied for

background subtraction. This is done to segment the moving objects (foreground) from the static objects background).Extracting the foreground from the frame helps in processing only the region of interest. The extracted foreground contains moving objects which can be either human or non-human such as automobiles. Dilation, a morphological operation is applied on the foreground to fill in the holes. Given the dilated image, contours are detected to find the outlines of the moving object. The contour area of each object detected is estimated and a range of area is set as threshold to remove the false positives. The area is defined in such a way that small irrelevant contours and very large contours of automobiles are eliminated. The objects in the defined range are identified as human and a bounding box is drawn over it. The output is shown in Fig.2.It can be seen that not all people are detected by the algorithm. The contour approach for detecting people is described in the Fig.1 below:

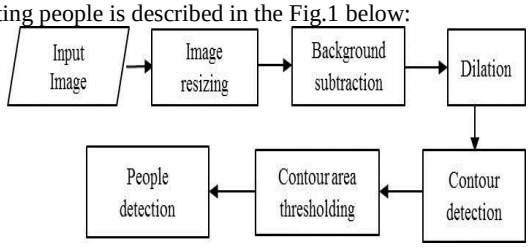


Fig. 1: Architecture of contour based people detection

.



(a) (b)

Fig.2: Contour based people detection: (a) input image; (b) output image.

HoG with SVM

Even though Histogram of Oriented Gradients descriptor is a decade old, it has proved to be a classic and traditional algorithm for people detection. Detailed step by step description of HoG with SVM approach is as follows:

- Two training dataset are taken in which one set contains only the images of the target, i.e. people to be detected while the other contains images that does not have any people.
- Certain numbers of samples say P are taken from the positive training dataset and sampled. Similarly N negative samples taken from negative training dataset are sampled. Always negative samples taken should be much greater than that of positive samples taken (N>>P).
- HoG features are extracted from these P positive samples and N negative samples.
- For classification, a linear SVM is trained on these positive and negative samples.
- For each input, HoG features are extracted and given to the pre trained people detection classifier.
- A bounding box is drawn over the detection from the SVM classifier. Some detection may have one or more bounding box over it. The output can be seen in Fig.5 below.
- Non maximum suppression is applied to remove those overlapping bounding boxes.



(a)(b)

Fig.3: Non maximum suppression: (a) before; (b) after.

The framework of HoG with SVM algorithm is described in the figure below:

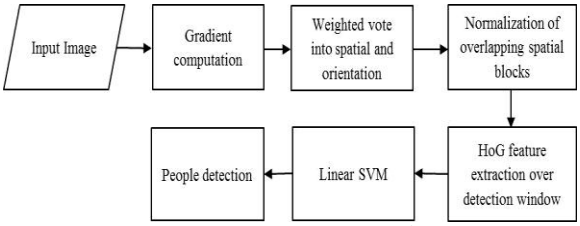


Fig.4: Architecture: HoG with SVM algorithm for people detection

.



(a) (b)

Fig.5: HoG with SVM: (a) input image; (b) output image.

LBP based Adaboost classifier

Although HoG has been a widely accepted as an efficient feature descriptor to extract the shape information in an image, LBP is also a powerful texture descriptor that is computationally efficient and provides detailed information of the image. Once the features are extracted the classifier is trained by combining a series of weak classifiers so that a strong classifier is obtained.

- Two training dataset are taken in which one set contains only the images of the target, i.e. people to be detected while the other contains images that does not have any people.
- Certain numbers of samples say P are taken from the positive training dataset and sampled. Similarly N negative samples taken from negative training dataset are sampled. Always negative samples taken should be much greater than that of positive samples taken (N>>P).
- The description file for the positive and negative samples is created. Every line in the text file corresponds to each image. Every line in the positive description file contains following fields: the image name, followed by the number of people to be detected in the image, which is followed by the x, y coordinates of the location of the people in the image.
- The description file is created using markup language.
- The positive samples are then packed into a vector file.
- The classifier is then trained using following arguments: training data, vector file, negative descriptor file, number of stages, memory allocated, width, and height.
- Once the training of the classifier is done, the training data created is converted to an xml file.
- For every input image, LBP features are extracted and given to the trained Adaboost classifier.
- A bounding box is drawn over each detection from the LBP based Adaboost classifier. Some detection may have one or more bounding box over it. The output can be

seen in Fig.6.

