SAFETY HELMET DETECTION IN INDUSTRIAL ENVIRONMENT USING DEEP LEARNING

Ankit Kamboj and Nilesh Powar

Advanced Analytics Team, Cummins Technologies India Pvt. Ltd, Pune, India

ABSTRACT

Safety is of predominant value for employees who are working in an industrial and construction environment. Real time Object detection is an important technique to detect violations of safety compliance in an industrial setup. The negligence in wearing safety helmets could be hazardous to workers, hence the requirement of the automatic surveillance system to detect persons not wearing helmets is of utmost importance and this would reduce the labor-intensive work to monitor the violations. In this paper, we deployed an advanced Convolutional Neural Network (CNN) algorithm called Single Shot Multibox Detector (SSD) to monitor violations of safety helmets. Various image processing techniques are applied to all the video data collected from the industrial plant. The practical and novel safety detection framework is proposed in which the CNN first detects persons from the video data and in the second step it detects whether the person is wearing the safety helmet. Using the proposed model, the deep learning inference benchmarking is done with Dell Advanced Tower workstation. The comparative study of the proposed approach is analysed in terms of detection accuracy (average precision) which illustrates the effectiveness of the proposed framework.

KEYWORDS

Safety Helmet Detection, Deep Learning, SSD, CNN, Image Processing

1. INTRODUCTION

The application of video surveillance is vast and multi-dimensional, from online facial expression to traffic signal rule break and even to health sectors. The monitoring of violations of wearing safety helmet in industrial environment involves a lot of manual effort hence the need of having an automatic surveillance system is of utmost importance. Deep learning and its applications in computer vision made a breakthrough due to its computational process, as well as accuracy of the detection of a target object but implementing the model for detection in real time is sometimes challenging if we are testing the model on low power device like raspberry pie. In this scenario, the state-of-the-art deep learning one stage object detection methods like SSD [1] (Single Shot Multi Box Detector) and Yolo [2] (You Only Look Once) are useful. Even though the model will run faster but there would certainly be a trade-off between speed and accuracy and SDD models doesn't provide good performance on small objects but for large objects, they provide competitive performance in comparison to other deep learning models. [3] SSD are often combined with lightweight feature extractors like MobileNet and they have different usage of depth wise separable convolution in comparison to traditional CNNs. [4]

In this paper we have proposed a novel and practical approach of detecting safety helmets by optimizing the performance of SSD MobileNet model for smaller size objects. The proposed approach utilizes application of two CNN models one after the other, first the SSD model is used

David C. Wyld et al. (Eds): ITCSE, NLCA, ICAIT, CAIML, ICDIPV, CRYPIS, WiMo - 2020
pp. 197-208, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.100518

for detecting persons from a video data and then the SSD model identifies whether the person is wearing the safety helmet.

The rest of the paper is organized in the following manner: Section II describes the literature review and related work. Data pre-processing and methodology of novel safety detection algorithm are explained in Section III. Experimental evaluation and results are presented in Section IV. Finally, the paper ends with conclusions and future work in Section V.

2. LITERATURE REVIEW AND RELATED WORK:

Object detection involves both object classification and localization, which requires identification of bounding box around the object that needs to be detected. It is problem of not determining whether an object is in an image but also its location. To predict the coordinates of bounding box we need x and y coordinates for the centre, height and width of the rectangle.

There are many research papers regarding image classification which are based on only feature space. To extract features for object recognition, HOG (Histogram of gradients), SIFT (Scale-invariant feature transform) and Haar-like features are used. [5]. It is very difficult to design a reliable feature extractor by human considering different illumination, backgrounds and appearances of an object. The initial work on implementing object detection algorithms was seen in 2001 using Haar cascade classifiers used by Viola-Jones in their face detection algorithm by minimizing computation time. Haar Cascades takes series of cascaded classifiers using Haar features and then uses a sequence of steps to identify a face and even though they are fast with decent accuracy but are difficult to develop, train and optimize. [6] Yet there are drawbacks for this method as detection of titled or turned faces is not that effective and is also sensitive to lighting conditions. [7]

