Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 165 (2019) 307–313

Procedia Computer Science

www.elsevier.com/locate/procedia

INTERNATIONAL CONFERENCE ON RECENT TRENDS IN ADVANCED COMPUTING 2019, ICRTAC 2019

Deep Learning Based Automated Attendance System

Joshan Athanesious J[a], Vanitha[a] , S. Adithya[b], C. Anirudh Bhardwaj[b], Jugat Singh Lamba[b] A, V. Vaidehi[c*]

[a] *Mardras Institute of Technology, Anna University, Chennai,60004, India.*
[b] *Vellore Institute of Technology, Chennai Campus, Chennai, 600127, India.*
[c] *Mother Teresa University, Kodaikanal, 624101, India.*

Abstract

A significant portion of the time allocated to a faculty for teaching purposes is consumed on the task of taking attendance of the students. This is an issue because it takes the valuable time of teachers which could be spent on more productive tasks such as teaching and interacting with students. In excess to the increase in chaos and loss of decorum in the classroom environment, the presence of proxy attendance also plagues the existing method of manual attendance keeping. To counter these issues, this paper proposed the Deep Learning Assisted Attendance System (DPAAS); which keeps track of students attending a particular class with the help of a continuous stream of pictures captured from a video streaming device located inside a classroom connected to the remote server. The proposed DPAAS method reduces the amount of time spent by the faculty on taking attendance, and leads to a reduction in chaos inside a classroom. DPASS is proposed handles the issues in existing systems such as multi-class identification for multiple individuals in a classroom, occlusion and differing light scenarios. The DPAAS methodology compares the results of the state of art algorithms, and uses the best fit architecture which provides the lowest false rate on evaluation. There is no need of user interaction in the proposed DPAAS. Experimental results show that the proposed DPAAS method gives 94.66% accuracy which is better than the other existing methods.

Keywords: Face recognition, Information and communication Technology,Deep Learning, Attendance system.

* Corresponding author. *Tel.: +91-9381041596*
E-mail address.: vaidehimitauc@gmail.com

1877-0509
10.1016/j.procs.2020.01.045

1. Introduction

Human beings have a limited amount of time to accomplish things. Wasting time on mundane and repetitive manual work results in a loss in productivity, diverting attention from key issues. In a quest to stop such meaningless work, we propose an automation-based approach using Deep Neural Networks to tackle the mass surveillance problem, concentrating specifically on the subset problem of increasing teaching efficiency of teachers by automating the attendance taking process. This paper proposed to reduce teacher's burden by implementing automatic facial detection and face recognition techniques using Deep Learning frameworks which gives fast and accurate result and also overcome the drawbacks of the existing methods such as multiclass identification problems, multiple occurrence of the same person, illumination variance and occlusion.

This paper is structured as follows: introduction of the automatic attendance system is described in section 1 followed by literature survey is explained in section 2. Section 3 gives complete description of the proposed DPASS and experimental result explains in section 4. Section 5 presents the conclusion.

2. Literature Survey

Deep Learning forms a subset of the Machine Learning world, wherein the data is passed through multiple layers of non-linear mappings to allow the algorithm to learn a representation. Deep Learning Models are nothing more than a function approximator capable of learning almost any and every function that depends on the input data. There are many fields inside Deep Learning, ranging from supervised regression models to reinforcement learning based artificial intelligence models. A particularly hard and hotly researched topic in Deep Learning is the topic of Face Detection and Face Recognition [13]. Large scale face recognition has largely been difficult due to the non-availability of a clean and labeled training dataset. The state-of-the-art solutions in this domain are uprooted by new innovations, partly fueled by the influx of Graphical Processing devices such as Nvidia Graphics cards which allow for extremely fast matrix operations. Historically, Image and Vision Computing has been the forte of non-deep learning-based methods such as Viola Jones [5], Clustering [17] and HOG [6] methods, but were severely hampered by noise in the data, such as Occlusion, Luminosity variance etc. Before Deep Learning, Face Recognition required Feature Engineering, Data Preprocessing [16], Output Parsing and KDD Tree Structures to enable decent performance. The compute power also means that state of art Deep Learning based Image Detection algorithms such as RCNN [1], Faster RCNN [3], Fast RCNN [2], Overfeat and YOLO [4] can execute in nearby real time on a commercially available workstation.The arrival of Convolutional Neural Networks directed to an explosion in the usage of Deep Learning techniques for Image and Vision Computing. The problems of data noise and incorrect predictions due to Occlusion, Luminosity variance etc. can be combated in Deep Learning by using Data Augmentation techniques at training time to ensure that the model is invariant to such noise, and has good generalizability.

