

VGG-SSD Model for Weapon Detection using Image Processing

Ravi Kiran Varma P

Dept. of CSE

Sagi Rama Krishnam Raju Engineering
College

Bhimavaram, India

ravikiranvarmap@gmail.com

Kishore Raju K

Dept. of IT

Sagi Rama Krishnam Raju Engineering
College

Bhimavaram, India

kkrsrkrit@gmail.com

Krishna Chaitanya R

Dept. of ECE

Sagi Rama Krishnam Raju Engineering
College

Bhimavaram, India

rkchaitanya@srkrec.ac.in

Sirisha G N V G

Dept. of CSE

Sagi Rama Krishnam Raju Engineering
College

Bhimavaram, India

sirishagadiraju@srkrec.ac.in

Dendukuri Narendra Varma

Dept. of CSE

Sagi Rama Krishnam Raju Engineering
College

Bhimavaram, India

narendravarma18042004@gmail.com

Abstract— Video surveillance demands application of technology like deep learning, image processing for automated detection of deadly weapons in the hands of anti-social elements. This work depicts the application of very deep convolutional neural network (CNN), Visual Geometry Group (VGG)-16, and popular object detection technique, Single Shot MultiBox Detector (SSD) for detecting a weapon in automated surveillance application. The model is trained and tested on benchmark data on normal, gun, and knife objects. Performance parameters are, Mean Average Precision (MAP), Recall, Intersection over Union (IoU), and classification loss. A MAP of 87%, a recall of 86.6%, and calculation loss of 0.07 and total loss of 0.35 are achieved and the model proves to be efficient.

Keywords—Deep learning, VGG, SSD, Weapon detection

I. INTRODUCTION

Analysis of CCTV / recorded video tapes is playing crucial role in identification of a crime, theft, or illegal activity. Novel image processing methods can be leveraged to auto detect presence of any weapon like a knife. The research aims in detection of weapons in hand with a human under the surveillance of cameras. A massive, deep Convolution Neural Network (CNN) was trained by Krizhevsky et al. [1] to categorize the 1.2 million high-resolution photographs in the ImageNet LSVRC-2010 contest into the 1000 distinct classes. With 60 million parameters and 650,000 neurons, the neural network is composed of three fully connected layers with a final 1000-way softmax, five convolutional layers, some of which are followed by max-pooling layers. They employed non-saturating neurons and a highly effective GPU convolution operation implementation to speed up training. They were able to attain an error rate of 17% to 35% with this network.

In comparison to the previous best result on VOC 2012, Donahue et al. [2] introduced a straightforward and scalable detection technique that improves mean average precision (mAP) by more than 30%, attaining a mAP of 53.3%. They called their approach R-CNN: Regions with CNN features since they integrated region suggestions with CNNs. In an effort to enhance the state-of-the-art deep convolutional neural network-based image classification pipeline, Howard [3] looked into a number of methods.

A Fast Region-based Convolutional Network technique (Fast R-CNN) for object detection was proposed by Girshick [4]. In contrast to earlier research, Fast R-CNN makes use of several advancements to boost detection accuracy and expedite testing and training. In order to provide almost cost-free region proposals, Ren [5] created a Region Proposal Network (RPN) that shares full-image convolutional characteristics with the detection network. The Single Shot Detector (SSD) technique, which was proposed by Anguelov et al. [6], discretizes the bounding box output space into a series of default boxes with varying aspect ratios and scales per feature map position. Even with a reduced input image size, SSD provides substantially better accuracy than other single stage approaches.

A class of effective models known as MobileNets was introduced by Howard et al. [7] for mobile and embedded vision applications. MobileNets are built on a simplified design that builds lightweight deep neural networks using depth-wise separable convolutions.

Albawi et al. [8] explained the usage of CNNs with the combination of pooling layers in an effective manner. In this paper they explained and defined all the elements and important issues related to CNN, and how these elements work. Chahal et al. [9] made a comparison of all present existing object detection algorithms. Live image weapon detection is proposed in [10], where Inception, Resnet, YOLO algorithms are used for real-time identification of dangerous weapons like guns in the surveillance footages. In another work by Yadav et al. [11] proved that deep NNs does produce better performance than typical ML methods. RCNN and YOLO4 are taken into consideration for various types of weapon recognition in [12]. YOLO 5,6,7 are used along with RCNN for weapon virtual surveillance in [13].

