Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.9854895

A CUSTOM DEEP CONVOLUTIONAL NEURAL NETWORK CDNN - (WITH YOLO V3 BASED NEWLY CONSTRUCTED BACKBONE) FOR MULTIPLE OBJECT DETECTION

S.T.Santhanalakshmi[1] and Rashmita Khilar [2]
[1]Research Scholar, Department of Computer Science and Engineering, Saveetha School of Engineering, SIMATS, Chennai-602117, Email: santhanalakshmi.phd2020@gmail.com
[2]Professor, Department of Information Technology, Saveetha School of Engineering, SIMATS, Chennai-602117, Email: rashmitakhilar.sse@saveetha.com
*Corresponding Author: Rashmita Khilar[2]

ABSTRACT
One of the major fields of research in automation is Object Detection and it has been widely applied in the domains including Image Retrieval (IR), medical diagnosis, Automated Vehicle System (AVS), Surveillance etc. As the application domain increases, the challenges also increased and most of the challenges are still unsolved due to the inefficiency of pre-existing models/architectures in detecting the small and occluded objects in an image. The resolution of input could be increased to detect small objects in a better way, or use techniques such as image sharpening or contrast adjustment to enhance images with low contrast or underexposure. Ensembling can also be done for multiple object detection models to improve overall performance. The proposed research work on CDNN model tries to improve the efficiency in detecting the small and occluded objects without sacrificing the processing time. In the proposed model, the existing backbone has been replaced by custom deep convolutional neural network with added augmentation layers. It is found that the proposed model improved the accuracy of the image detection significantly for small and occluded objects, even if the object is far away from the focus of the camera. With proper feature selection along with hyper parameter tuning, the proposed Custom deep convolutional neural network model (CDNN) resulted with an accuracy rate of 99.02277%.
Keywords: Object Detection, Deep Learning, YOLO, Computer Vision, Deep Convolutional Neural Network, Augmentations.

INTRODUCTION
An object or any item can be located and found in a photograph by a human without any difficulty. But in Computer Vision (CV), for the so called human visible device it is difficult. Here, the complicated responsibilities like figuring out one or two objects are carried out rapidly and can be corrected with few limitations. Today, with the provision of big data, cloud computing, high speed processors, machine learning algorithms, it is possible to locate and identify greater number of small objects with excessive accuracy. Object detection is mainly done through classification and regression algorithms, that falls under supervised and unsupervised approaches that detects the classes or categories of the objects by means of proper training, hyper parameter tuning etc. During testing, the target is achieved through good selection of Machine learning / Deep learning (ML / DL) algorithms as per the dataset. The

classification models fall under two categories i.e., linear models and non-linear models. Under linear models, algorithms such as Logistic Regression (LR) [1], Support Vector Machines (SVM) [2] plays a vital role in image classification. In non-linear approach, algorithms like K-Nearest Neighbour the proposed (K-NN) [3], Kernel SVM, Naïve Bayes (NB), Decision Tree Classifier (DTC) [4], Random Forest (RF) [5] provides the solution for both binary / multi class problems.

Finding correlations between dependent and independent variables is done using regression. Regression models come in a variety of forms, including simple linear regression, multiple linear regression, polynomial regression, support vector regression, decision tree regression, and random forest regression. In a classification algorithm, variables are matched to predetermined classes using a mapping function, and the prediction uses unordered discrete values. Here the calculation is done by measuring the accuracy. Whereas in regression, the mapping function maps to continuous values in an ordered fashion and error calculation has been done by measuring the Root Mean Square Error (RMSE). In many research works, both the classification and regression approaches have been utilized to detect the object in an image. The object detection locates the presence of object within a bounding box, or the classes of objects located and labelled within the bounding box. Multiple number of bounding boxes and its classes may be marked based on the presence of objects.

The main objective of object detection in the real-world applications includes identifying objects with good accuracy along with less computational complexity in the uncertain environment. The major focus during deploying the model includes object localisation, object detection, viewpoint variation, multiple aspect ratios and spatial sizes declaration, deformation, occlusion, lighting, to define whether object is in a cluttered or textured background, intra-class variation, real time detection speed, dataset limitations etc. Though the recent works are giving better efficiency in detecting the objects, yet they continue to struggle in finding the small objects especially the overlapped or hidden occluded objects. The real time applications of object detection include security, military surveillance, transportation, medical science, intruder detection and much more day to day life applications. Face detection, facial landmark localisation, head position estimation, and gender identification have all been widely recommended and put into practise with deep learning models in the context of security. The military makes use of Detection Flyer, Topographic Survey, and Remote Sensing for object detection. Automatic driving, traffic sign recognition, and licence plate recognition are all done under transportation. Deep learning techniques have been proposed in the field of medical science for medical image detection, cancer/non-cancer detection, and health care monitoring with CAD models. Pattern detection in images, event detection (Internet news of celebrations, tragedies, speeches, and elections), rain/shadow detection, Image Caption Generation (ICG), species identification, etc. are only a few examples of how object detection is used in daily life. For ICG, the semantic information of images is captured and expressed using Natural Language Processing (NLP) which is the most challenging task. The foundation of all these applications is object detection where the accuracy of top-level classification and localizations still remains challenging. This proposed model tries to find better solution for object detection using the available dataset in the repositories.

