A Deep Learning Approach for Face Detection using YOLO

Dweepna Garg
Computer Engineering Department
Devang Patel Institute of Advance Technology and Research, CHARUSAT
Changa, Anand, India
dweeps1989@gmail.com

Parth Goel
Computer Engineering Department
Devang Patel Institute of Advance Technology and Research, CHARUSAT
Changa, Anand, India
er.parthgoel@gmail.com

Sharnil Pandya
Computer Science & Engineering Department
Navrachana University
Vadodara, India
sharnil.pandya84@gmail.com

Amit Ganatra
Devang Patel Institute of Advance Technology and Research
Charotar University of Science and Technology
Changa, Anand, India
amitganatra.ce@charusat.ac.in

Ketan Kotecha
Symbiosis Institute of Technology
Symbiosis International University
Pune, India
drketankotecha@gmail.com

***Abstract*—Deep learning is nowadays a buzzword and is considered a new era of machine learning which trains the computers in finding the pattern from a massive amount of data. It mainly describes the learning at multiple levels of representation which helps to make sense on the data consisting of text, sound and images. Many organizations are using a type of deep learning known as a convolutional neural network to deal with the objects in a video sequence. Deep Convolution Neural Networks (CNNs) have proved to be impressive in terms of performance for detecting the objects, classification of images and semantic segmentation. Object detection is defined as a combination of classification and localization. Face detection is one of the most challenging problems of pattern recognition. Various face related applications like face verification, facial recognition, clustering of face etc. are a part of face detection. Effective training needs to be carried out for detection and recognition. The accuracy in face detection using the traditional approach did not yield a good result. This paper focuses on improving the accuracy of detecting the face using the model of deep learning. YOLO (You only look once), a popular deep learning library is used to implement the proposed work. The paper compares the accuracy of detecting the face in an efficient manner with respect to the traditional approach. The proposed model uses the convolutional neural network as an approach of deep learning for detecting faces from videos. The FDDB dataset is used for training and testing of our model. A model is fine-tuned on various performance parameters and the best suitable values are taken into consideration. It is also compared the execution of training time and the performance of the model on two different GPUs.**

Keywords—Face Detection, YOLO, Neural Network, object detection, Convolutional Neural Network

I. INTRODUCTION

In early times, research was carried out on the various hand-crafted features extraction methods which were used in training the traditional machine learning algorithms for detection and recognition. It leads to an increase in the computation power and time for extracting features and gives less accurate results. To overcome the computation time, power and accuracy, the same was implemented using the models of neural networks and thereafter deep neural networks.

There are various deep learning [1] models like convolutional neural network, recurrent neural network etc. but among all, deep convolutional neural networks (CNNs) [2] are the best model for finding patterns from images. CNN also has the capability to classify, detect and label the object with high accuracy. Region-based CNN (R-CNN) [3], Fast R-CNN [4], Faster R-CNN [5], and YOLO [6] are popular object detection networks in recent years.

Face detection has a plethora of applications. It plays a crucial role in face recognition algorithms. Face recognition has several applications such as person identification in surveillance and authentication for a security system. It is also help for emotion recognition and based on detected emotion, further analysis can be used for emotion based applications. Hence, it is considered to be a way to deliver rich information like age, emotion, gender and many more about an individual. Other applications of face detection are to automatically focus on human faces in camera, to give tag and to identify different parts of faces. Automated face detection has gained attention in computer vision and pattern recognition. Earlier face detection systems could handle only simple cases but now it has outperformed in various situations using deep learning algorithms. Due to large variation caused by occlusions, illumination and viewpoints, face detection remains a challenging problem in the area of computer vision. So accuracy, training time and processing time in real-time videos for detecting faces are still research issues.

In this paper, section two presents related work of face detection algorithms. Section three describes the working of YOLO framework for detecting objects. Proposed work is explained in section four. Experimental setup and dataset information are discussed in section five. Results are analyzed in section six. Finally, conclusion and future work are described in section seven.

II. RELATED WORK

Face detection is one of the challenging problems in the field of pattern recognition. Early in 1994 Vaillant et al. [7] had applied the algorithm named neural networks for detecting the faces. They had proposed a model which could detect the absence or presence of the face in an image by training a neural network. In this method, the entire image was scanned with the network at all possible locations. In the year 1998, [8] rotation invariant face detection method was used wherein a "router" network estimated the orientation of the face and proper detector network was applied. For detecting the semi-frontal face from a complex image, a neural network was developed by Gracia in the year 2002 [9]. Convolutional neural network for pose

estimation and detection of the face was proposed by Osadchy [10]. Wilson et al. presented harcascading for facial feature detection [11]. But limitation arises for [10, 11] when the face is exposed to various illuminations, poses and expressions.