In general Face Recognition can be constructed as a two-part problem, the first is to detect a face in the image and the second is to correctly categorize the image. This can be translated into an Object Detection problem and a Object Classification Problem. The problem statement for an object detection framework can be constructed as an issue of image localization, meaning the neural network has to know in which part of the image lies the object to be detected, and secondly the confidence with which the Neural Network predicts that the object belongs to a particular class.

There are many approaches for solving the problem, and generally there are two main methods which can be categorized as: Branch Networks and Single Network. In Branch Networks, as the name suggests, use multiple Neural Network architectures to perform the different sub-tasks required to accomplish the pen-ultimate task. The branched sub-networks are decoupled in nature, each of them trained to approximate a different function. In the context of Object Detection networks, one branched sub-network works on the problem of region proposal, meaning that the network tries to learn which is the most promising region in the image that might contain the desired object to be detected. The second branched sub-network learns how to classify the detected object.

Girshick et. al [1] proposed Rich feature in hierarchical order for precise object recognition and semantic segmentation using R-CNN with deep learning. The first step of R-CNN involves extracting region from the input image, which is achieved by using techniques such as objectness and selective search [7], constrained parametric

min cuts etc. For all the experiment purpose the selective search technique has been used to extract the region proposals.

Cortes et al. [8] proposed R-CNN based Alex net as a feature extractor for objection detection and classification [8]. Once the regions are extracted, then the features are further classified by the Support Vector Machines (SVM). Similar to the pyramid type representation scheme achieved by the Bag of Words model in Natural Language Processing, the features extracted by the R - CNN model, using Convolutional Neural Networks, are represented in a rather low dimensional space capable of capturing latent information in the data. Krizhevsky et al. [9] proposed ImageNet classification with deep CNN. Once features are extracted from CNN, the resultant features are fed into class specific set of SVM whose mod as a consequence gives score for each of this class. Non-Maximal Suppression (NMS) is applied to remove the regions extracted which has lower score than the region having higher score when IoU (Intersection over Union) between these regions is higher than the threshold. Innovations in these single network-based models have led to multiple breakthrough innovations which led to multiple models beating state of the art object detections implementations. The innovations also led to model architectures capable of performing object detection in real time. Some of the popular single network architectures include but are not limited to YOLO, Single Shot Multibox Detector, YOLO 9000 etc.

Redmon et al. [10] proposed YOLO which is Unified with real-time object detection. The YOLO while in detection mode of the input size is reduced from 448 to 416. This is done to have center cell inside the feature map and it is better to predict center instead of four coordinates for large objects as it tends to occupy a large space. However, it increases number of predictions in the image with a decrease in accuracy. Redmon et al. [11] proposed YOLO9000 for better, faster and stronger object detection. YOLO9000 is an improved version of the classical YOLO model. The new YOLO9000 is better because it uses Batch-Normalization which replaces the dropout layer solving both overfitting and adding faster convergence. Moreover, it requires initialization of anchor centre.

Liu et al. [12] proposed an objection detection method using Single Shot Multibox Detector Model (SSD). This method contains a base network and an auxiliary network where base network is used for creating feature maps in the SSD model. The auxiliary network uses these feature maps to predict the bounding boxes. The auxiliary part of the network is created by adding convolutional feature layers which decrease in size allowing prediction at multiple scales. SSD network uses this added feature layer of size of $M \times N$ and p channels by applying a kernel of size $3 \times 3 \times p$ for either predicting the class score or the offsets which are measured comparative to the default box position of each feature map location. Lowe et al. [13] proposed Scalar Invariant Feature Transformation (SIFT) to extract the key points of object for object recognition. Euclidean distance-based feature matching is used to find the candidate matching features. But this method requires more time. He et al [14] proposed deep residual learning for image recognition. ResNet makes it possible to train upto hundreds or even thousands of layers. Network performance is not affected by the stacking layers used in ResNet, because it simply stacks identity mappings and the resulting architecture would perform the same. This specifies that the deeper model should not yield a learning error.