RELATED WORKS

PASCAL VOC and MS-COCO are the two popular datasets used in traditional object detection models. A Single data contains the image under text file which contains the coordinates of the object and its class. COCO dataset contains 80 object instances and PASCAL VOC has 20 object classes. Both the dataset has images which includes person, dog, cat, car etc. The availability of the large collection of images in both COCO and PASCAL datasets allows both the dataset to be used by many researchers. The primary evaluation metrics used in both COCO and PASCAL dataset are Mean Average Precision (MAP). A model begins to learn from the noise and erroneous data entries in the data set when it is trained with such a large amount of data. Due to too many details and noise, the model fails to appropriately categorise the input. This is called as overfitting. For example, to provide variations in the base dataset we have added augmentation to the images in the dataset. Adding Rotation to the image, will definitely affect the coordinates of the object and hence we are using other augmentation methods like grey scaling, saturation altering, brightness altering, hue altering. The kernel values works efficiently. All the thresholds for the augmentations are assigned randomly. Each and every image from the dataset will have an altered version of it in the augmented dataset. This will reduce the possibility of overfitting Fast RCNN [22] fixes the disadvantages of RCNN and SPP net and improves speed and accuracy, higher detection quality MAP, training done in single stage, no disk storage is required for features caching. The SSD [23] model extends a base network with a number of feature layers that forecast the offsets to default boxes with various scales and aspect ratios as well as the corresponding confidences. On the VOC2007 test, SSD with a 300 300 input size greatly exceeds its 448 448 YOLO equivalent in terms of accuracy while enhancing speed. Based on VGG16 with some subsampling, this experiment. Within YOLO [24], Bounding boxes and class probabilities are directly predicted by a single neural network from complete images in a single assessment. Since the entire detection pipeline consists of a single network, detection performance can be optimised from beginning to end.

When it comes to common detection tasks like PASCAL VOC and COCO, YOLOv2 [25] is cutting edge. The same YOLOv2 model can run at different scales using an innovative, multi-scale training strategy, providing a simple trade-off between speed and accuracy. On VOC 2007, YOLOv2 receives 76.8 mAP at 67 FPS. YOLOv2 achieves 78.6 mAP at 40 FPS, exceeding cutting-edge techniques. YOLOv3 [26] is a good detector. Its fast, it’s accurate. It’s not as great on the COCO average AP between 0.5 and 0.95 IOU metric. But it’s very good on the old detection metric of 0.5 IOU. The experimental Dataset used in this study, created by Yajing Guo Xiaoqiang GuoZhuqing Jiang Yun Zhou [7], is PASCAL VOC. On the PASCAL VOC2007 trainval set, this model built on VGG16-Net achieved a mAP of 71.6%, which are 5.6 points higher than R-CNN's (66.0%) and 4.7 points higher than R-CNN's (66.9%). To show that this work has outperformed previous state-of-the-art approaches in object detections, Jawadul H. Bappy and Amit K. Roy Chowhury conducted experiments [8] on variations of datasets such PASCAL called SUN, MIT-67 Indoor, and MSRC datasets. With accuracy of 38.72%, the proposed R2-CNN-FT is much faster. FanjieMeng, Yi Xiao, Xinqing Wang, Peng Zhang, and Faming Shao [12] When PASCAL VOC 2007 and 2012 Data Sets were compared against R-CNN, YOLO v3, SSD512, and other object detection algorithms, the mean Average Precision increased by 6.857%. ZhaohuiZheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye,

DongweiRen [15] declared that PASCAL VOC and MS COCO GIoU can improve the performance with 3.29% AP and 6.02%, CIoU gains 5.67% Ap, 8.95%. Danyang Cao Zhixin Chen Lei Gao[6] found The speed of this model is tested with PASCAL VOC07 which includes 4500 test pictures. We have trained different versions of the proposed model in PASCAL VOC and MS-COCO. We have constructed multiple versions of backbones from the proposed architecture which were trained on the combination of augmented datasets and Input layer dimensions which resulted in an improvement in both mAP and computing cost and it also overcame the issues faced by the previous architectures.

PROPOSED METHODOLOGY

In the proposed CDNN model, we have altered backbone Darknet53 of YOLOv3 with complete change in its backbone with the proposed architecture. The filters are generalised hence it redefines the object with any of the proposed or dimension or brightness. Here the kernel values work efficiently.