In recent years, face detection is carried out using deep learning models. One of the most popular models for it is CNN (convolutional neural network) [12]. Faster R-CNN is also achieving remarkable results for object detection. This paper proposes an architecture of a convolutional neural network to detect the face using the YOLO framework.

Our architecture does not rely on the hand-crafted features. Faces are detected based on the CNN which extract features by itself. Training and testing of a model are carried out on two GPU and it detects the faces at a faster rate in real time.

III. Overview of Yolo

YOLO is a state-of-the-art deep learning framework for real-time object detection. It is an improved model then the region based detector and outperformed on standard detection datasets like PASCAL VOC [13] and COCO [14] dataset. Detecting the object on real-time basis is comparatively faster with respect to other detection networks. This model can run on different resolutions thereby giving good speed and accuracy. To improve the performance towards scale invariant, the images can be resized to a random scale. The detector should be capable to learn the features for a wide range of image sizes.

Object detection should be fast, accurate in a manner that a variety of objects can be recognized [15]. With the help of neural network, the YOLO frameworks are becoming increasingly fast and accurate for detection. Still, a constraint is observed for small set of objects. Presently, the datasets of object detection are limited as compared to that of classification and tagging. The object detection datasets consist of thousands of images with tags which are object coordinates in image. The classification datasets consist of millions of images with categories. Assigning a tag of an object to the image for detection is more expensive as compared to assigning a label for classification.

Region-based CNN generates a bounding box in an image and then runs the classifier on these boxes. The bounding boxes are then refined using post-processing like non-maximum suppression to eliminate duplicate detections. A single CNN can predict multiple bounding boxes and class probabilities of objects. YOLO optimizes the performance as it is fast in detection. In YOLO (You Only Look Once), a single neural network is applied on the entire image during training and testing time. It encodes the information about the appearance and the classes.

In our work, the bounding box is predicted based on the features from the image. The bounding boxes across an image are predicted in parallel. Hence, it can be said that the network scans the full image as well as the object in the image. With the help of YOLO, end-to-end training is applied along with real-time speed. This enables to maintain high average precision.

The working of YOLO is as follows: The input image is divided into S x S grid. In case the center of the object falls into a grid cell, then it is the responsibility of the grid cell to detect the object. Each cell of the grid predicts the bounding box and the confidence score for that box. The confidence score depicts the accuracy with which the object is detected in the bounding box. If no object is found in the cell, then the confidence score is zero else it is calculated using the intersection over union (IOU) between the predicted box and the ground truth. There are in all mainly 5 predictions in the bounding box: x, y, w, h and confidence. The center of the box with respect to the bounds of the grid is represented by the (x, y) coordinates. The height and width are predicted relative to the whole image. Each cell also predicts the conditional class probabilities. Multiplication of conditional class probabilities with the individual box confidence prediction gives the confidence score for each box. The calculated confidence score depicts that how accurate the predicted box fits the object.

There are various versions of YOLO. Yolov1 suffers from the localization errors and has a low recall compared to the other region based detection methods. The network classifier of original YOLO is trained at 224 x 224 and for detection, the resolution is increased to 448 x 448. For ImageNet dataset, the network classifier is trained at 448 x 448 resolution by YOLOv2. The downsampling of the image is carried out by the convolutional layer of YOLO by a factor of 32, hence an image which is fed as an input of 416 gets an output feature map of 13 x 13.

IV. Proposed Architecture

Our proposed network takes an input as a colour image of size 448 x 448. The architecture consists of 7 convolutional layers followed by max pooling layer of size 2 x 2. Then three fully connected layers are attached and output layer is followed by last fully connected. The convolutional layers find the simple features to complex features from the images and the fully connected layer predicts the coordinates and probabilities. Finally, the output layer predicts both class probabilities and the coordinates of the bounding box using NMS (Non-Maximum Suppression) technique.

V. Experimental Setup & Dataset Information

The experiment is performed on two machines. The first experiment is performed on core i5 processor, 8GB RAM and 2GB GeForce 820M GPU. The second experiment is performed on core i7 processor, 16 GB RAM, and 4 GB NVIDIA GTX 1050 Ti GPU. The proposed architecture of convolutional neural network is trained and tested for face detection on FDDB (Face Detection Dataset and Benchmark) dataset [16].

FDDB Dataset is used in our work to train the proposed architecture. It consists of 5171 faces in a set of 2845 images. This dataset consists of regions of persons designed for studying the problem of detection. This work deals with 2667 number of images and the total size of the dataset (in our study) is 52.2 MB. Dataset was divided into 70% training dataset and 30% testing dataset.