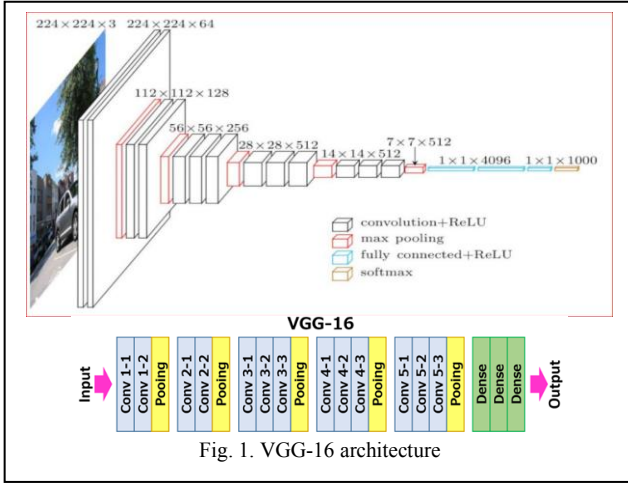
Few more recent works in this area are [14] [15]. This work aims to do a thorough survey of deep learning-based current object detection techniques. The study concentrated on two categories of object detection algorithms: the Faster R-CNN class of two step detectors and the SSD class of single step detectors. An approach is proposed, where real time CCTV video can be given as input to the SSD Mobile net algorithm. This algorithm is built on basis of VGG-16 architecture and CNN. Here, bounding boxes are drawn

against the detected object along with its corresponding accuracy score. The detected objects are weapons. For example, if input video frame contains a knife, then a bounding box is drawn around the knife object in the frame and class label is written with its accuracy (ranging from 0% to 100%). An alert message is sent to the concerned user if a threat object is detected via WhatsApp using selenium browser automation tool.

II. METHODOLOGY

A. VGG-16

One of the best vision model architectures available now is VGG-16 [16]. VGG-16 comes pre-trained on a massive dataset (ImageNet) containing millions of labelled images. Due to its architecture and pre-training, VGG16 achieves high accuracy on image classification tasks. Therefore, the vgg16 is chosen as the model. The most distinctive feature of VGG16 is that, rather than having a lot of hyper-parameters, they concentrated on having 3x3 filter convolution layers with a stride of 1, constantly using the same padding and maxpool layer of a 2x2 filter with a stride of 2. A greater number of specific features will be retrieved from the input image when additional convolution layers are applied. Throughout the whole architecture, the convolution and max pool layer arrangements are maintained. Ultimately, it consists of two FC (completely connected layers) with a softmax as the output. The 16 in VGG16 stands for the 16 weighted layers it contains.



B. SSD Mobilenet Architecture

Transfer learning with VGG16 offers a powerful approach to leverage pre-existing knowledge from large datasets and apply it effectively to new tasks, leading to improved performance, reduced training efforts, and faster model deployment. SSD MobileNet can be effectively used as a transfer learning model by leveraging pre-trained features from the MobileNet backbone and adapting them to object detection tasks through fine-tuning and domain adaptation techniques. The Mobilnet Single Shot Detector algorithm takes the Con. 4 to 9 layers of VGG 16, does additional convolution, and combined with classifier to generate the class label. SSD reduces the training parameters from a massive 138 million of VGG-16. Say for example, at the end of Conv4_3, Feature map is of size 38x38x512. A 3x3 conv is applied to get further

refined feature map. Each of the four bounding boxes will have four outputs (classes + 4) in total. As a result, there are $38 \times 38 \times 4 \times (c+4)$ parameters at Conv4_3.

Algorithm

Step -1:

SSD takes images input along with actual bounding boxes of the weapon object i.e, coordinate (x,y) of starting point, width & height (w,h) of the bounding box.

Step -2:

It applies some of layers VGG-16 architecture on the input image to get Feature Map.

Step -3:

If Feature Map output of above step is of size $m \times n$ (no. of locations = $m \times n$),

then for each of these locations, SSD initializes 'k' default boxes of **random** size and shape. So, there will be $k \times m \times n$ default boxes with.

Step -4:

For each of the $k \times m \times n$ default boxes, SSD calculates the following :

4.1 c class scores,

where 'c' is the number of different classes that need to be tested. [For example, in our project, No. of classes (c) = 3 - Normal, Knife & Gun classes]

4.2 '4' offset parameters,

These parameters define the closeness of the algorithm calculated bounding box to the actual bounding box taken as input. They are:

$\Delta c_x, \Delta c_y$ - Change in values of (x,y) or error calculated for the starting point of bounding box.

$\Delta w, \Delta h$ - Change in value of (x,y) or error calculated for the width and height of bounding box.