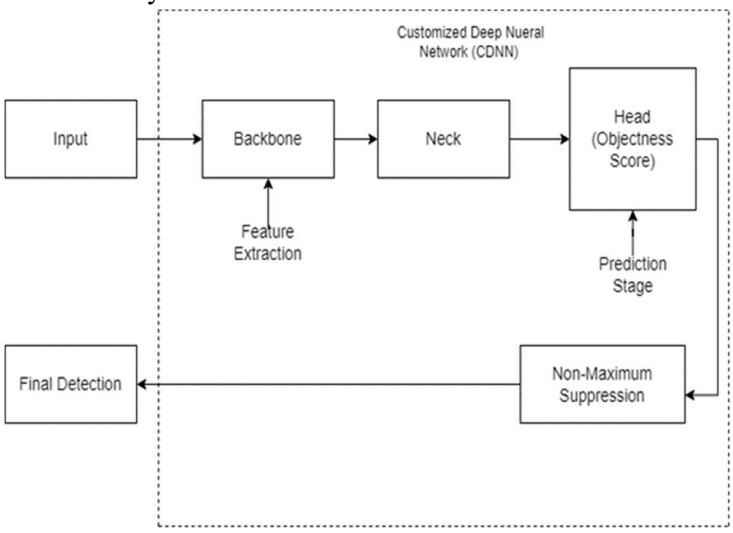


Fig 1: General Architecture of CDNN model

3.1 Pre-processing

Before the image is fed to the CNN model, pre-processing methods such data centering, resizing, and scaling were performed on it. To make the training images suitable for the network, each training image was resized to 256 256 pixels. To create more training images, every training image was elastically biased. The suggested randomly constructed elastic deformed images were created for each original training image and then downsized to 256 256 pixels.

3.1.1 Resizing

Additionally, each training image was enhanced to correct several objects belonging to the same class. Following the addition of these additional distortions, an extra 20% of training examples were created for each original training image. Each training mask was rotated and transformed in order to create an additional 18,000 training photos, reaching a total of 20000 training images.

3.1.2 Normalization

To mitigate the variations produced by the size distribution of objects, the input photos that have been resized are subsequently normalized. In order to speed up the operation of the suggested model, each convolution layer performs Batch Normalization (BN), which normalises the input X. The means for each feature, or the squared means across the first and final dimensions, are expressed as E[(X(k))2], the average for each feature E[(X(k))], which includes the average for the first and last dimensions. The buffers to store the mean E[(X(k))] and variance Var [(X(k)) exponential moving averages

When the input X Є $R^{B\times C\times H\times W}$ is a batch of image representations, where B is a batch size, C is the number of Channels, H is the height and W is the width. γ Є R^C and β Є R^C Here both γ and β values are predicted based on number of channels C. ε

$$BN(X) = \gamma.\frac{X - E_{B,H,W}[X]}{\sqrt{Var_{B,H,W}[X]+\varepsilon}} + 1 \qquad \text{(i)}$$

When input X Є $R^{B\times C}$ is a batch of embeddings, where B is the batch size and C is the number of features. . γ Є R^C and β Є R^C

$$BN(X) = \gamma.\frac{X - E_{B}[X]}{\sqrt{Var_{B}[X]+\varepsilon}} + 1 \qquad \text{(ii)}$$

When the input X Є $R^{B\times C\times L}$ is a batch of a sequence embeddings, where B is the batch size, C is the number of features and L is the length of the sequence. γ Є R^C and β Є R^C

$$BN(X) = \gamma.\frac{X - E_{B,L}[X]}{\sqrt{Var_{B,L}[X]+\varepsilon}} + 1 \qquad \text{(iii)}$$

Here the channels are the number of features in the input and ε is used for numerical stability

3.2 Object detection using CDNN

The proposed image object detector adopts YOLO's backbone network and adds the new trick in convolution operation. The proposed architecture uses the same Darknet-53 as backbone with layers repeating certain number of times additionally. The Architecture consists of 24 convolutional layers with size being varied between 3x3 and 1x1. Alternating 1x1 convolutional layers reduce the features space from preceding layers. ReLU is used as the activation function in the convolution layers and the scale layer which is a dense layer uses softmax activation function. An Average pooling layer is added to every convolution block to reduce the dimensions of the image.

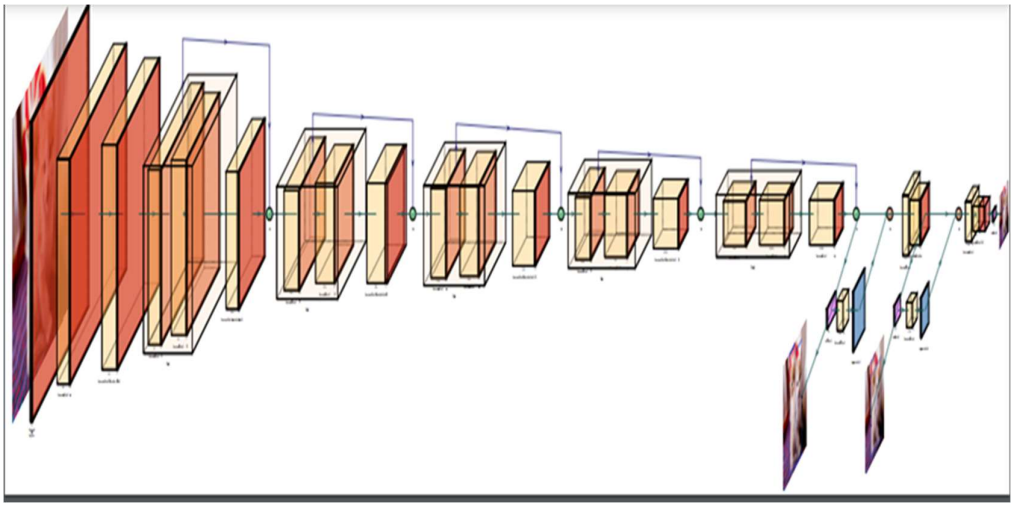


Fig.2: Model 3D Architecture diagram of the proposed Custom Deep Convolutional Neural Network - A Custom Deep Convolutional Neural Network CDNN (With YOLO V3 Based Newly Constructed Backbone) For Multiple Object Detection with altered Darknet 53 backbone with multiple residual blocks and scaling factors undergone with upsampling and With proper feature selection along with hyper parameter tuning, the proposed Custom deep convolutional neural network model resulted with an accuracy rate of 99.02277%.