For any deep learning computer vision task there are sequence of steps that needs to be followed. Firstly, we need to collect visual input composed of images or videos from an imaging device like camera. Then each image needs to be passed through a pre-processing step like noise reduction, colour correction, scaling to enhance the quality and detail of the image. Then the area of interest needs to be selected in an image by annotating each frame so that the model could be trained with relevant features. Some of the images that are kept for testing the model are fed to trained model that could recognize the object of interest and it does so with a certain probability. The input for a CNN is an image and output are the distribution of class scores from which we can get the predicted class for that image. The CNN is made up of series of layers that learn to extract relevant features out of any image such that each layer finds progressively more and more complex features. The backbone of CNN is convolutional layer, where a sequence of many image filters is applied which are also called as convolutional kernels. The filters may have extracted the features like edges of objects or colours that distinguish the different classes of image. As the CNN trains, it updates the weights that define the image filters in this convolutional layer using backpropagation. The result is classifier with convolutional layers that have learned to filter images to extract distinguishing features. [5]

Neural Networks take extremely long time to train, even if you use GPU. Deep learning researchers these days are committed to openness. We can take an existing pretrained neural network and its weights which is called transfer learning. [8] Then by doing some fine tuning we make the network for completely new problems. e.g. we can take SSD MobileNet trained on COCO data and use it on totally new data. [9] We can further do fine tuning of pretrained weights using backpropagation which would eventually result in better accuracy. One major advantage of transfer learning is that when you don’t have enough data for model training.

Single shot multi box detector: SSD is the real milestone in computer vision as before this object detection algorithms were slow and require multiple stages, but with SSD we can get real time performance. As depicted in Figure 1, a feature extractor VGG16 is used. The name VGG stands for visual geology group, which is the research group that invented it and is one of the main benchmarks in image classification algorithms. The SSD object detection algorithm has two main steps, one is to extract the feature maps and other is the convolution filters application to detect objects. Instead of thinking of entire CNN as the state-of-the-art feature extractor, we can think of each subpart of CNN as feature extractor. We therefore take output from multiple parts of CNN and let them do their own object detection. The aspect scores and default object positions are uniquely defined such that while prediction, the SSD network produces scores for the existence of each object class in each default box and generates a change to box for a better match of their object shape. Instead of creating a single grid, SSD create several grids with different scales and feature maps of different resolutions are used to predict object of different scales. [1]

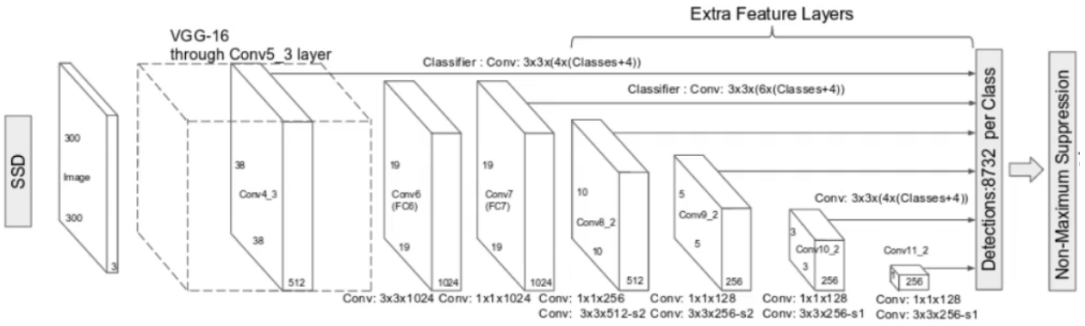

Figure 1: Single Shot Multibox Detector: SSD architecture. Reprinted from [1]

3. SAFETY HELMET DETECTION SYSTEM

3.1. Data Preprocessing

3.1.1. Data Preparation

The videos are collected from areas in industrial environment where safety compliance of wearing helmet is mandatory. We have considered footages from cameras installed at different locations to build a robust set of images. Our main objective is to recognize persons without helmet therefore video footages having persons with and without helmet are considered for analysis. The cameras are installed at the entrance of restricted zones and the experiment is conducted such that a person must face towards the camera while entering the hazardous zones. The cameras are placed at head level elevation from the ground (roughly 6 feets) to have videos with proper alignment of person's head for better detection of helmet. The videos are taken from cameras with resolution (1920*1080) with frame rate of 25 frames per second.