- Non maximum suppression is applied to remove those overlapping bounding boxes.



(a) (b)

Fig. 6: LBP based Adaboost classifier: (a) input image; (b) output image

Deep Neural Network

Existing people detection methods based on deep learning include Faster R-CNNs, You Look Only Once (YOLO), Single Shot Detectors (SSDs). Accuracy and processing time are two main factors that determine the performance of the algorithm. The processing of YOLO algorithm is much faster than the faster implementation of R-CNNs but the accuracy is compromised. Single Shot Detectors (SSDs) are balances between these two factors. They are also more accurate than YOLO. Building object detection networks using existing network architecture, such as VGG or ResNet, can be very large in terms of memory consumption. They are also unsuitable for resource constrained devices due to their sheer size and resulting number of computations.

MobileNets developed by Howard et al. [13], use depth wise separable convolution in which the convolution is split into two stages:

Stage 1: A 3 * 3 depth wise convolution followed by batch normalization and ReLU

Stage 2: A 1 * 1 point wise convolution followed by batch normalization and ReLU

Splitting the network in two stages significantly reduce the number of parameters in the network and are resource efficient but are not much accurate as others.So combining MobileNets architecture and Single Shot Detector framework provide much faster and efficient deep learning based people detection.

Training our own dataset:

The mobile SSD was trained for our own dataset was fine-tuned on PASCAL VOC reaching 69.6% mean average precision. Around 10000 images were taken from the dataset out of which 60% are negative samples and 40% are positive samples. The positive samples contain the images of people to be detected whereas the negative samples do not contain any people in it.

Training of our own dataset was done through the following steps:

1. Converting our own dataset to Lightning Memory-Mapped Database (lmdb).In this way the data is prepared for training.
2. Now the model comprising MobileNetSSD definition and the solver constituting information about gradient descent is set up.
3. A symbolic link is created between the neural net and lmdb files.
4. The gradient descent parameters are specified in the solver.prototxt file. Some of the parameters defined in the file are:

Table I: Training Parameters

Parameter	Value
Test iteration	673
Test interval	10000
Base Learning Rate	0.0005
Learning Rate Policy	multistep
Gamma ratio	0.5
Step interval	20000
Maximum iterations	120000
Weight decay	0.00005
Solver mode	GPU

5. The training is started by calling the Caffe binary by only specifying the solver file as the model is already mentioned in the solver and data is mentioned in the model file.
6. Once the training is finished, the testing is done using deploy.prototxt by adding an input layer. The input given is as follows:

```
input: data
input_shape {
dim: 1
dim: 3
dim: 300
dim: 300}
```

Once the training model is ready the people are detected using DNN as described in the architecture diagram below:

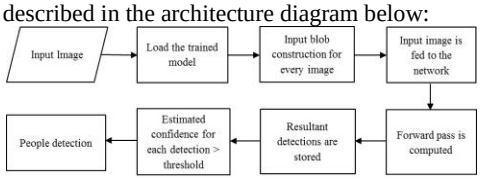


Fig.7: Architecture: DNN algorithm for people detection.

The image below depicts the output of DNN when an image is fed to the network.



(a) (b)

Fig. 8: Deep Neural Network: (a) input image; (b) output image

4. Experimental Analysis

In this section, the dataset acquired to perform the experiment, the development environment, evaluation metrics used to estimate the performance of the algorithm and its results are explained.

Dataset specification

The experiment is performed in an outdoor environment under daylight, a condition where the illumination is constantly varying. The dataset is acquired as a live stream video from a camera placed at an inclined position, so that a large extent of area is covered. The scope of coverage in turn is useful in obtaining even the large deviations in people density flow. Apart from density flow and camera orientation, partial or full occlusion that occurs due to the presence of another human or object also plays a pivotal role in determining the performance of the algorithm. The video is obtained in MPEG format, comprising the information of students, faculties and vehicles traversed through the entrance gate in the daytime during a weekday. The quality of information from live dataset is much better than the open dataset available. It also provides a more challenging environment for estimating the performance of the algorithm.



Fig.9: Sample input image from the dataset.

Development environment

All the algorithms were implemented in Open Source computer Vision library using python interface under Ubuntu 16.04 environment. The machine used was a 64 bit Intel Core: i5-3210M processor with 4.00GB RAM.