3.2.1 Non-Max suppression

Numerous computer vision tasks use a method known as Non Maximum Suppression (NMS). To select one thing (like bounding boxes) from a huge number of overlapping elements, a class of algorithms is utilised. We can choose the selection criteria to acquire the desired outcomes. The requirements are most typically an overlap measure and a probability number (e.g., intersection over union). IOU loss only functions when the predicted bounding boxes overlap with the ground truth box. The largest overlap between the anticipated bounding box and the ground truth box is required by the IOU loss function. The speed of convergence of the IOU loss is low.

Equation of IOU and IOU loss function,

$$IoU = \frac{|B \cap B^{gt}|}{|B \cup B^{gt}|} L_{IoU} = 1.\frac{|B \cap B^{gt}|}{|B \cup B^{gt}|} \qquad \text{(iv)}$$

We used ADAM (Adaptive Moment Estimation) because it is crucial that the proposed predictions are accurate and optimised. This method computes the

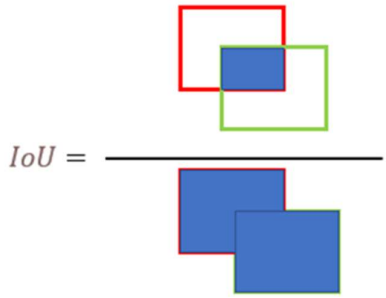


Fig 3: IOU Loss Function

exponentially weighted average of previous gradients as well as the exponentially weighted average of the squares of previous gradients. In order to increase the performance of the proposed model; the softmax `entropy function to test the reliability of the model.

$$\sigma(\vec{Z}_i) = \frac{e^{zi}}{\sum_{j=1}^{k} e^{zj}} \qquad \text{(v)}$$

Where σ represents Softmax function,

$\vec{Z}$ is Input Vector ,

e^{zi} is Standard exponential function for input vector,

K is the number of classes in the multi class classifier,

$e^{zj}$is Standard exponebtial function for output vector

Global Activation Function

In the proposed model we have used Leaky ReLU since it allows a small non zero, constant gradient α

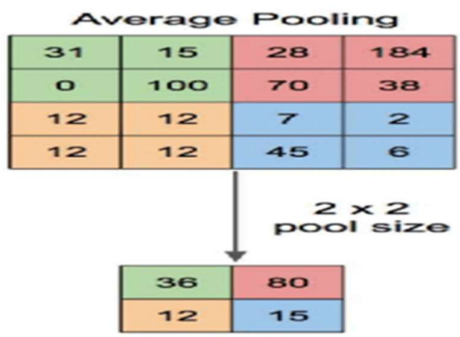


Fig 4: Average Pooling

3.2.2 Object detection

The proposed CDNN model has following layers

Two convolutional layers with 32 and 64 filters giving feature map output of size 128x128. A residual block of two convolution layers with 32 and 64 filters giving feature map of 128x128. This residual block is repeated 2 times. One convolution layer with 128 filters giving feature map output of size 64x64 is followed by a residual block of 2 convolution layers repeating two times. One convolution layer with 256 filters giving feature map output of size 32x32 is followed by a residual block of 2 convolution layers repeating 8 times. One convolution layer with 512 filters giving feature map output of size 16x16 is followed by a residual block of 2 convolution layers repeating 8 times. One convolution layer with 1024 filters giving feature map output of size 8x8 is followed by a residual block of 2 convolution layers repeating 12 times. One scale layer which outputs the class, and creates a bounding box (anchor box). One convolutional layer with256 filters giving a feature map output of size 1x1 is followed by an upsampling layer. Two convolutional layers with 256 and 512 filters is followed by a scale layer. One convolutional layer with 128 filters giving a feature map output of size 1x1 is followed by an upsampling layer. A scale layer comes after two convolutional layers with 128 and 256 filters. The next step in the suggested model is average pooling, a technique for determining the average value for individual feature map patches. Moreover, it is frequently used to produce a down- or pooled-sampled feature map. Typically, the convolutional layer is finished before the average pooling process. There is a modest degree of translation invariance

added by an average pooling. This demonstrates that a slight translation of the image has no impact on the values of the majority of pooled results. Compared to Max pooling, it aims to extract features more seamlessly.