3.1.2. Data Annotation

In our data, proper annotation plays a major role. Annotations are metadata which describes the position of an object in the image. For an image classification algorithm, annotation is not at all necessary as the output only says about the class of the image (i.e. if the object is in that image or not) but in object detection algorithms it is necessary to train an algorithm with the exact position of the object in images.

For annotation we have used a Python library called labelimg which helps to annotate an image manually (makes a bounding box outside the object in the frame) with multiple classes and

multiple objects for a frame. Every single object when annotated, it makes a .xml file which contains its four co-ordinates in the frame along with its given class in it (in PASCAL VOC format). For multiple object in a frame with different class, it just follows same rule and add the co-ordinates and classes of the other object in a new line of the .xml file. It is important to annotate properly otherwise it would result in underfitting or overfitting the model for the data and would detect wrong objects or even would not detect objects at all.

3.2. Methodology

In the novel safety helmet detection system as shown in Figure 2 and Figure 3, first CNN detects persons from the video data and then the second CNN detects whether the person is wearing the safety helmet. The detection system consists of four steps: a) SSD algorithm for person detection b) Cropping the precited images of persons c) Manually annotating the cropped images d) Implementing SSD algorithm for helmet detection.

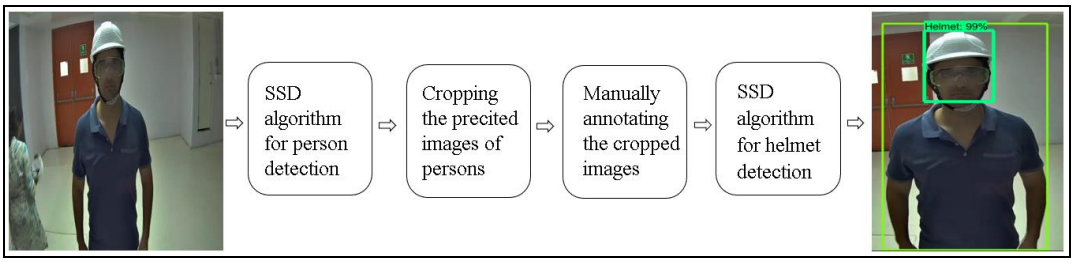

Figure 2. Safety helmet detection system

Figure 3. Safety helmet detection system

3.2.1. SSD Algorithm for Person Detection

In object detection problem it is beneficial to identify object of interest within in image to lower down false positives in predictions. Also, smaller the region to search for an object, lesser will be the processing time for a detection algorithm. [10]

We have used here SSD MobileNet object detection algorithm that is pretrained on coco dataset and implemented on the TensorFlow deep learning software platform. Even though SSD is state-of-art deep learning object detection model, there are many advancements in feature extractor for SDD (like VGG, ResNet or MobileNet). The model SSD-mobilenet-v1-coco with the weights pretrained on COCO data is used at the first step for detecting persons. There are 80 classes in COCO dataset which is widely used for benchmarking performance of deep learning algorithms. [9] The model is further modified to detect only the person class.

3.2.2. Cropping the Precited Images of Persons

While running the SSD person detection algorithm we have simultaneously cropped the images of predicted class around the bounding box in an image. SSD performs well even with a simple extractor like MobileNet but it is challenging task when it comes for detection of small objects. [3] Also, the SSD algorithm used here needs to have input images of size of 300*300(height: 300 width: 300), hence to improve the detection accuracy the cropped images with dimensions lesser than 160*160 are discarded for further processing as they would be of lower quality after resizing to fixed shape resizer of 300*300.