VI. Result Analysis

The model was trained for 25 epochs with gradient decent optimizer algorithm. It was observed that accuracy

remained nearly constant 92.2% after 20 epochs and the best value of learning rate is considered after trying different values and it is 0.0001 as shown in Fig. 1. Same epochs and learning rate are considered for comparison of experimental analysis on CPU and GPU.

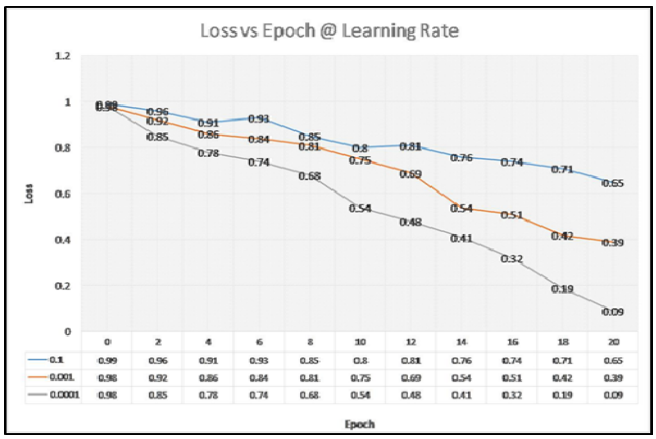

Fig. 1. Loss vs learning rate.

Fig. 2 shows 92.2% accuracy which was achieved on test dataset for 20 epochs. Network was also trained with different batch size. The batch size was kept 1, 8, 16 and 32. It was observed that when the batch size was 32 or 16, the network was not able to get trained on 2 GB 820M Graphics card. It happened due to less size of GPU memory which could not accommodate increased batch size. The same is depicted in Fig. 3.

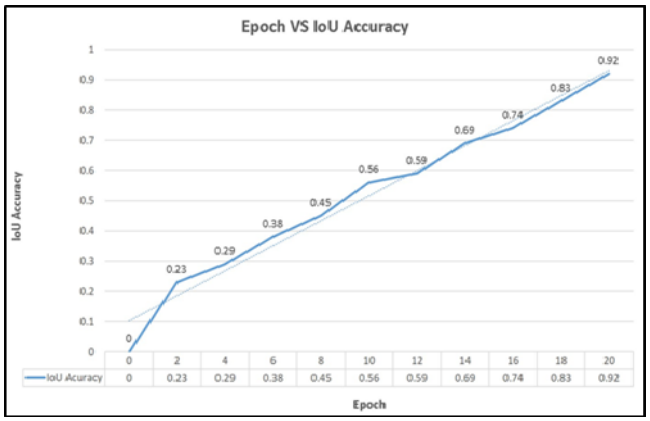

Fig. 2. IoU accuracy vs Epoch.

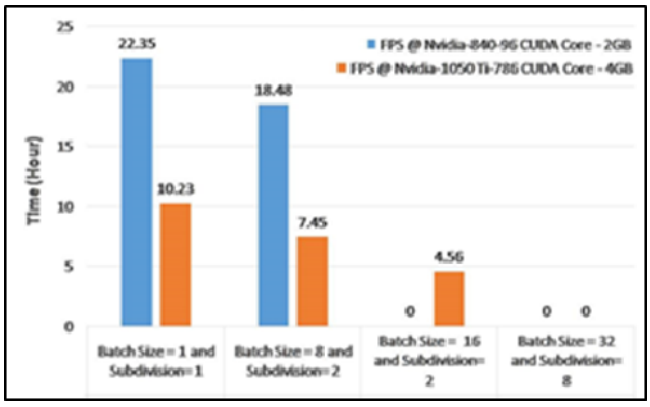

Fig. 3. Batch size vs Training time (hours)

After training a network, weight file and network configuration file were tested on different resolutions of videos. Fig. 4 shows that resolutions were also a parameter which affects the FPS (frames per second) rate. It was observed that FPS was increased as the resolution was decreased. Low-resolution image has less number of pixels, so GPU process it speedily because of less number of calculations of parameters.

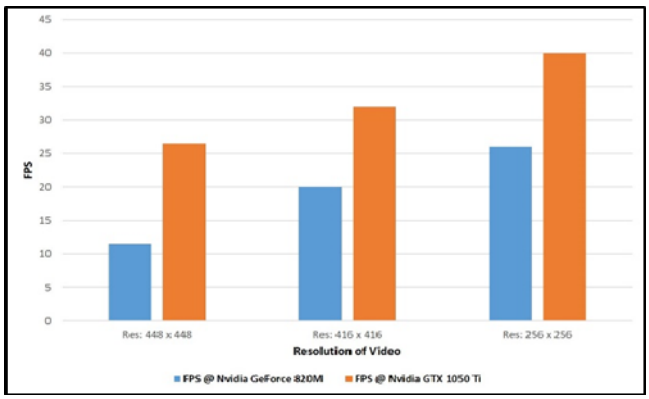

Fig. 4. Resolution of video vs FPS

The accuracy of the proposed model was compared with other face detection algorithms after fine-tuning all parameters and hyperparameters of the proposed model. It was shown that proposed model accuracy was higher than the haar cascade algorithm and R-CNN based face detection model which is depicted in Fig. 5.