Pseudocode:

```
# Load the customized yolov3 based backbone model and configuration file
model = load_ customized yolov3 based backbone _model(CDNN)
# Process each input frame or image
for input in input_data:
# Preprocess the input data
preprocessed_input = preprocess_input(input)
# Feed the input through the customized yolov3 based backbone model
detections = model(preprocessed_input)
# Apply non-maximum suppression
nms_detections= non_maximum_suppression(detections)
# Extract the relevant information from the detections
object_info = extract_object_info(nms_detections)
# Draw the bounding boxes and labels on the input image or frame
output = draw_output (input, object_info)
# Display or save the output image or frame
display_output(output)
```

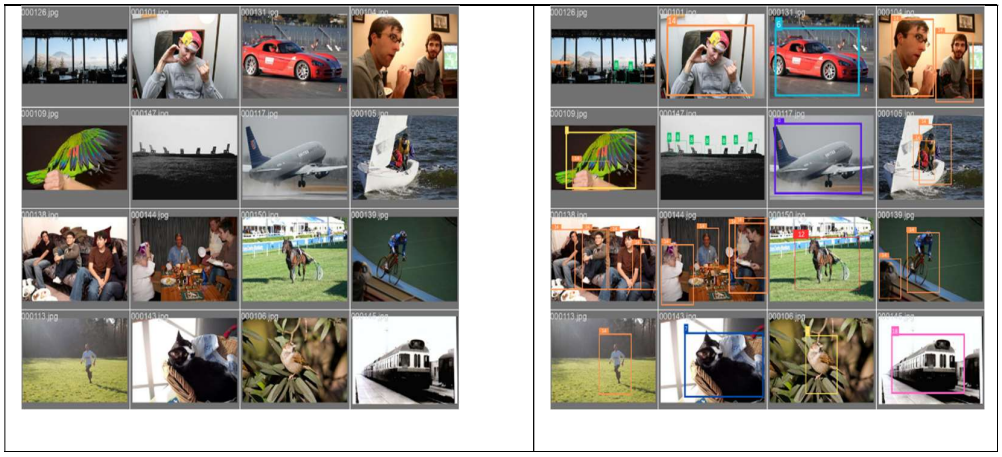
Type	Filters	Size	Output	Cycles
Convolutional	32	3×3	256×256	1
Convolutional	64	3×3/2	128×128	1
Convolutional Convolutional Residual	32 64	1×1 3×3	128×128	2
Convolutional	128	3×3/2	64×64	1
Convolutional Convolutional Residual	64 128	1×1 3×3	64×64	2
Convolutional	256	3×3/2	32×32	1
Convolutional Convolutional	128 256	1×1 3×3	32×32	8

Residual				
Convolutional	512	3×3/2	16×16	1
Convolutional Convolutional Residual	256 512	1×1 3×3	16×16	8
Convolutional	1024	3×3/2	8×8	1
Convolutional Convolutional Residual	512 1024	1×1 3×3	8×8	12
Scale 1	**OUTPUT 1 (Large objects)**			
Convolutional	256	1×1		1
Up sampling	Upscaling Block			
Convolutional	256	1×1		1
Convolutional	512	3×3		1
Scale 2	**OUTPUT 2 (Medium objects)**			
Convolutional	128	1×1		1
Up sampling	**Upscaling Block**			
Convolutional	128	1×1		1
Convolutional	256	3×3		1
Scale 3	**OUTPUT 3 (Small Objects)**			

Fig.4: Table of execution layers for the proposed Custom Deep Convolutional Neural Network- A Custom Deep Convolutional Neural Network CDNN (With YOLO V3 Based Newly Constructed Backbone) For Multiple Object Detection which is constructed on altered Darknet 53 backbone which includes residual blocks repeated multiple times, it also includes scale layers at three different places for detecting large, medium and small objects. Up sampling is done at the last two prediction layers to get an increased accuracy.

RESULT AND DISCUSSION

YOLO is a Convolutional Neural Network (CNN) that can quickly identify objects. CNNs are classifier-based systems that are able to analyse incoming images as organised arrays of data and spot patterns in them. Darknet-53 is more powerful than Darknet-19 and more effective than rival backbones (ResNet-101 or ResNet-152) since it uses 53 convolutional layers as opposed to the preceding 19 convolutional layers. The same Darknet-53 serves as the framework for the proposed design, with layers repeating a specific number of times in addition. As a result, a totally new backbone is created. This architecture was created with the goal of accurately and precisely identifying extremely tiny things.



5.a **5.b**

Fig. 5.a) Sample Input of image with multiple objects 5.b) Output of the proposed Custom Deep Convolutional Neural Network CDNN

The precision for small objects in YOLOv2 was incomparable to other algorithms because of how inaccurate YOLO was at detecting small objects. The mean average precision score for YOLOv3 416 is 55.3 and YOLOv3 608 is 57.9, which is significantly improved in the proposed model to 58. 803186.There is a marginal improvement of 0.903186 compared to yolov3 608 and a significant improvement of 3.503186 compared to yolov3 416.The accuracy in detecting small objects is increased from all the previous versions of YOLO in the proposed model. Small object detection accuracy is 82% for all the previous versions of yolo, whereas the proposed model increases the accuracy score to 86%. The classification accuracy for small objects for YOLOv3 is 47%, which is also increased to 47.2%. Over all object detection of yolo is 95% and the proposed is 99.02277%

In the proposed model, the overall output of hidden convolutional layers would undergo upscaling and upsampling in the head to get resolute output image. The upscaling uses two formulas to perform the task and both use certain derived variables. First part of upscaling is PSNR-peak signal-to-noise ratio. PSNR uses mean squared error (MSE) as the optimizing variable in equation (i)

$$\text{PSNR} = 10\log_{10}\left(\frac{S^2}{MSE}\right) \qquad \text{(vi)}$$

$$\text{MSE} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(Y(i,j) - X(i,j))^2 \qquad \text{(vii)}$$

MSE in equation (ii) is generally very small value; it can be approximated and increased.