3.2.3. Manually Annotating the Cropped Images

Annotating bounding box on any visible object makes it recognizable for machines and data of annotated images is used in training deep learning algorithm to learn patterns of the object of interest. Further manual annotations are done for all the cropped images using Python library called labelimg by drawing bounding boxes around the object of interest.

Here we have labelled two classes: “Helmet”: for persons wearing the safety helmet and “No_Helmet” otherwise. If multiple persons are present in a single image, both the classes are annotated in that image and the bounding boxes are drawn for only for the head area as shown in figure. Fig.4. The rectangular coordinates of annotated objects are stored in a .xml file along with its class in PASCAL VOC format and these later act as ground truth values for while training CNN in the next step of helmet detection.

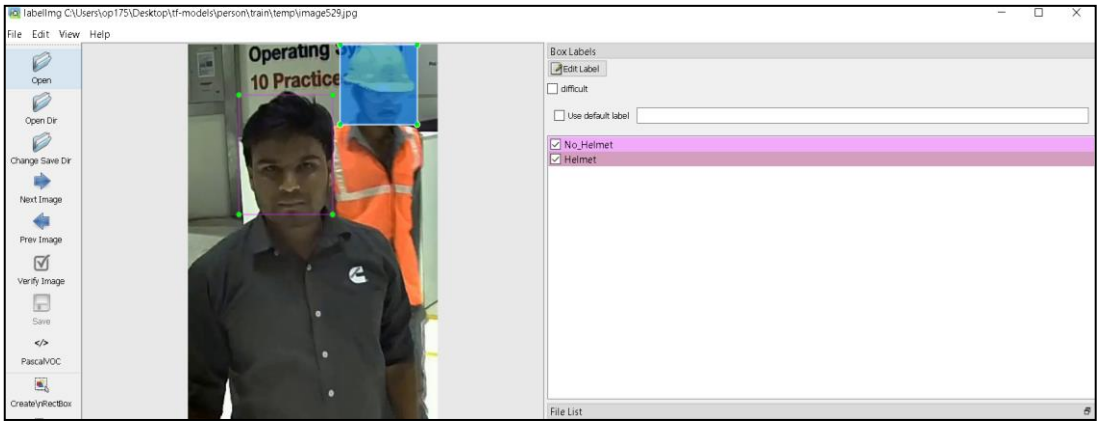

Figure 4. Manual Annotations in Labelimg

3.2.4. Implementing SSD Algorithm for Helmet Detection

The SSD MobileNet algorithm (SSD-mobilenet-v1-coco) with the weights pretrained on COCO data is then used as second CNN for detections of two classes: Helmet and No_Helmet. In computer vision, model that is trained on large benchmark dataset to provide a solution for a similar problem and hence providing more accurate models with lesser time is called pretrained model and the process is called transfer learning. The transfer learning helps to learn the patterns of new dataset by further fine tuning of pretrained weights using backpropagation.

4. EXPERIMENTAL EVALUATION AND RESULTS

4.1. Literature Review of Evaluation Metrics

Mean Average Precision(mAP) is the most widely used evaluation metrics for object detection problem. [11] The Area under curve (AUC) curve fails when to check the performance of an object detection algorithm as the curve goes up and down in zigzag manner. Even when comparing two graphs become very hard. To get rid of this problem an 11 points interpolation is done to summarize precision vs recall curve by taking average of the precision values at different recall values in $[0,0.1,0.2, \dots ,0.9,1]$, where $\rho_interp(r)$is the maximum precision value at that point.

$$\text{Average Precision} = \frac{1}{11} \sum_{r \in \{0,0.1,\dots,1\}} \rho_{\text{interp}(r)} \qquad (1)$$

$$\text{Precision} = \frac{\text{total number of correct detection}}{\text{total number of detection}} \qquad (2)$$

$$\text{Recall} = \frac{\text{total number of correct detection}}{\text{total no. of target detect ground truth}} \qquad (3)$$