So, there will be a total of $(c+4) \times k \times m \times n$ parameters in total for all locations

Step -5:

These $(c+4) \times k \times m \times n$ parameter values will be recalculated for each iteration (or epoch) of training using backpropagation gradient descent approach.

Step -6:

At the end, out of 'k' default boxes, the bounding box with minimum error (i.e., bounding box yielding minimum error of this $\{\Delta c_x, \Delta c_y, \Delta w, \Delta h\}$ tuple) is retained. Out of 'c' class accuracy scores, the object in the bounding box is marked with label of the class with highest accuracy score.

III. RESULTS AND DISCUSSION

A. Dataset

For Weapon detection, data of weapon images [17] is gathered from benchmark sources, for training & testing




purpose. These weapons include Knives, guns & normal images.

Here, an image can fall under three classes.

1. Normal class (images of human without any weapon)
2. Knife class (where a knife or multiple knives exists in the image)
3. Gun class (where a gun or multiple guns exists in the image)

An XML file is created for each image manually to train the algorithm about the existence of object. This XML file generation is called as Labelling and it is done by a tool called LABELIMG [18]. Table 1 list the sample of train and test objects considered in the experiments. Three categories of objects are considered for training and evaluation. Normal

Table.1 Train and Test dataset taken

IMAGE	CLASSES	TEST	TRAIN
	Normal	55	105
	Knife	67	559
	Gun	29	57
All Classes	Total	151	721

cases (non-weapon), knife as a weapon, and gun as a weapon. A total of 721 images are used for training and 151 images are used for evaluation. Several evaluation parameters are considered to support the proposed models. Mean Average Precision (MAP), recall, IoU, loss.

Average Precision (AP):

The accuracy-recall curve can be condensed into a single number that represents the average of all precisions, called the average precision (AP). The following formula is used to compute the AP. The difference between the current and subsequent recalls is computed using a loop that iterates through all precisions and recalls, and the result is multiplied by the current precision. Stated otherwise, the AP is equal to the sum of weighted precisions at each threshold, with the weight representing the gain in recall.

$$AP = \sum_{x=0}^{x=n-1} [Rec(x) - Rec(x+1)] * Prec(x) \quad (1)$$

where,

n = Number of thresholds, Rec(x) is the recall at xth definite value, Prec(x) is the precision at xth definite value.

Fig. 2. Gives the graph for Mean Average Precision as one of the main evaluation parameters. For the selected sub samples an MAP of 0.87 is achieved that is considerable. Fig. 3. Tells

about the recall metric. Recall shows how well all the relevant objects are selected. It is the ratio of TPs to (TPs + FNs).

IoU (Intersection over union)

The overlap between two borders is measured by IoU. This is how the degree to which our estimated border is calculated and the ground truth (the actual object boundary) overlap. To determine if a forecast is a true positive or a false positive, an IoU threshold (let's say 0.5) is predefined for some datasets.

$$IoU = \frac{Intersection\ Area}{Union\ Area} \quad (2)$$

$$class(IoU) = \begin{cases} Positive \rightarrow Iou \geq Threshold \\ Negative \rightarrow Iou < Threshold \end{cases} \quad (3)$$

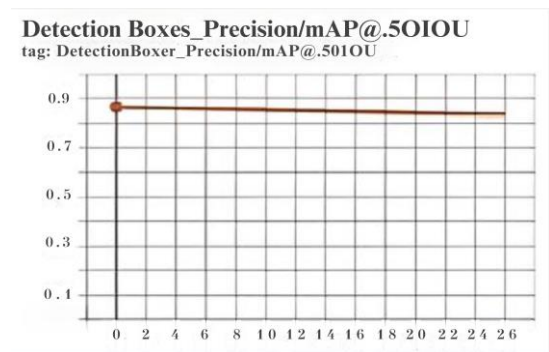


Fig. 2. Mean Average Precision (MAP) 0.87

As show in Fig. 2 a value of greater than 0.8 MAP is treated as excellent performance by the model.



Fig. 3. Recall

As shown in Fig. 3, a recall value of above 0.85 is achieved by the VGG16-SSD model indicates robustness in detecting relevant objects, especially in applications where comprehensive detection and sensitivity are critical.

Fig. 4. Shows how the learning rate is tuned. The dot shows the chosen value. It is not too small that takes impractical wait times, and it is not too high to miss the optimal weights. A decreasing learning rate curve over epochs indicates that the learning rate is gradually reduced as the training progresses. This approach is often used in machine learning training to improve convergence, stabilize optimization, and fine-tune the model's performance.