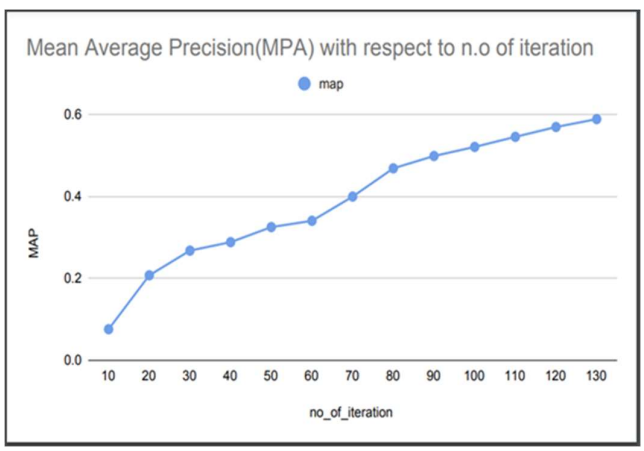
M and N are dimensions of original image. We have taken a particular frame and increase the value. The X and Y functions yields the value of complete up scaling function.
S Represents the image size with its pixel representation and in the proposed model we have used S value as 416. X and Y are the images. X (i,j) and Y(i,j) are vectors. M and N in MSE are variables to calculate the sum of the error.
SSIM in equation (iii) is the Second part of the upscaling.

$$\text{SSIM}=\frac{(2\mu_x\mu_y+c1)(2\sigma_x\sigma_x+c2)}{(\mu_x^2+\mu_y^2+c1)(\sigma_x^2+\sigma_y^2+c2)},\begin{cases}c1=(K_1L^2)\\c2=(K_2L^2)\end{cases} \quad \text{(viii)}$$

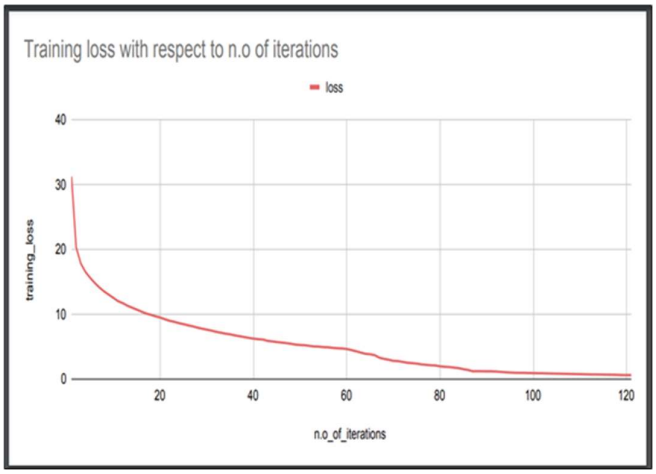
Here μ_x and μ_y are mean value of image X and Y. σ_x and σ_yare the standard variance of X and Y. c1 and c2 are two stabilizing constants with L where L is a distributed value between [-3, +3], where this Constant was distributed between [-2.50 to +2.50] [74]. And this value is modified to get the exact size and resolution for the proposed architecture. The dynamics of pixel value k1 and K2 are generally set to be 0.0133 and 0.0372 respectively. These two parts are combined using a python function.
The mAP is calculated by finding Average Precision (AP) for each class and then average over a number of classes using the Mean Average Precision Formula

$$mAP=\frac{1}{N}\sum_{i=1}^{N}APi \quad \text{(ix)}$$

The mAP incorporates the trade-off between precision and recall and considers both false positives (FP) and false negatives (FN). This property makes mAP a suitable metric for most detection applications. mAP is improved from 55.3 (yolov3 416), 57.9 (yolov3 608) to 58.803186 (proposed_model416) 0.903186 improvement compared to yolov3 608 and 3.503186improvemint compared to yolov3 416. Small object detection accuracy 82% for all the previous versions of yolo, over models accuracy of detecting small object is 86% the classification accuracy for small objects for yolov3 is 47% and the proposed 47.2%. Over all object detection of yolo is 95% and the proposed CDNN is 99.02277% The following Table compared with existing models and results recorded.



Graph 1: Mean Average precision of the proposed CDNN model when compared with existing YoloV3(55.3) is 58.803186



Graph 2: Training loss with respect to number of iterations which gradually decreased and hence loss very less in CDNN.

CONCLUSION AND FUTURE WORK

The proposed CDNN architecture uses the same Darknet-53 as backbone with layers repeating certain number of times additionally. Hence a completely different backbone is formed. This architecture is developed with an objective of detecting even small objects with high precision and accuracy. The mean average precision score for YOLOv3 416 is 55.3 and YOLOv3 608 is 57.9, which is significantly improved in the proposed model to 58.803186. There is a marginal improvement of 0.903186 compared to yolov3 608 and a significant improvement of 3.503186 compared to yolov3 416.The accuracy in detecting small objects is increased from all the previous versions of YOLO in the proposed model. Small object detection accuracy is 82% for all the previous versions of YOLO, whereas the proposed model increases the accuracy score to 86%. The classification accuracy for small objects for YOLOv3 is 47%, which is also increased to 47.2%. The same model can also be used to find the shadow hidden and dim light object, if trained with a well-defined dataset.