If C is number of classes (in our case C=2 (Helmet and No_Helmet)), mAP is calculated by

$$\text{mAP} = \frac{\sum_{c=1}^{C} \text{average Precision}(c)}{C} \qquad (4)$$

4.2. Model Hyperparameters

There is no predefined rule to select some of the parameters of any deep learning model. That's why tuning certain hyper parameters play a vital role in deploying such a model. How a deep network will train itself solely depends on these hyperparameters. Below we have discussed about certain optimization hyper-parameters some model hyper-parameters as they are flexible to the deployer of the algorithms and setup before training. These parameters affect the runtime of the model as these are the core features indicates convergence criteria or stopping criteria of the algorithm.

4.2.1. Number of Steps

The steps parameter indicates the number of training steps to run over data. Here we have kept 69,744 i.e. approx.(70k) steps for model training.

4.2.2. Learning Rate and Decaying Rate

When learning rate is very low the model takes much more time to converge again if it's too high the point of convergence may be missed. Taking into consideration these ideas, we have taken the default learning rate values for SSD_mobilenet_vi_coco and started training with an initial exponential decaying learning rate of .004, the decay steps are 800720 and decay factor is .95.

4.2.3. IoU (Intersection over Union)

IOU is a function that is used for both object localization in non-max suppression and evaluating the object detection algorithm. It is computed by the given formula, where intersection and union of area refers to the intersection and union of two bounding boxes. IOU value greater than 0.5 is a benchmark result. More IOU implies better accuracy. Here we have used IoU ratio as 0.5, i.e. when the predicted bounding box and the ground truth box overlap more than 50%, the predicted box has been considered as a correct result.

$$\text{IoU} = \frac{\text{size of the intersection area}}{\text{size of union area}} \qquad (5)$$

4.3. Experimental Results

We have used advanced workstation for running this model, which has Intel(R) Xeon(R) Gold 6148 CPU @2.40Ghz, 384 GB RAM and 16 GB NVIDIA Quadro P5000 GPU on Windows10. The versions of few important libraries are: Python:3.6.8, Tensorflow:1.13.1, Numpy:1.16.2, open-cv:4.1.0, PIL: 6.0.0, Cuda 10: V10.0.130, CUDNN: 7.4.1. For evaluating the performance of the model, we have the dataset of cropped images from person detection model. The dataset consists of 5773 images, among which (4043 images) are in training set, (865 images) are in testing set and (865 images) are in validation set. The deep learning inference benchmarking is done on the above-mentioned workstation and it took around 17 hours for 69,744 steps of training the model.

The graphs of experimental results are taken from TensorBoard, which is a visualization software to analyse, debug and understand the flow of tensors and different performance metrics. In TensorFlow we have two basic components: operations and tensors. [12] When a model is created, it consists of set of operations and data or tensors are feed into the model and then the tensors will flow between operations until you get the output tensor.

TensorBoard are mainly used to write summaries to visualize learning and we have used the detection evaluation metrics used by COCO. [13] As depicted in Figure 5, it is a plot of mean Average Precision(mAP) at 50% IOU. On the x axis we have number of steps used for training and on the y axis we have the value of mAP. In the starting of training, the mAP has a sharp rising slope till 10k iterations, after that there is slight gradual increase till 35k(approx.) iterations and post that the accuracy is stagnant with very less significant change.