Moreover, the region proposal based decoupled architectures are slow, and have a hard time achieving convergence to a near-optimal solution. To overcome the problem of the branch network, this paper proposes a single network based deep learning method for automatic attendance system. The proposed DPAAS gives fast and accurate result when multiple object identification in multiple times, in the presence of illumination variation and partial occlusion.

3. Proposed DAAS

The goal of the Deep Learning based Automated Attendance System (DPAAS) is to robustly detect faces in different conditions. The proposed DPAAS method is multifaceted, and involves solving multiple problems which have a non-trivial solution. There are two key steps of the proposed DPAAS, first one is detecting if a face is present in a given region of the image, and the second one is recognizing the label of the person being detected in the image by the detection algorithms. The proposed DPAAS method uses a Neural Network named as single shot multi-box detector for face detection purpose and VGG network for multi-class face recognition purpose. SSD contains a base and auxiliary network where base network is used for creating feature maps in the SSD model[12]. The auxiliary network uses feature maps to predict the bounding boxes. The auxiliary part of the network is created by adding convolutional feature layers which decrease in size allowing prediction at multiple scales. SSD network uses this

added feature layer whose size is of m×n and p channels by applying a kernel of size 3×3×p for either predicting the class score or the offsets which are calculated by box position comparative to each feature map location. Default boxes are associated with every feature map cell, these boxes are like a tile the cell in a convolutional form. This makes the position of the box relative to the cell. Titled default boxes are also designed so that precise feature maps study to be receptive to specific scale. A similar kind of approach is followed for aspect ratio where the aspect ratios are {1, 2, 3, ½, ⅓ }. For the object with aspect ratio 1, another similar one is added leaving us with 6 default boxes per feature map cell. The scalar invariance is achieved by using features maps from different layers for prediction as it has low level layer information for handling small sized images but also huge context information to handle large images.

$$DPL(x,c,l,g) = \frac{1}{N}(DPL_{conf}(x,c) + \alpha DPL_{loc}(x,l,g)) \tag{1}$$

$$DPL_{IOC}(x,l,g) = \sum_{i\in Pos, m\in\{cx,cy,w,h\}}^{N} \sum^{k} x_{ij}\ smooth_{L_1}(l_i^m - g_l^m) \tag{2}$$

$$c_i^P = \exp(c_i^p) / \sum_P \exp(c_i^P) \tag{3}$$

Where the DPL is the overall loss function, DPL_{conf} is the localization loss, DPL_{loc} is the confidence loss, α is the weight term set to 1, N is the number of matched default box. l and g are the predicted and ground truth box respectively. cx and cy are the centers of default bounding box(d) for its weight (w) and height (h). c is the multi classes confidences.

The default boxes are of different sizes, aspect ratio and vary in location. For training purposes, the ground truth is allotted to particular default box space. The jaccard overlap is 0.5 for the matching of default box with ground truth. This streamlines the learning problem, permitting the network to predict high scores in event of an overlap. The x^p_{ij} term indicates the matching between ith default box to the jth ground truth box of category p, N denotes the number of corresponding default boxes. The localization loss is defined as a $Smooth_{L1}$ loss and the confidence loss is softmax. The Single Shot MultiBox Detector (SSD) was selected because SSD has given more robust results as compared YOLO9000 in case of Object detection when compared in AP(Average Precision) - 50 scores of MS COCO dataset. SSD is faster and more robust than any YOLO and branched networks, hence SSD is chosen as the network for object detection for DPAAS.In Face Detection scenario, Faces are the only objects of interest. In this case, objects which are faces can be chosen as positive examples for parameter optimization, while the other objects are treated as background and used in techniques such as Negative Hard Mining to improve the generalizability of the Face Detection model. It describes the inference time taken by the network to process each and every frame captured from the video camera. It can be observed that the average inference is estimated to be 0.035 secs but this term is variable depending on the resolution of the frame captured by the camera. Once these feature vectors are processed from a sequence of fully connected layers the resultant feature vector is fed into 2 branch networks, wherein the first network calculates the likelihood of an object of interest being in region using softmax regression method[15], while the other branch network calculates the bounding box coordinates using regression. One key advantage that the ROI pooling layer gives is the ability to easily visualize its function. The ROI pooling layer creates feature vectors of size H*W from the object proposal. The H and W are hyperparameters independent of the (row,column, height,width) coordinates of the ROI. The ROI pooling layer works by converting the ROI of size h*w into small regions h/H * w/W size and max pool them into H*W size. For fine tuning, the Fast R-CNN samples mini batches hierarchically i.e first it sample N images and then sample R/N ROI from these images. When N is kept small the number of RoI's from the same images are increased and thus the computations are shared amongst the RoI from the same image, but this might lead to a slower convergence for the model. While the R/N ROI's are equally sampled from every image, only RoI's with over 50% IOU score are selected for parameter tuning, and the rest are used as a negative examples or background. This process of selecting RoI's with IoU score over 50% for parameter optimization and the rest a negative example is known as Hard Mining.Fast R-CNN [2] uses Multi-Task