REFERENCES

[1] Chinnalagu A, Durairaj AK. 2021. Context-based sentiment analysis on customer reviews using machine learning linear models. PeerJ Computer Science 7:e813 https://doi.org/10.7717/peerj-cs.813

[2] H. Alquran et al., "The melanoma skin cancer detection and classification using support vector machine," 2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), Aqaba, Jordan, 2017, pp. 1-5, doi: 10.1109/AEECT.2017.8257738.

[3] Ala'raj, M., Majdalawieh, M. & Abbod, M.F. Improving binary classification using filtering based on k-NN proximity graphs. J Big Data 7, 15 (2020).https://doi.org/10.1186/s40537-020-00297-7

[3] Kaur, H. and Kumari, V. (2022), "Predictive modelling and analytics for diabetes using a machine learning approach", Applied Computing and Informatics, Vol. 18 No. 1/2, pp. 90- https://doi.org/10.1016/j.aci.2018.12.004

[4] "Increasing the views and reducing the depth in random forest" AbolfazNadi, HadiMoradi https://doi.org/10.1016/j.eswa.2019.07.018 , Exper system with Applications Dec,2019

[5] Sandino, J.; Vanegas, F.; Maire, F.; Caccetta, P.; Sanderson, C.; Gonzalez, F. UAV Framework for Autonomous Onboard Navigation and People/Object Detection in Cluttered Indoor Environments. Remote Sens. 2020, 12, 3386. https://doi.org/10.3390/rs12203386

[6] "An improved object detection algorithm based on multi scaled and deformable convolutional neural network" Danyang Cao, Zhixin Chen, Lei Gao Springer2020.

[7] "Cascaded Convolutional Neural Networks for Object Detection"YajingGuo,XiaoqiangGuo,Zhuqing Jiang, Yun Zhou ,IEEE 2017.

[8] "CNN Based Region Proposals for Efficient Object detection", JawadulH.Bappy and AmitK.RoyChowhury, IEEE 2016.

[9] "Fast Object Detection in compressed JPEG Images" Benjamin Deguerre Clement Chatelain Gilles Gasso IEEE(ITSC)2019.

[10] "R2– CNN: Fast Tiny Object Detection in Large Scale Remote Sensing Images" Jiangmiao Pang Cong Li Jianping Shi ZhihaiXuHuajunFeng, IEEE Transaction on Geo Science and Remote Sensing 2019.

[11] "Few-shot object Detection on remote sensing images", Xiang Li, Jingyu Deng, Yi Fang, IEEE Transactions on Geoscience and Remote Sensing2021.

[12] "Object Detection based on Faster CNN Algorithm with Skip Pooling and Fusion of Contextual Information", Yi Xiao, Xinqing Wang, Peng Zhang, FanjieMeng and Faming Shao, Sensors 2020 MDPI.

[13] "Object Detection Based on an adaptive attention mechanism" Wei Li, Kai Liu, Lizhe Zhang, Fei Cheng Scientific Reports 2020.

[14] "Learning Rotation – Invariant and Fisher Discriminative Convolutional Neural Networks for Object Detection" Gong Cheng, Junwei Han, Peicheng Zhou and Dong Xu, IEEE ,2018.

[15] "Distance IoU Loss – Faster and Better Learning for bounding box regression" ZhaohuiZheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, DongweiRen, Association for the Advancement of Artificial Intelligence. 2019.

[16] "A Multispectral image based object detection approach in natural scene", Chuanyuan Zhao, Xiangjuan Li
2021 IEEE 6th International Conference on Intelligent Computing and Signal Processing.

[17] "Image Detection in Noisy Images", KushagraYadav, Dakshit Mohan, Anil Singh Parihar, IEEE Explore (ICICCS 2021)

[18] "CNN – Based Target Detection and classification when sparse SAR Image Dataset is Available", Hui Bi (Member IEEE), Jiarui Deng, Tianwen Yang, Jian Wang and Ling Wang, IEEE 2021

[19] "Bilateral Attention Network for RGB -D Salient Object detection (Algorithm aims to explore objects or regions more", Zhao Zhang, Zheng Lin, Jun Xu, Wen Da Jin, Shao ping Lu, Deng -Ping Fan, IEEE Transaction on Image Processing, 2021.