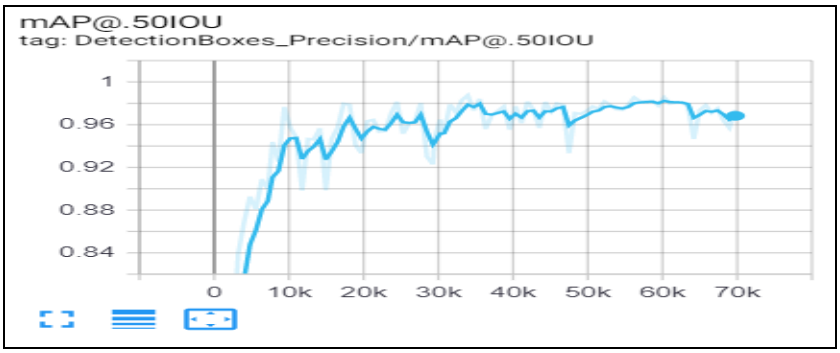

Figure 5. mean Average Precision(mAP) at 50% IOU

In the next Figure 6, it demonstrates the Average Recall (AR) given for 1 detection per image (AR@1), 10 detections per image (AR@10) and 100 detections per image (AR@100). [13]. On the x axis we have number of steps used for training and on the y axis we have the value of AR, for example, AR@10 would imply that for a single image we take 10 highest confidence predictions and then calculate the metrics on those 10 detections. Since in our dataset after the output of person detection algorithm, most of the time we have only one ground truth object per image and hence we have similar curves for all AR@1, AR@10 and AR@100.

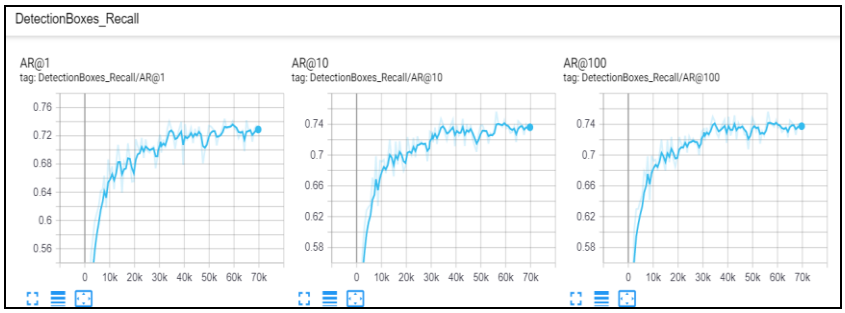

Figure 6. Average Recall with 1,10 & 100 detections per image

The training loss and validation loss are shown in Figure 7 and Figure 8 respectively. Here on the x axis we have number of steps used for training and on the y axis we have the loss value. As depicted in Training loss, we observe that the loss is decreasing at faster till 10k training iterations and beyond that there is gradual decrease in training loss. Visualizing the validation loss, we notice that loss decreased till 10k training steps, then there are some fluctuations in loss and it gradually decreases till 35k training steps which is further shows increasing trend till 70k.