Loss/classification_loss
tag: Loss/classification_loss

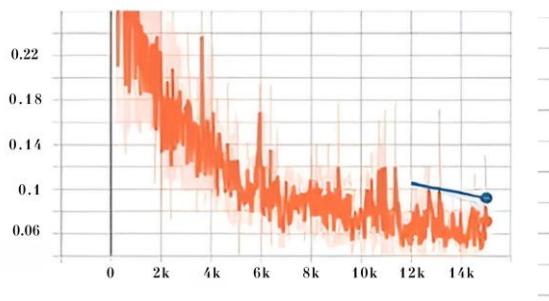


Fig. 5. Classification Loss

Fig. 4. Learning Rate

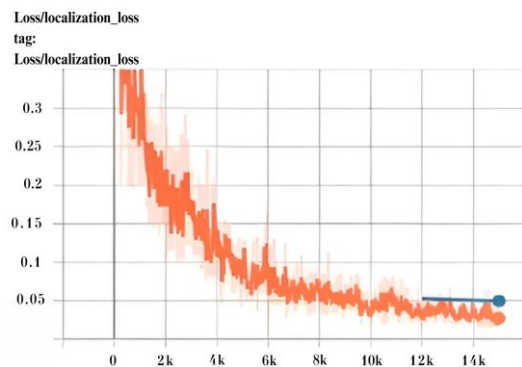


Fig. 6. Localization Loss

Fig. 5 shows the loss curve related to classification. Fig. 6. Is the curve related to localization loss. The decreasing trend of classification and localization losses over epochs in object detection tasks signifies improved accuracy, precision, convergence, and overall quality of object detection, validating the effectiveness of the training process and the model's performance improvements. Fig. 7 gives the drawing of normalized total loss, and the reducing curve over the epochs signifies the quality of the model. Fig. 8 is the sketch of regularization loss. The decreasing curve of regularization

loss is indicative of effective regularization, optimal model complexity, robustness to noise, improved generalization, stable training, and provides guidance for hyperparameter tuning in machine learning models used for object detection and other tasks. Fig. 9 show the output screen click of the real-time testing of the surveillance prototype. The system is able to localize and box the knife with a probability of 53% and a closed fist normal case with a probability of 79%.

Loss/normalized_total_loss
tag: Loss/normalized_total_loss

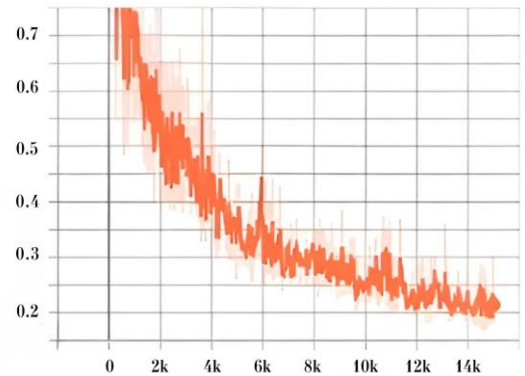


Fig. 7. Normalized_Total_Loss

Loss/regularization_loss
tag: Loss/regularization_loss

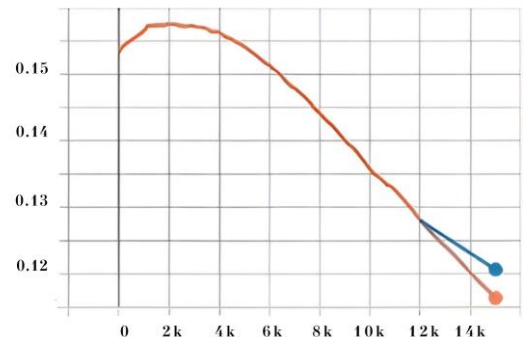


Fig. 8. Regularization_Loss

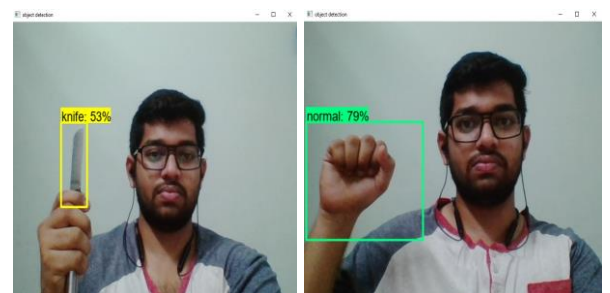


Fig. 9. Output of the system on real-time images

The detection rate or accuracy of weapon detection is calculated with the ratio of true cases of weapons detected to the total number of weapons in the test set. A detection rate of 87% is obtained in the experiments.

