Object Detection in Low Light

Naveen Jami, Harshitha S Gaddadhar
{ jami.n, gaddadhar.h } @husky.neu.edu
INFO 7390, Summer 2019, Northeastern University

***Abstract* - Low-light images are elusive to the human eye and computer vision algorithms, due to their low visibility and noise. Hence, our aim is to determine if object detection algorithms like YOLOv3 are as efficient in detecting objects in low light as they are in normal conditions. For this research, we used an open-source dataset named Exclusively Dark (ExDark) Image Dataset which is a low light image dataset. In this paper, we have compared the mAP (mean Average Precision) of YOLOv3 obtained on COCO dataset to that of the mAP obtained on ExDark dataset and identified that the efficiency of the algorithm is very less in low light when compared to normal image dataset. This paper sums up the contrast-based image enhancement techniques applied to preprocess our dataset and compares the results with our benchmark obtained on original ExDark dataset. Also, our insights on the comparison between training YOLOv3 model from the ground-up to that of YOLOv3 using transfer learning technique, have been presented.**

I. Introduction

Object detection is a computer vision technique for locating instances of objects in images or videos. Object detection and recognition tasks have always been the main focus in computer vision. There is constant research in order to find a detection system as efficient and powerful as humans. Traditionally normal light images are used for object detection systems. But for an image taken in low light conditions needs more preprocessing to identify and classify the objects. Our paper concerns the application of state of the art existing object detection algorithm on low light images.

II. YOLO (You look only once)

YOLO is a new approach to object detection where a single neural network predicts the bounding boxes and associated class probabilities from the full image in one evaluation.

YOLO trains on full images and directly optimizes detection performance. YOLOv3 (Figure 1) uses a new feature extractor called Darknet- 53 (Figure 2) with the layers originally trained on ImageNet. YOLOv3 has elements like residual blocks, skip connection, upsampling which was absent in the previous versions of YOLO. But the computational speed of the YOLOv3 is reduced due to the extensive layered fully convolutional underlying architecture.

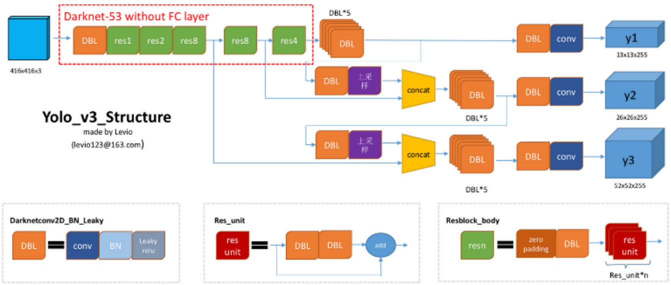
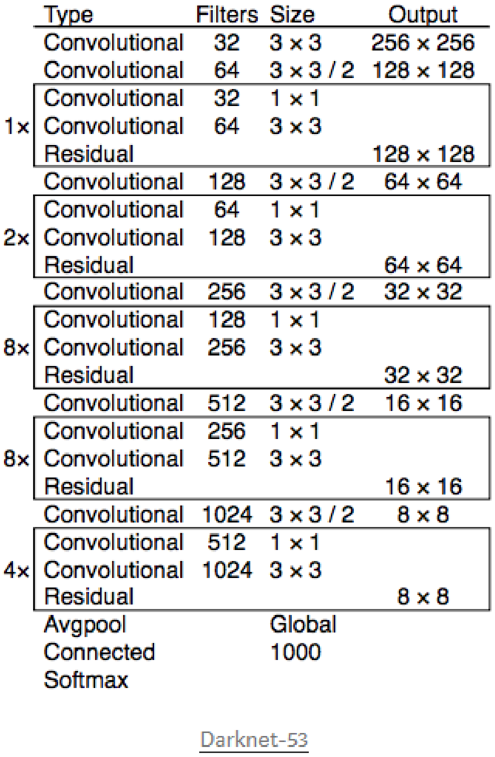


Figure 1



	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Darknet-53

Figure 2

In this paper, we have used pre-trained weights which has 80 classes (COCO dataset). The class labels are represented as c and their id's are numbered from 1 to 80. The image features learned by the convolutional layers are passed onto a classifier and regressor which makes the detection prediction. YOLOv3 uses 9 anchor boxes, 3 for each scale of detection.