Evaluation metrics

The ground truth of the dataset is obtained manually to determine the accuracy of the algorithm. People detected using the algorithm can be categorized into: True Positives, which are the people that are correctly classified by the algorithm, False Negative which are non-recognized human and False Positive also known as False Alarm, which are the non-humans that were erroneously classified by the algorithm as human. Another metric, True Negative which is non-humans correctly classified as non-humans is not considered because they are not required for the algorithm to be identified.

Six metrics were used to measure the performance of the algorithm. The metrics are as follows:

- Accuracy, in the context of people detection is a measure that estimates how correctly the algorithm. It is the proportion of true positives (humans) among the total number of objects present in the frame.

$$Accuracy = \frac{TruePositive + TrueNegative}{CompleteGroundTruth} \quad (1)$$

- Precision, also known as detection rate measures the proportion of people that were correctly identified by the algorithm among the total detection. It is defined as follows:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (2)$$

- Recall, also known as True Positive Rate or Hit rate measures the proportion of people that were correctly recognized by the algorithm among the total number of people in the frame.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (3)$$

- F-measure**,** estimates the accuracy of the algorithm.

$$F - Measure = \frac{2 * Precision * Recall}{Precison + Recall} \quad (4)$$

- False Positive Rate measures how well the algorithm correctly rejects false positives. The algorithm with less false positive rate relative to another algorithm is much preferred.

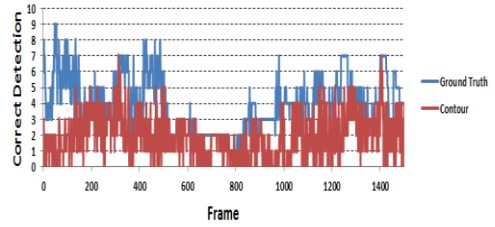
$$FalsePositiveRate = \frac{FalsePositive}{FalsePositive + TrueNegative} \quad (5)$$

- False Negative Rate also known as Miss Rate measures the proportion of people that were not recognized by the algorithm among the total number of people in the frame.

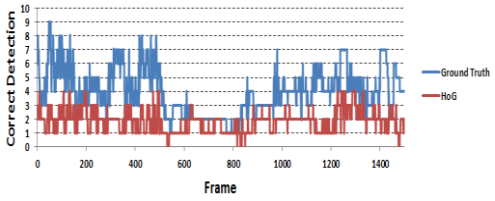
$$FalseNegativeRate = \frac{FalseNegative}{FalseNegative + TruePositive} \quad (6)$$

Observations and results

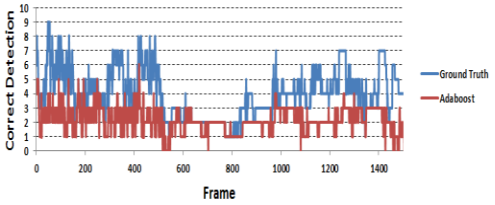
This section explains the results and observations of people detection through each algorithm. As already said, the camera is placed in inclined position and the flow density of people varies with time from low (1-2 person per frame) to high (9-10 person per frame).The correct detection vs. ground truth for every frame is represented in the graph below where the x-axis measures the frame number and y-axis measures the correct detection by the algorithm. For better visualization, only 1500 frames are in the graphical plot and each algorithm is depicted separately. Besides, the ground truth is estimated manually for each frame.



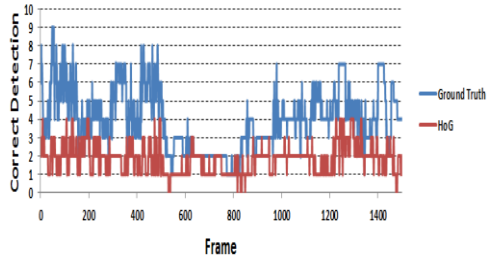
(a) Contour vs. Ground Truth.



(b) HoG with SVM vs. Ground Truth.



(c) LBP based Adaboost vs. Ground Truth.



(d) DNN vs. Ground Truth.

Fig.10: Detection curves for each algorithm.

It can be clearly visualized from the graph that the detections of DNN are much closer to the ground truth among the other three algorithms. HoG with SVM and LBP based Adaboost classifier shows similar results leaving behind contour based classifier whose detection are not close at any time to the ground truth. This shows the hit rate of DNN is very high.