loss using both sibling branches. Equation 4 corresponds to this multi-task loss. The regression part has 4 coordinates (tx,ty,tw,th)=[x coordinate, y coordinate, width, height] for each of the class where tx,ty scalar-invariant translation and tw, th are log-space width, height shift relative to an object. The multitask loss equation can be seen in the image where DPL_{cls} is the log loss for a particular class and DPL_{loc} can be seen in the eqn 5 and eqn 6 [u >= 1] term means only the loss is calculated for the non-background classes. In eqn 5, p represents the predict probability value, u represents the ground truth label, t^u represents the offset for each of the 4 x,y,w,h coordinates w.r.t and v is similarly the ground truth coordinate values. In eqn 6, DPL_{loc} is the localization loss where t_i is the offset of x,y,w,h coordinates and v_i is similarly the ground truth coordinate values. Fig.1 shows the complete workflow of Deep learning based Automatic Attendance System.

$$DPL(p,u,t^u,v)=DPL_{cls}(p,u)+\lambda[u\geq 1]DPL_{loc}(t^u,v) \quad (4)$$

$$DPL_{loc}(t^u,v)=\sum_{i\in\{x,y,w,h\}} smooth_{L_i}(t_i^u-v_i) \quad (5)$$

$$smooth_{L_i}(x)=0.5x^2, if\ |x|<1 \quad (6)$$

$$smooth_{L_i}(x)=|x|-0.5, otherwise$$

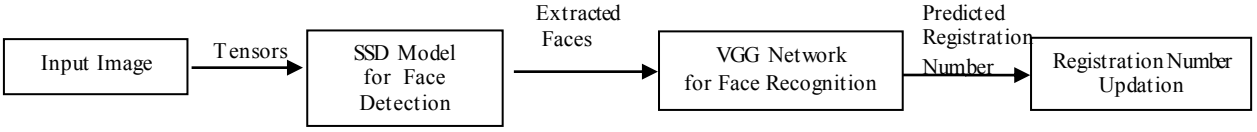


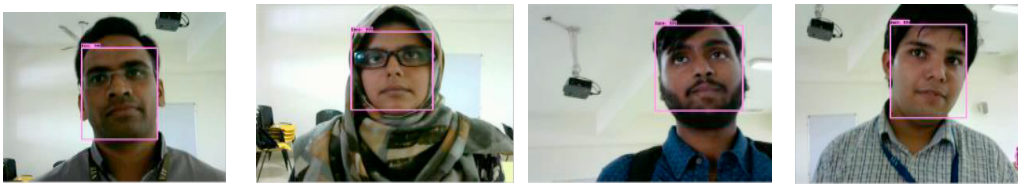
Fig. 1. Complete workflow of Deep learning based Automatic Attendance System