III. EVALUATION METRIC

A. *mAP (mean Average Precision)*

Evaluation of the machine learning algorithm is an essential step to measure the performance of the model. The accepted way to evaluate models in Object Detection is using the **"mAP" Score (mean Average Precision)**. This metric is always evaluated against the ground truth data. For an object detection model, the ground truth data includes the image, the classes of the objects and the true bounding boxes of each of the object in the image. When a model returns predictions, most of them might have a very low confidence score associated with it. Thus only the predictions with an admissible score above a threshold are considered. Firstly the correctness of the detection i.e. assigning of the bounding box needs to be evaluated by the metric called Intersection over Union (IoU). The IoU is a ratio between the intersection and the union of the predicted boxes and the ground truth boxes.

B. *Loss Function*

- YOLO predicts multiple bounding boxes in a cell. But to compute the loss for the true positive, only one of the bounding box is responsible for the object. For this reason, we select the highest value of IoU with the ground truth. YOLO then computes the sum-squared error between the predicted and ground truth to compute loss. This function sum of three losses:

a) Classification Loss: If an object is detected, the classification loss at each cell is the squared error of the class conditional probabilities of each class.

$$\sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2$$

where

$\mathbb{1}_i^{obj} = 1$ if an object appears in cell i, otherwise 0.

$\hat{p}_i(c)$ denotes the conditional class probability for class c in cell i.

b) Localization Loss: This loss measures the errors in the predicted boundary box locations and sizes. We only count the box responsible for detecting the object.

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2\right] + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2\right]$$

where

$\mathbb{1}_{ij}^{obj} = 1$ if the j th boundary box in cell i is responsible for detecting the object, otherwise 0.

λ_{coord} increase the weight for the loss in the boundary box coordinates.

c) Confidence Loss: If an object detected in the box, the confidence loss(measuring the objectness of the box).

$$\sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i\right)^2$$

where

$\hat{C}_i$ is the box confidence score of the box j in cell i.

$\mathbb{1}_{ij}^{obj} = 1$ if the j th boundary box in cell i is responsible for detecting the object, otherwise 0.

If the object is not detected in the box, then the confidence loss is

$$\lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i\right)^2$$

where

$\mathbb{1}_{ij}^{noobj}$ is the complement of $\mathbb{1}_{ij}^{obj}$.

$\hat{C}_i$ is the box confidence score of the box j in cell i.

λ_{noobj} weights down the loss when detecting background.

The sum of localization, confidence, and classification losses adds up to the total loss.

$$\begin{aligned}
&\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2\right] \\
&+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2\right] \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i\right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i\right)^2 \\
&+ \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2
\end{aligned}$$

IV. DATASET

Our implementation uses the Exclusively Dark (ExDark) dataset. The Exclusively Dark (ExDark) dataset is a collection of 7,363 low-light environments to twilight (i.e. 10 different conditions) with 12 object classes (similar to PASCAL VOC) annotated on both image class level and local object bounding boxes. This data set can be downloaded from https://github.com/cs-chan/Exclusively-Dark-ImageDataset/blob/master/Dataset

V. IMAGE ENHANCEMENT

Low-light is a challenging environment for image processing and computer vision tasks, either in contrast enhancement for better visibility and quality, or application-oriented tasks such as detection. Hence, to improve the quality of images we are employing three image enhancement techniques.

A. Histogram Equalization

Histogram Equalization is a computer image processing technique used to improve contrast in images. It accomplishes this by effectively spreading out the most frequent intensity values, i.e. stretching out the intensity range of the image. This method usually increases the global contrast of images when its usable data is represented by close contrast values. This allows for areas of lower local contrast to gain a higher contrast.

B. Dynamic Histogram Equalization

This dynamic histogram equalization (DHE) technique takes control over the effect of traditional HE so that it performs the enhancement of an image without making any loss of details in it. DHE partitions the image histogram based on local minima and assigns specific gray level ranges for each partition before equalizing them separately.

C. Image contrast Enhancement using Exposure Fusion Framework

This is an exposure fusion framework and an enhancement algorithm to provide an accurate contrast enhancement. Specifically, the weight matrix for image fusion using illumination estimation techniques is designed. Then, the camera response model is introduced to synthesize multi-exposure images. Next, the best exposure ratio is found so that the synthetic image is well-exposed in the regions where the original image under-exposed. Finally, the input image and the synthetic image are fused according to the weight matrix to obtain the enhancement result.