[20] "YOLO Z(Zoom): Improving small object detection in YOLO v5 for autonomous vehicles", AduenBenjumea
IzzeddinTeeti, Fabio Cuzzolin, Andrew Bradely, Visual Artificial Intelligence Laboratory, Oxford University, UK

[21] "Rich features hierarchies for accurate object detection and semantic segmentation", Ross Girshick Jeff Donahue, Trever Darrell, Jitenra Malik.
[22] "Fast RCNN", Ross Airsick, Microsoft research 2015.
[23] "Single Shot Multibox detector", Wei Liu, DragomirAnguelov, DumitruErhan, Christian Szegedy,
Scott Reed, Cheng-Yang Fu, Alexander.
[24] "You Only Look Once: Unified, Real-Time Object Detection", JosephRedmon, SantoshDivvala, Ross Girshick, Ali Farhadi 2016.
[25] "YOLO9000: Better, Faster, Stronger", Joseph Redmon, Ali Farhadi 2016.
[26] "YOLOv3: An Incremental Improvement", Joseph Redmon, Ali Farhadi2018.
[27] "Deep Residual Learning for Image Recognition", Kaiming He, Xiangyu Zhang, ShaoqingRen, Jian Sun, Microsoft Research 2015. (Base taken)
[28] "Recent Advances in Convolutional Neural Networks", JiuxiangGu, ZhenhuaWang, JaseonKuen, Lianyang Ma, Amir shahroudy, Bing Shuai, TingLiu, Xinxing Wang, Li Wang, Gang Wang, Jianfeicai, Tsuhan Chen 2017.
[29] "Rich feature hierarchies for accurate object detection and semantic segmentation, Tech report (v5)", Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, 2014, UC Berkeley.
[30] "SSD: Single Shot MultiBox Detector", Wei Liu, DragomirAnguelov, DumitruErhan, Christian Szegedy, Scott Reed, Chen-Yang Fu, Alexander C.Berg, 2016.
[31] D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox, and A. Farhadi. Iqa: Visual question answering in interactive environments. arXiv preprint arXiv:1712.03316, 2017. 1
[32] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016. 3
[33] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. 3
[34] I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, A. Veit, S. Belongie, V. Gomes, A. Gupta, C. Sun, G. Chechik, D. Cai, Z. Feng, D. Narayanan, and K. Murphy. Open images: A public dataset for large-scale multi-label and multi-class image classification. Dataset available from https://github.com/openimages, 2017. 2
[35] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2117–2125, 2017. 2, 3
[36] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. ´ Focal loss for dense object detection. arXiv preprint arXiv:1708.02002, 2017. 1, 3, 4
[37] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft coco: Com- ´ mon objects in context. In European conference on computer vision, pages 740–755. Springer, 2014. 2
[38] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.- Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In European conference on computer vision, pages 21–37. Springer, 2016. 3

[39] I. Newton. Philosophiaenaturalis principia mathematica. William Dawson & Sons Ltd., London, 1687. 1

[40] J. Parham, J. Crall, C. Stewart, T. Berger-Wolf, and D. Rubenstein. Animal population censusing at scale with citizen science and photographic identification. 2017. 4

[41] J. Redmon. Darknet: Opens the proposedce neural networks in c. http://pjreddie.com/darknet/, 2013–2016. 3

[42] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, pages 6517–6525.IEEE, 2017. 1, 2, 3

[43] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. arXiv, 2018. 4

[44] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497, 2015. 2

[45] O. Russakovsky, L.-J.Li, and L. Fei-Fei. Best of both worlds: human-machine collaboration for object annotation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2121–2131, 2015.

[46] Brooks, A.C., et al. (2008) Structural Similarity Quality Metrics in a Coding Context: Exploring the Space of Realistic Distortions. IEEE Transactions on Image Processing, 17, 1261-1273.https://doi.org/10.1109/TIP.2008.926161

[47] Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study Umme Sara, MoriumAkter, Mohammed ShorifUddin, Jtheproposednal of Computer and Communications Vol.7 N0.3, March 2019.

[48] Analogy. Wikipedia, Mar 2018. 1

[49] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. International jtheproposednal of computer vision, 88(2):303– 338, 2010. 6

[50] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg.Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659, 2017. 3

[51] P Ashok Babu, L Kavisankar, Jasmine Xavier, V Senthilkumar, Gokul Kumar, T Kavitha, A Rajendran, G Harikrishnan, A Rajaram, Amsalu Gosu Adigo. 2022. Selfish Node Detection for Effective Data Transmission Using Modified Incentive Sorted Pathway Selection in Wireless Sensor Networks. Wireless Communications and Mobile Computing. 2022.

[52] B. Anitha, & Rajaram, A. (2014). Efficient Position Based Packet Forwarding Protocol For Wireless Sensor Networks. Journal of Theoretical & Applied Information Technology, 69(2).

[53] Rahamat Basha S., Chhavi Sharma, Farrukh Sayeed, AN Arularasan, PV Pramila, Santaji Krishna Shinde, Bhasker Pant, Dr. A Rajaram, Alazar Yeshitla. 2022. “Implementation of reliability antecedent forwarding technique using straddling path recovery in manet”. Wireless Communications and Mobile Computing. 1-9.

[54] S. Kannan., & Rajaram, A. (2012). QoS Aware Power Efficient Multicast Routing Protocol (QoS-PEMRP) with Varying Mobility Speed for Mobile Ad Hoc Networks. International Journal of Computer Applications, 60(18).

[55] C. R. Rathish., & Rajaram, A. (2017). Robust early detection and filtering scheme to locate vampire attack in wireless sensor networks. Journal of Computational and Theoretical Nanoscience, 14(6), 2937-2946.