This model takes image as an input which is then converted into a tensor which is a suitable format for applying matrix operation to achieve speed up in computation. Along with image to tensor conversion, images are appended together to form a batch of images for further speeding up the process. After this step, the SSD model is supplied with an input tensor. It yields an output containing the x,y,w,h values which are compared with the ground truth values and the error between these values are back-propagated to update the weights. The training procedure of the network is completed when the validation loss of the network remains constant over many epochs or the validation loss reaches a desired value. The inference version of the SSD model can be viewed where in the model predicts the class probability of all the classes present in the image, it will be the probability that the detected region is a face and the detected region is obtained using 4 coordinates which the network predicts. These coordinates are further used to crop the face from the video frames recorded. The extracted images are then fed into the VGG network. The pipeline is constructed in such a way that the input image is treated with multiple transformations such as rotation, scaling, flips etc. These transformations exponentially increase the number of images available for the network to be trained on increasing the generalizability of the network. Once all of these operations are performed on the image these are converted into the tensor and batched with the other images similar to the SSD model. The VGG network is supplied with these tensors for training the network. The prediction of the network is compared with the ground truth class labels. Cross entropy loss is calculated between the predicted label and ground truth label. Once the network is a trained VGG network, it predicts the face. Once the face's registration number is obtained, it is updated in the database.

4. Experimental result and analysis

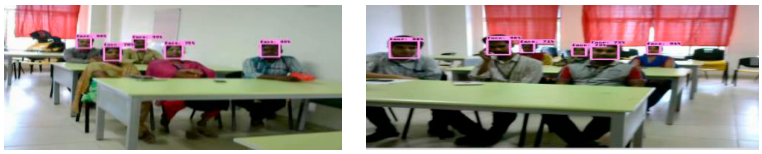
The architecture consists of two neural network models, the training time associated with both models on a computer with 16GB memory, NVIDIA Pascal Architecture Graphical Processing Units (GTX 1050 (640 CUDA cores with 4GB GDDR5 Memory) & GTX 1050 Ti (768 CUDA Cores with 4GB GDR5 Memory)) GPU and 7th Generation Kaby Lake i7 (2 x 7700HQ (8 Threads at 3.8 Ghz) & 1 x 7820HQ (8 Threads at 4.1 Ghz)) processor. The memory associated with the software is the space taken to load the weights and model architecture stored in HDF5 files. Additionally, memory to load a test image is also required. However, this is variable and depends on the camera used in the setup. The first module to be developed was the face detector which was developed on top the popular deep learning framework TensorFlow (TF). The training procedure is established by TF records which

boosts up the ease of access data by enabling batch extraction of data, easy initialization and re-initialization and makes it easier to applying transformation over the data which in case of image might be horizontal flips, vertical flips and rotations etc. Once TF records are created the network is ready to be trained. The training of the network was performed on GTX 1050 Ti for 2 complete days where the hyper-parameters were optimized by using random search. Along with training of the detector the classifier was also developed using popular deep learning framework Keras because the keras can be used to quickly prototype large convolutional neural networks and moreover the VGG network doesn't use Default boxes so the need for low level implementation such as in the TensorFlow is not needed. The VGG network was trained on personally scrapped image dataset. The training dataset in SSD model is the WIDER dataset which contains coarsely annotated face of different people in different situation. The WIDER dataset was split in such a way that training images were 9798, validation set images with the size of 1960 and finally the test set with 1305 images. The main advantage of this dataset is that it provides much faster generalization as compared to any other dataset because it provides images of faces people with conditions such as multiple scales, different poses, occlusion, different expressions, makeup and most importantly illumination. The VGG network on contrary was trained with the help of scraped data from internet as it an abundant source of data. The training data for VGG network split in such a way 349 images per class, 66 images for validation per class and 55 images for testing per class. Final step involved integration of the face detector and face recognizer together to perform joint detection and classification of the input images which contains the student's the faces. Fig.2 and Fig.3 show the single and multiple face detected from that input images in VIT Dataset and the network is able to recognize multiple faces at the same time. The captured faces are then fed into the face recognizer which outputs the names of all the faces extracted from the SSD model and the output of the face recognizer using DPAAS is shown in Fig. 2 and Fig. 3. Fig. 4 shows the performance in terms of loss and accuracy for several epochs. It is seen that loss decrease and accuracy increases for increasing epochs. Table 1 displays the confusion matrix for three different people. Thus, the proposed DPAAS gives 94.66% accuracy.