IV. CONCLUSION

Surveillance systems can be automated to aid the legal authorities and security personnel to generate alerts if any person is carrying illegal weapons within the scope of the camera footage. Recent deep learning variations, VGG-16 and SSD frameworks are used to efficiently identify a knife and gun in the hand of the person under surveillance. The model is trained and evaluated, and the goodness is proved through various measure of performance like MAP, recall, IoU, loss etc. With a sub sample of the whole dataset used for train and test, an accuracy of nearly 87% is achieved. Further, alert can be generated to the registered party in case of any malignant event. In the SSD framework can be compared against latest versions of YOLO, and other recent object detection frameworks like FR-CNN, HOG, Detectron2, EDET, etc. as the future direction. Accuracy can be improved with the aid of data augmentation tasks like noise insertion, cropping, rotation, flipping, and qualitative generalized learning. This can be the future scope in this domain.

V. REFERENCES

- [1] A. Krizhevsky, I. Sutskever and G. E Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM.*, vol. 60, no. 12, p. 84-90, 2017.
- [2] R. Girshick, J. Donahue and T. Darrell, "Rich feature hierarchies for accurate object detection and semantic segmentation," 2014, *arXiv:1311.2524*.
- [3] A. G. Howard, "Some Improvements on Deep Convolutional Neural Network Based Image," 2013, *arXiv:1312.5402*.
- [4] R. B. Girshick, "Fast R-CNN," 2015, *arXiv:1504.08083*.
- [5] R. Shaoqing, R. Girshick and H. Kaiming, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, p. 1137-1149, 2015.
- [6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed C. Fu and A. C. Berg, "SSD: Single Shot MultiBox Detector," in *Computer Vision – ECCV 2016* (Lecture Notes in Computer Science), vol. 9905, B. Leibe, J. Matas, N. Sebe, M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.
- [7] A. G. Howard, Z. Menglong, C. Bo, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 2017, *arXiv:1704.04861*.
- [8] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proc. Int. Conf. Eng. Tech. (ICET)*, Antalya, Turkey, Aug. 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.
- [9] K. S. Chahal and K. Dey, "A Survey of Modern Object Detection Literature using Deep Learning," 2018, *arXiv:1808.07256*.
- [10] M. T. Bhatti, M. G. Khan, M. Aslam and M. J. Fiaz, "Weapon Detection in Real-Time CCTV Videos Using Deep Learning," *IEEE Access*, vol. 9, pp. 34366-34382, 2021.
- [11] P. Yadav, N. Gupta and P. K. Sharma, "A comprehensive study towards high-level approaches for weapon detection using classical machine learning and deep learning methods," *Expert Syst. Appl.*, vol. 212, pp. 1-20, 2023.
- [12] K. P. Vijayakumar, K. Pradeep, A. Balasundaram and A. Dhande, "R-CNN and YOLOV4 based Deep Learning Model for intelligent detection of weaponries in real time video," *Math. Biosci. Eng.*, vol. 20, no. 12, p. 21611–21625, 2023.
- [13] N. T. K. Tram, D. T. Son and A. V. Thái, "Weapon Detection Using Deep Learning," in *Proc. 12th Int. Symp. Inf. Commun. Tech. (SOICT)*, Ho Chi Minh, Vietnam, Dec. 2023, pp. 101-109, doi: 10.1145/3628797.3628967
- [14] A. Kiran, P. Purushotham and D. D. Priya, "Weapon Detection using Artificial Intelligence and Deep Learning for Security Applications," in *Proc. 2022 Int. Conf. Adv. in Smart Secur. Intell. Comput. (ASSIC)*, Bhubaneswar, India, Nov. 2022, pp. 1-5, doi: 10.1109/ASSIC55218.2022.10088403.
- [15] A. Kambhatla and K. R. Ahmed, "Firearm Detection Using Deep Learning," in *Intelligent Systems and Applications* (Lecture Notes in Networks and Systems), vol. 544, K. Arai, Eds. Cham, Switzerland: Springer, 2022, pp. 200-218, doi: 10.1007/978-3-031-16075-2_13
- [16] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 2014, *arXiv:1409.1556*.
- [17] *Guns-Knives Object Detect*. Accessed: Jan. 10 2024. [Online]. Available: <https://www.kaggle.com/datasets/iqmansingh/guns-knives-object-detection/>
- [18] *LabelImg*. Accessed: Jan. 10 2024. [Online]. Available: <https://pypi.org/project/labelImg/>