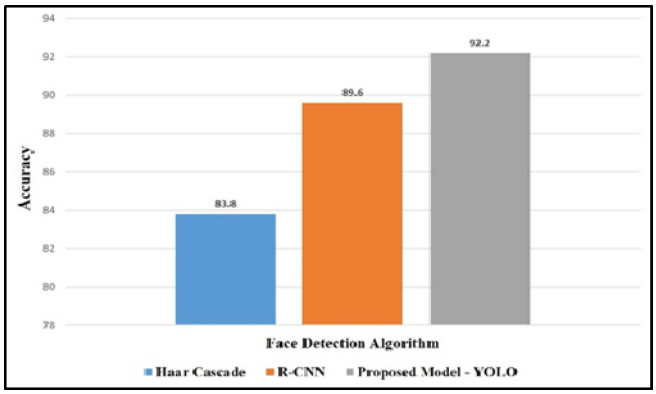

Fig. 5. Comparison of accuracy of proposed model with other face detection algorithm

VII. Conclusion

It can be concluded that processing a huge amount of data using deep learning requires a high configuration NVIDIA graphics card (GPU). If the configuration of the GPU is high, then computation of the task can be achieved at a faster rate. There are various parameters which are responsible for detecting the face from either an image or a video. Based on the analysis carried out by the proposed model, the following points can be concluded. Firstly, the learning rate depends on the network size and size of object too. If the network is medium or large and size of object is compared to less, then the learning rate should be kept small. In our work, the network consists of 18 layers, so the calculated learning rate = 0.0001. Secondly, if number of times the dataset are trained on the network, better results are obtained. It also provokes data overfitting issue so, epoch size should be kept at an optimal number which can produce neither network overfitting nor underfitting. In our work, after 20 epochs it was observed that the IoU accuracy obtained is the best i.e 92.2%. Also, resolution of the image

plays a very important role. Resolution of the image as concluded is inversely proportional to the frames per second. In future work, proposed model can be further optimized for very small face detections, on different viewpoint variations, and partial face detection.

REFERENCES

[1] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1. Curran Associates Inc., pp. 1097–1105, 2012.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.

[4] R. Girshick, "Fast R-CNN," Proc. IEEE International Conference on Computer Vision, ICCV 2015, pp. 1440–1448, 2015.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1. MIT Press, pp. 91–99, 2015.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2015.

[7] R. Vaillant, C. Monrocq, and Y. Lecun, "Original approach for the localisation of objects in images," IEEE Proceedings on Vision, Image, and Signal Processing, vol. 4, 1994.

[8] H.A. Rowley, S. Baluja, T. Kanade, "Neural network-based face detection", IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, no. 1, pp. 23–38, 1998.

[9] C. Garcia and M. Delakis, "A neural architecture for fast and robust face detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 26, no. 11, pp. 1408–1423, 2004.

[10] M. Osadchy, Y. Le Cun, and M. L. Miller, "Synergistic Face Detection and Pose Estimation with Energy-Based Models," Journal of Machine Learning Research, vol. 8, pp. 1197-1215, 2007.

[11] F. J. Phillip Ian, "Facial feature detection using Haar classifiers," J. Comput. Sci. Coll., vol. 21, no. 4, pp. 127–133, 2002.

[12] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A Convolutional Neural Network Cascade for Face Detection.", IEEE International Conference on Computer Vision and Pattern Recognition, CVPR 2015, pp. 5325-5334, 2015.

[13] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes (VOC) Challenge", International Journal of Computer Vision, vol. 88, no. 2, pp. 303-338, 2010.

[14] T.-Y. Lin *et al.*, "Microsoft COCO: Common Objects in Context," European Conference on Computer Vision, ECCV 2014, Lecture Notes in Computer Science, vol 8693. Springer, Cham, pp. 740-755.

[15] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. Curran Associates Inc., pp. 2553–2561, 2013.

[16] V. Jain and E. Learned-Miller, "FDDB: A Benchmark for Face Detection in Unconstrained Settings.", Technical Report UM-CS-2010-009, Dept. of Computer Science, University of Massachusetts, Amherst. 2010.