Table II. shows the results obtained from feeding the dataset to each algorithm. It can be observed that DNN and LBP based Adaboost classifier correctly identifies more number of people than HoG with SVM and Contour based classification. Besides true positives, false positives and false negatives are two important measures to look into. HoG with SVM performs best when compared to LBP based Adaboost classifier and contour based people detection in terms of false positive and false negative measures. While HoG with SVM is said to perform better than contour based people detection and LBP based Adaboost classifier, it is very evident that DNN outperforms it all relatively with respect to true positive, false positive and false negatives measures.

Table II: Comparison of Measures from Confusion Matrix Ground Truth: 6074

ALGORITHM	MEASURE			
	True Positive	False Positive	False Negative	True Negative
Contour	2926	1411	3148	0
LBP based Adaboost	3289	1345	2785	0
HoG with SVM	2771	30	3303	0
DNN	5632	121	442	0

The metrics from Table III are aggregated based on the results from Table II.

From Table III, it can be inferred that people detection through contour is worst in terms of accuracy and precision compared to all the other three algorithms. The reason behind it is when two person walk close to each other, the dilation applied before finding contours, makes the detection as a single person. Besides HoG features are more robust than LBP features, which makes the HoG with SVM accuracy to be slightly better than LBP based Adaboost classifier.

It must be noted that the hit rate and miss rate of LBP based Adaboost classifier is better than HoG with SVM. This is because LBP features are rotation invariant. Thus disorientation of moving people does not affect LBP based Adaboost classifier and more number of people are recognized by the algorithm.As mentioned before, DNN exceeds its performance in terms of accuracy and recall.

In the same way its false negative rate is very much less when compared to all the other three algorithms. The performance of DNN is much better than the traditional and classical HoG with SVM because Linear SVM can be considered as a single layer neural network, which even on feeding HoG features fails to identify the target from the dataset which in turn can be achieved much easily by simply training a multilayer perceptron.Finally by analyzing the evaluation metrics altogether it can be seen that HoG with SVM has lowest false positive rate.

An algorithm with low false positive rate is always preferred. However it cannot be denied that accuracy and precision plays a much vital role for real time applications. This makes the traditional and classical algorithms like HoG with SVM and LBP based Adaboost classifier to be suppressed by DNN, the trends of recent times.

Table III: Evaluation of each Algorithm

ALGORITHM	METRIC					
	Accuracy	Precision	Recall	F-Measure	False Negative Rate	False Positive Rate
Contour	0.407573	0.621808	0.5255389	0.520639	0.47446	0.61195
LBP based Adaboost	0.483108	0.735262	0.60484	0.619679	0.395159	0.61195
HoG with SVM	0.50455	0.98074	0.511995	0.638905	0.488005	0.019698
DNN	0.90533	0.98553	0.91523	0.939336	0.08477	0.048588

In the experiment, nearly 20 videos with a rate of 15frames/second were taken for analysis. Each video constituted nearly 300 frames. The performance of each algorithm is further verified using "Receiver operating characteristic (ROC) curve" and "Detection error trade off (DET) curve".

ROC curve is said to be relative because it compares the True Positive Rate with that of False Positive Rate.

An algorithm can be said to have a good performance if the ROC has a steep rise. It can be observed in the graphical plot that DNN curve has a very sharp rise while all the other three algorithm fall rapidly. Among those three algorithms, contour based people detection has sharp fall when compared to LBP based Adaboost classifier and HoG with SVM.

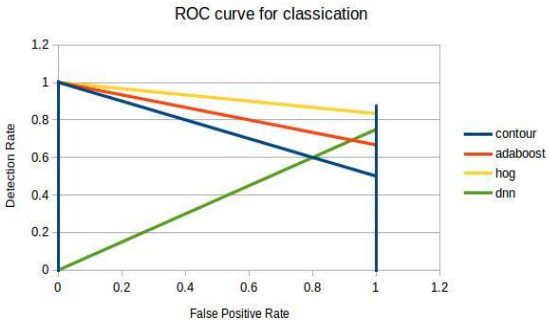


Fig.11: ROC curve for people detection.

DET curve, a variant of ROC curve, reflects the miss rate to false positive rate per image. It denotes that a good classifier has the lowest curve. The reason behind it is that detection rate is better for lower curves. Here DNN curve is lower in comparison to the other three algorithms. In the same way contour based people detection algorithm alone is a rising curve which indicates that its detection rate is worst.