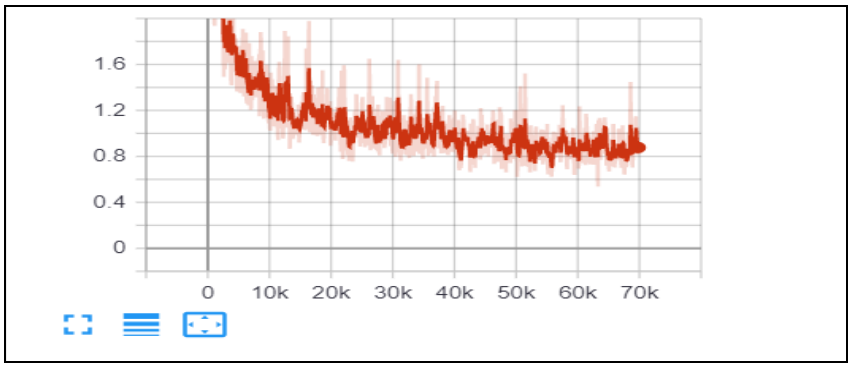

Figure 7. Training Loss wrt no. of training steps

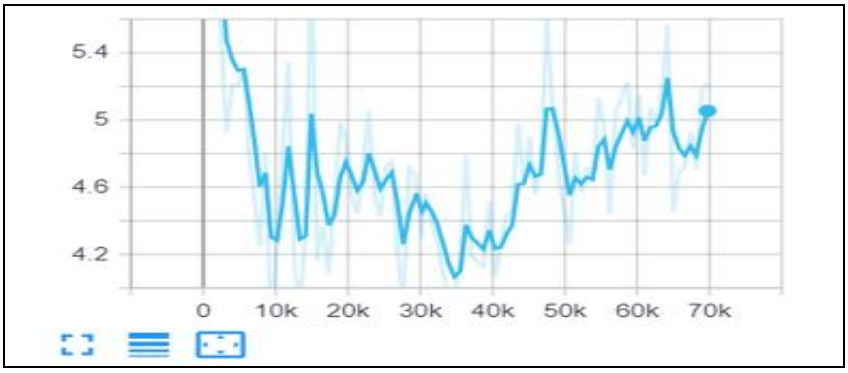

Figure 8. Validation Loss wrt no. of training steps

Based on the Figures 5, 6 & 8, it is observed that the number of training steps should be near to 35k as the model might be overfitting the data if we are moving further with the number of training steps. We might need to fine tune the model hyper parameters or add more samples for training and validation using data augmentation techniques. [14]

To evaluate how well the model has trained, there is a specific functionality of images tab in TensorBoard Figure 9. It shows the bounding box detections on images in validation set and for each image detections are shown in left side and ground truth are shown in right side to compare the results and so the title of image "Detections_Left_Groundtruth_Right". The slider located on the top can be used to go backward and forward in number of training steps used to train the model and hence to analyse the training progression about how the predicted detection bounding box is changing w.r.t change in number of steps in training.

Figure 9. Bounding box detections in left and ground truth in right for images in validation set

5. CONCLUSIONS AND FUTURE WORK

In this paper we have demonstrated the safety helmet detection system to identify whether workers in industrial environment are wearing helmet or not. For better detection accuracy and to reduce false positives in predictions the persons are detected first using pretrained SSD MobileNet model on coco dataset. The predicted images are further cropped and annotated to pass through another SSD MobileNet algorithm for helmet detection.

Comparison of the proposed two stage SSD detection system is done with one step SDD for detection of person with and without Helmet on the same training data and model hyperparameters and further the performance is evaluated on same test data. The mAP@.5IOU for the proposed detection system converges to .96 with an increase in the number of training steps while for the one step SSD detection algorithm it only converges to .7 which proves that comparatively the proposed detection system provides an approximate 37% increase in model accuracy. The experimental results have illustrated that our approach is efficient and effective for helmet detection.

The future work for this paper is to tune different hyperparameters by debugging TensorBoard graphs and to optimize the model. To make a robust algorithm and detector, augmentation plays a key role. Most of the computer vision and deep learning algorithms need more data to train. If the set of training images consist of different angles and different sizes of the object, it automatically enhances the performance. [14] Using the proposed model, the deep learning inference benchmarking needs to be done with Raspberry Pie and Nvidia Jetson for its practical implementation. Further for accounting violations of wearing safety helmets, the algorithm will provide real time monitoring system where the images with No_Helmet predicted class are stored for further inspection, which would henceforth promote awareness of safety among workers.

In this experimentation the cameras are installed at the entrance, which has a narrow passage and each person is instructed to face towards the camera while entering the restricted zone area. One of the limitations of this detection system it that it cannot detect occlusions with multiple persons coming in front of camera at the same time. Since person detection is the first step in the proposed two-stage detection system and occlusions with multiple persons will not provide us precise cropped images of workers.