(a) (b) (c) (d)

Fig. 2. (a – d) Single Face detected from VIT Dataset

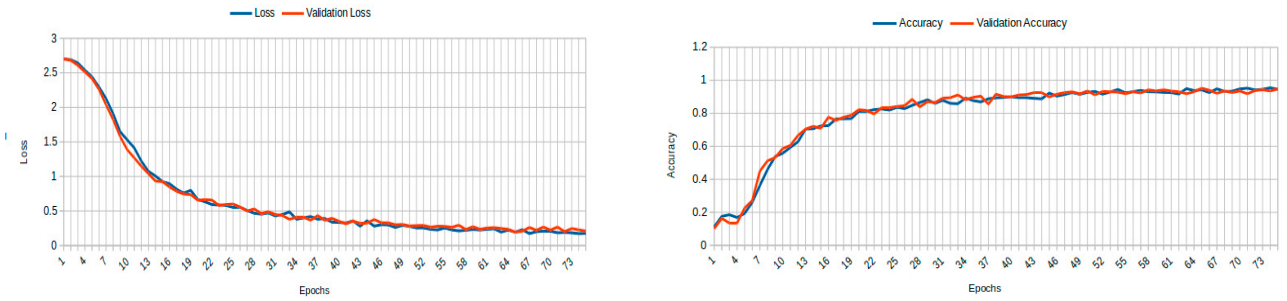


(a) (b)

Fig. 3. (a) Faces Detected from Group dataset 1 of VIT; (b) Faces Detected from Group dataset 2 of VIT

Table. 1. Confusion Matrix

Actual Name\ Predicted Name	A	AB	JS	Total
A	53	0	2	55
AB	0	55	0	55
JS	0	1	54	55
Total	53	56	56	165



(a) (b)

Fig. 4. Performance Analysis. (a) Loss versus Epochs; (b) Accuracy versus Epochs

5. Conclusion

The novel DPAAS framework can be summarized as a deep learning based automated attendance system which uses Single Shot Multibox detector neural network model to extract the confident region which exhibit the features of a human face. These facial features are captured by the ResNet base network. These facial regions are further used as an input to VGG network. The VGG network is again a neural network but here it plays the role of a classifier rather than the detector or a feature extractor. When the recorded images are forward propagated, the VGG network recognize the person with unique registration number. This registration number can be further used to update the attendance of the detected candidates. The computation in the entire pipeline starting from face detection to the recognition process takes place on a Movidius Neural Compute Stick. In future advancements in hardware, such as efficient, powerful and portable GPUs would make this work applicable in a wider set of use cases, particularly where workstation class desktops and servers are unavailable.

References

[1] Girshick, R., Donahue, J., Darrell, T. and Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580-587).

[2] Fast R-CNN," *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, 2015, pp. 1440-1448. doi: 10.1109/ICCV.2015.169.

[3] Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91-99).

[4] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

[5] Viola, P., & Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*, *57*(2), 137-154.

[6] Zhu, Q., Yeh, M. C., Cheng, K. T., & Avidan, S. (2006). Fast human detection using a cascade of histograms of oriented gradients. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)* (Vol. 2, pp. 1491-1498). IEEE.

[7] Ding, W., Wang, R., Mao, F. and Taylor, G., 2014. Theano-based large-scale visual recognition with multiple gpus. arXiv preprint arXiv:1412.2302.

[8] Cortes, C. & Vapnik, V. Mach Learn (1995) 20: 273. https://doi.org/10.1007/BF00994018.

[9] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

[10] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

[11] Redmon, J. and Farhadi, A., 2017. YOLO9000: better, faster, stronger. *arXiv preprint*.

[12] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C., 2016, October. Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Springer, Cham.

[13] Lowe, D.G. International Journal of Computer Vision (2004) 60: 91. https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[14] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[15] Lin, T.Y. and Maji, S., 2016. Visualizing and understanding deep texture representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2791-2799).

[16] Yang, S., Luo, P., Loy, C. C., & Tang, X. (2016). Wider face: A face detection benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5525-5533).

[17] Joshan Athanesious. J, Sibi Chakkaravarthy. S, Vasuhi. S, Vaidehi. V, Trajectory based abnormal event detection in video traffic surveillance using general potential data field with spectral clustering", Multimedia Tools and Applications, vol. 78, no. 14, pp, 19877-19903, July, 2019.