VI. COMPARISON BETWEEN TRAINING FROM SCRATCH AND TRANSFER LEARNING

Our goal was to build a better object detection model that would work in low light than the existing YOLOv3. Hence, we started training the YOLO model with the choice of hyperparameters used in the official YOLOv3 paper. The observed model started training at a large total loss of around 2000 which eventually decreases at a very slow rate. Then we loaded our model with the pre-trained weights of YOLO trained on coco dataset in the hidden layers and sampled the required class weights in the last layer. We have frozen the hidden layers and currently fine-tuning our last layers for all three scales in our machines. The model started learning from a total loss of nearly 50 and the total loss wasn't reducing after 90 steps. So, we are able to infer that features learned on COCO dataset really helped our model to get a good start but was not helpful in learning new about the dark dataset when only the last layers were fine-tuned. Loss curves have been plotted for monitoring the training process using Tensorboard.

VII. RESULTS

The accuracy of the YOLOv3 on the ExDark dataset was given by mAP value of 21.35%.

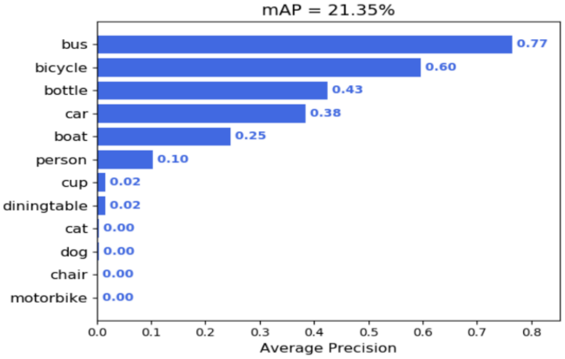


Figure 3: Average precision values per each class obtained using pre-trained YOLOv3 when tested on original ExDark data

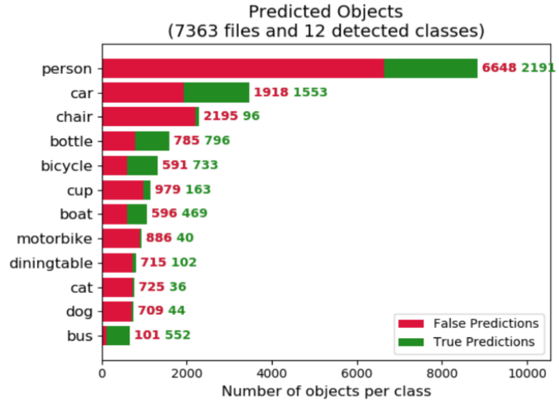


Figure 4: Number of true positives and false positives predicted for each class using a pre-trained YOLOv3 model for original data

The mAP score for the ExDark data preprocessed by histogram equalization image enhancement technique is 19.69%.

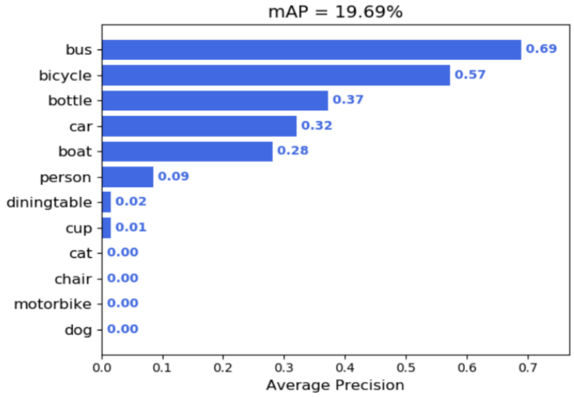


Figure 5: Average precision values per each class obtained using pre-trained YOLOv3 when tested on "he" preprocessed ExDark data

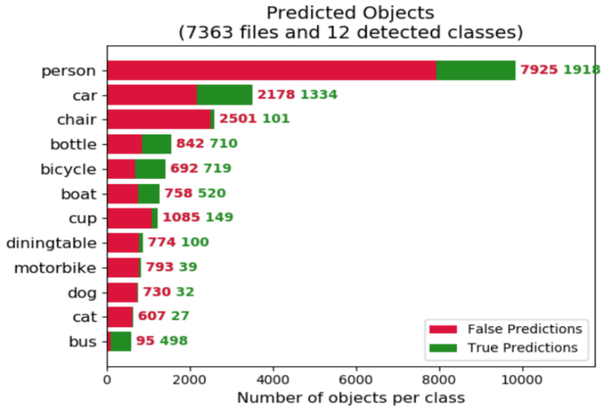


Figure 6: Number of true positives and false positives predicted for each class using a pre-trained YOLOv3 model using 'he' preprocessing technique

The mAP score for the ExDark data preprocessed by dynamic histogram equalization image enhancement technique is 20.97%.