REFERENCES

[1] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed,C. Fu, and A.C. Berg. SSD: single shot multibox detector. In Computer Vision ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I, pages 21–37,2016. [Online]. Available: https://arxiv.org/abs/1512.02325

[2] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pages 779–788, 2016. [Online]. Available: https://arxiv.org/abs/1506.02640

[3] J. Huang, A. Fathi, V. Rathod, I. Fischer and C. Sun. Speed/accuracy trade-offs for modern convolutional object detectors. In Computer Vision and Pattern Recognition (cs.CV), April 2017, vol arXiv:1611.10012. [Online]. Available: https://arxiv.org/abs/1611.10012

[4] A. Howard, W. Wang, M. Zhu, T. Weyand, B. Chen and D. Kalenichenko. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. Computer Vision and Pattern Recognition (cs.CV), April 2017, vol arXiv:1704.04861. [Online]. Available: https://arxiv.org/abs/1704.04861

[5] Z.Q. Zhao, P. Zheng, S.T. Xu and X. Wu. Object Detection with Deep Learning: A Review. IEEE Transactions on neural networks and learning systems, April 2019, vol arXiv:1807.05511. [Online]. Available: https://arxiv.org/abs/1807.05511

[6] P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. Computer vision and pattern recognition 2001. [Online]. Available: https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf

[7] K. Aashish and A. Vijayalakshmi. Comparison of Viola-Jones and Kanade-Lucas-Tomasi Face Detection Algorithms. Oriental Journal of Computer Science and Technology, March 2017, ISSN : 0974-6471 Online ISSN : 2320-8481. [Online]. Available: http://www.computerscijournal.org/vol10no1/comparison-of-viola-jones-and-kanade-lucas-tomasi-face-detection-algorithms

[8] K. Židek, P. Lazorík, J. Pitel and A. Hošovský. An Automated Training of Deep Learning Networks by 3D Virtual Models for Object Recognition. Symmetry, April 2019, 11(4), 496. [Online]. Available: https://doi.org/10.3390/sym11040496

[9] T.Y Lin, J. Hays, M. Maire, S. Belongie, L. Bourdev and R. Girshick. Microsoft COCO: Common Objects in Context. Computer Vision and Pattern Recognition, February 2015, vol arXiv:1405.0312 . [Online]. Available: https://arxiv.org/abs/1405.0312

[10] R. Silva, K. Aires and R. Veras. Helmet Detection on Motorcyclists Using Image Descriptors and Classifiers. 27th SIBGRAPI Conference on Graphics, Patterns and Images, August 2014. [Online]. Available: https://ieeexplore.ieee.org/document/6915301

[11] Z. Zou, Z. Shi, Y. Guo and J. Ye. Object Detection in 20 Years: A Survey. IEEE TPAMI, May 2019, arXiv:1905.05055. [Online]. Available: https://arxiv.org/abs/1905.05055

[12] A. Mobiny. How to Use TensorBoard? ITNEXT, June 2018. [Online]. Available: https://itnext.io/how-to-use-tensorboard-5d82f8654496

[13] COCO Common object in context, Evaluate tab. [Online]. Available: http://cocodataset.org/#detection-eval

[14] C. Shorten and T. M. Khoshgoftaar. A survey on Image Data Augmentation for Deep Learning, Journal of Big data, July 2019, 6, Article number: 60 (2019). [Online]. Available: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0

AUTHORS

Ankit Kamboj received his bachelor's in engineering from LNMIIT, India in 2014 and his Masters in Operations Research from Indian Statistical Institute, Kolkata, India in 2017. He has over 3+ years' experience in the field of Applied Statistics, Machine Learning and Computer Vision. Mr Kamboj is working as a Senior Data Scientist in Cummins. Recent efforts involve Multivariate Time Series Forecasting, proactive engine failure prediction using Machine Learning and attrition risk prediction.

Nilesh Powar received his bachelor's in engineering from University of Bombay, India in 1999, his M.S. in Computer Engineering from Wright State University in 2002, and a PhD in Electrical and Computer Engineering in 2013 from the University of Dayton. He has over 20+ years' experience in field of image processing, machine learning, statistical pattern recognition and system integration. Dr Powar worked in the US as Distinguished Research Scientist for University of Dayton Research Institute, Dayton, OH, USA. Currently he leads the Advanced Analytics team for Cummins - India as a Director. Recent efforts involve data analytics for die casting, predictive analysis for supply chain management and video summarization using deep learning.