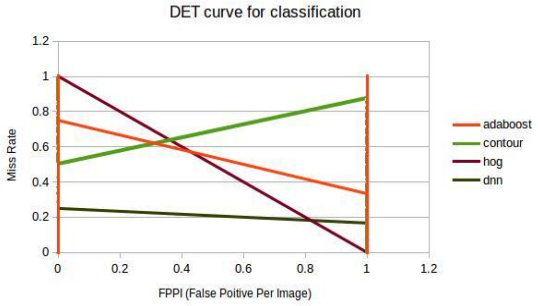


Fig.12: DET curve for people detection.

5. Conclusion

In this paper, DNN based people counting has been performed and compared with three different algorithms for people detection and the results are analyzed and recorded. The performance of each algorithm is tested under conditions like fluctuating density flow of people and presence of occlusion. They have been compared based on various evaluation metrics and the performance of each algorithm has been represented graphically using ROC and DET curve. From the evaluation metrics and performance graph, it can be clearly seen that DNN outperforms all the other three traditional and classical algorithms like HoG with SVM, LBP based Adaboost classifier and contour based people detection in terms of accuracy and precision which is much required for real time applications. Although contour based people detection is much easier to implement it has shown the worst performance among the other three classifiers. This study shows that the traditional method of extracting features for people detection can be surpassed by neural networks. However the epoch time for training a neural network is relatively higher to that of other classifiers. The need for faster people detection systems is incessant. So our future work involves developing a highly accurate and faster people detection system compatible to real time applications.

References

[1] Riachi S, Walid K & Hanna G, "An improved real-time method for counting people in crowded scenes based on a statistical approach", *IEEE 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, Vol.2, (2014), pp. 203-212.

[2] Li B, Jian Z, Zheng Z & Yong X, "A people counting method based on head detection and tracking", *IEEE International Conference on Smart Computing (SMARTCOMP)*, (2014), pp. 136-141.

[3] Wang Y, Huicheng L, Pei C & Zhenzhen L, "Counting people with support vector regression", *IEEE 10th International Conference on Natural Computation (ICNC)*, (2014), pp.139-143.
[4] Baozhu Z, Zhu Q & Xing Y, "People counting system based on improved Gaussian background model", *IET International Conference on Smart and Sustainable City* and *Big Data (ICSSC)*, (2015), pp.118-122.
[5] Pongsakon B & Suteera P, "Robust people counting using a region-based approach for a monocular vision system", *IEEE International Conference on Science and Technology (TICST)*, (2015), pp.309-312.
[6] Dimitris S, Evaggelos S, Giorgos S & Theodoros G, "Counting and tracking people in a smart room: An IoT approach", *IEEE 10th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP)*, (2015), pp.1-5.
[7] Del PL, Pasquale F, Antonio G, Gennaro P & Mario Vento, "A versatile and effective method for counting people on either RGB or depth overhead cameras", *IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, (2015), pp.1-6.
[8] Kuo JY, Guo DF & Tai Yu L, "People counting base on head and shoulder information", *IEEE International Conference on Knowledge Engineering and Applications*, (2016), pp.52-55.
[9] Zhou B, Ming L & Yonggang W, "Counting people using gradient boosted trees", *IEEE Information Technology, Networking, Electronic and Automation Control Conference*, (2016), pp.391-395.
[10] Perng JW, Wang TY, Hsu YW & Wu BF, "The design and implementation of a vision-based people counting system in buses", *IEEE International Conference on System Science and Engineering (ICSSE)*, (2016), pp.1-3.
[11] Cai Z, Yu ZL, Liu H & Zhang K, "Counting people in crowded scenes by video analyzing", *IEEE 9th Conference on Industrial Electronics and Applications (ICIEA)*, (2014), pp.1841-1845.
[12] Xu J, Wu Q, Zhang J, Silk B, Ngo GT & Tang Z, "Efficient people counting with limited manual interferences", IEEE *International Conference on Digital image Computing: Techniques and Applications (DlCTA)*, (2014), pp.1-6.
[13] Howard Andrew G., Menglong Z, Bo C, Dmitry K, Weijun W, Tobias W, Marco A & Hartwig A, "Mobilenets: Efficient convolutional neural networks for mobile vision applications", (2017).
[14] https://opencv.org/