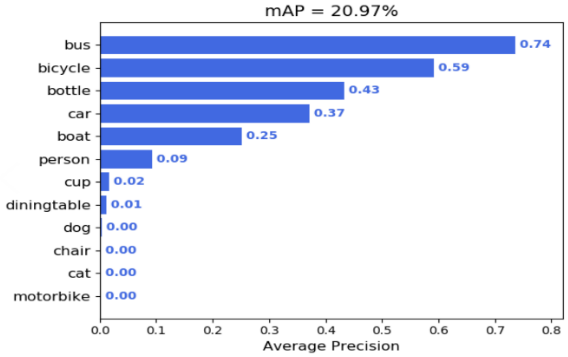


Figure 7: Average precision values per each class obtained using pre-trained YOLOv3 when tested using 'dhe' preprocessed ExDark data

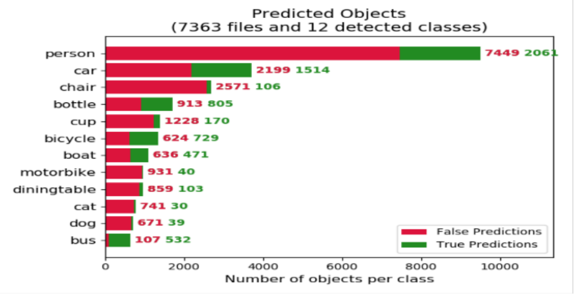


Figure 8: Number of true positives and false positives predicted for each class using a pre-trained YOLOv3 model on 'dhe' preprocessed data

The mAP score for the ExDark data preprocessed by exposure-based contrast enhancement technique is 20.49%.

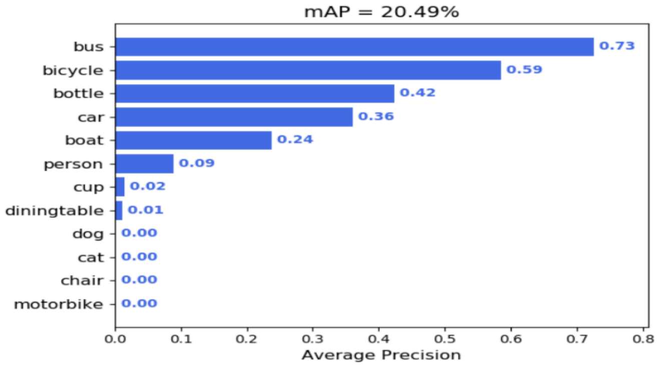


Figure 9: Average precision values per each class obtained using pre-trained YOLOv3 when tested using exposure-based enhancement on ExDark data

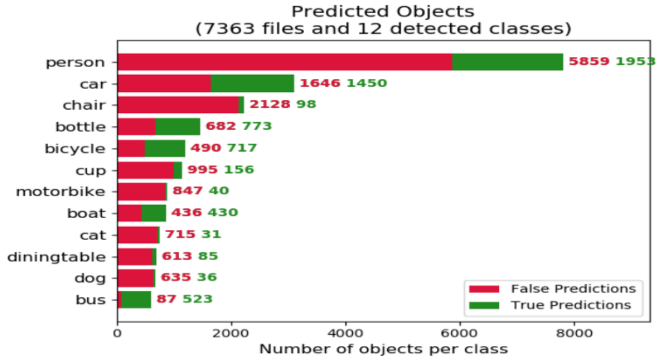


Figure 10: Number of true positives and false positives predicted for each class using a pre-trained YOLOv3 model using exposure-based contract enhancement technique

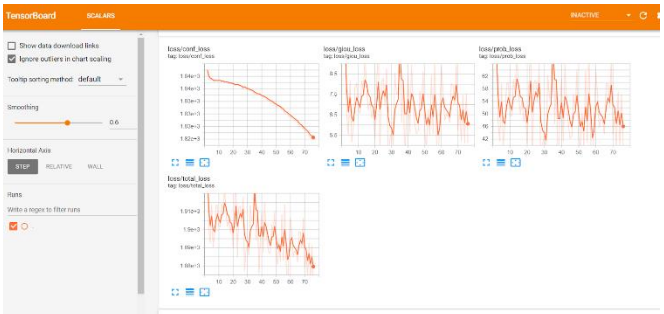
YOLOv3 mAP Comparison Table (Table 1)

Benchmark mAp on COCO dataset	57.9%
Benchmark mAp on ExDark dataset	21.35%
mAP with (he) enhancement technique on ExDark dataset	19.69%
mAP with (dhe) enhancement technique on ExDark dataset	20.97%
mAP with exposure fusion framework enhancement technique on ExDark dataset	20.49%

We have observed that the image enhancement techniques weren't very constructive in improving the performance of YOLO for object detection.

Model training with hyperparameters stated in YOLOv3 paper

We can see model learning from a loss of 2000 and slowly learning after a few mini-batches, so we stopped this approach to check if transfer learning helps our case (Figure 11).



☐ Show data download links
☑ Ignore outliers in chart scaling
Tooltip sorting method: default
Smoothing 0.6
Horizontal Axis: STEP

Figure 11: Model loss observed when training from scratch

Transfer learning (Fine-tuning) loss curves

The model started learning from a much lesser loss around 50 and is not learning anything new when we observed the loss curves. Highly oscillating loss curve seems to be an indication that our model may have hit the plateau problem and probably won't learn much than it already learned from COCO data even if we leave it for some more epochs (Figure 12).

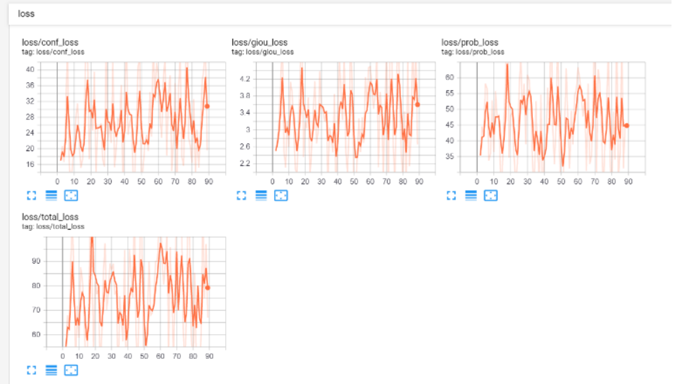


Figure 12: Model loss observed using transfer learning on the last layers

VIII. CONCLUSION

With YOLOv3 being one of the fastest and most powerful object detection systems, it still isn't efficient in detection of low light images. This can be observed from mAP comparison table (Table 1).

As presented by the paper '*YOLOv3: An incremental improvement*', the mAP score for YOLO trained on COCO dataset is 57.9%. We have considered this mAP score as our Benchmark mAP. But when trained on ExDark dataset it gave us a benchmark of 21.35% (Refer Figure 3). We opted for image contrast enhancement techniques as a data preprocessing method. The enhancement techniques used were Histogram Equalization (he), Dynamic Histogram Equalization (dhe) and exposure fusion framework. These enhancement techniques gave a mAP score of mAP(he) = 19.69% (Refer Figure 5) , mAP (dhe) = 20.97% (Refer Figure 7) and mAP(exposure) = 20.49% (Refer Figure 9) . Hence, we observe that the enhancement techniques didn't prove useful as the mAP score didn't improve when compared to the Benchmark mAP score of 21.35% on ExDark dataset.

To sum up, contrast-based image enhancement techniques applied to preprocess the given dataset, are adding more noise to the original images, deteriorating the results (Table 1), though they seem to yield better quality pictures when seen through the naked eye. In transfer learning approach, as seen in Figure 12, we observed that only fine-tuning last layers is not helping the model to understand more about low light dataset. Hence, in forthcoming steps, we plan to apply transfer learning on feature detection layers in YOLOv3 architecture i.e. Darknet-53 layers (Figure 2). This seems plausible as it might help our model to extract features in low light.

Also, we would like to explore other image enhancement and image denoising techniques, as our data preprocessing methods.

ACKNOWLEDGMENT

We are extremely thankful to Prof. Nik Bear Brown, Assistant Professor, Northeastern University for his valued guidance, inspiration and assistance while implementing the project.

REFERENCES

[1] https://github.com/YunYang1994/TensorFlow2.0-Examples/tree/master/4-Object_Detection/YOLOV3

[2] https://github.com/AndyHuang1995/Image-Contrast-Enhancement.

[3]]https://baidut.github.io/OpenCE/caip2017.html

[4] https://pjreddie.com/media/files/papers/YOLOv3.pdf

[5] https://pjreddie.com/media/files/papers/yolo_1.pdf

[6] @article{yolov3,
title={YOLOv3: An Incremental Improvement},
author={Redmon, Joseph and Farhadi, Ali},
journal = {arXiv},
year={2018}
}