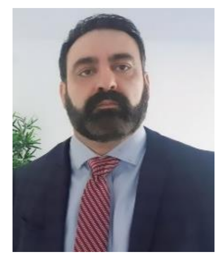
A Machine Vision Approach for PAPI Lights Detection and classification

Hamidreza Yazdi Zadeh Alborz
Robotics Engineer
University West
Trollhättan/ Sweden
Fax: +46735879910
E-mail: hamidrezaalborz@gmail.com

Mikael Ericsson
Avdelningschef
University West
Trollhättan/ Sweden
Fax: +46 520223318
E-mail: mikael.ericsson@hv.se

Mahmood Khabbazi
Universitetslektor, Biträdande
University West
Trollhättan/ Sweden
Fax: +46 0520223284
E-mail: mahmood.khabbazi@hv.se



ABSTRACT

The substitution of the human's vision ability with a machine, require to be mentioned that any ability of human uses prior knowledge to learn how can detect and classify the huge amount of information. Object detection and classification algorithm is an indispensable technology that is used for many purposes when interacting with the environment. In this way, the main problem is finding a robust algorithm to decrease time and computation for fast moving agents such as aircrafts and UAVs. However, finding the object in all parts of the image, is time consuming with increment on computation and the probability of false detection.

This paper describes a robust model to detect PAPI lights both in color and combination with a camera that installed on the aircraft as a sensor. The processing is done in real time without substantial memory requirements. The advantages of this work are: Decrease the workload of Pilots during inspection and improvement of decision making for PAPI lights color.

INTRODUCTION

The next generation of transportation system should be owed to enhancement of the automated technologies. Many companies are involved in the research area of this field. These companies involving this technology, to decrease the role of humans in control of the system as much as better the human factor. In this way, there are a lot of parameters that must be considered for optimized system behavior. Landing procedure during the flight is one of the important phases during a flight. In Visual Meteorological Conditions (VMC) condition, which pilots have sufficient visibility, Visual aid provides vertical and horizontal information to help a pilot to find and maintain the correct approach to a runway. Path Precision Approach Indicator (PAPI) lights are located generally on the left-hand side of the runway approximately 1000 feet from threshold and 50-150 feet from runway boundary [1]. Substitution of pilots' eyes as a reference point for detecting, tracking and classification of the PAPIs with the computer vision is the main object in this work. The automated system based on computer vision should recognize the position of the spotlights in the image and classify them as a red or white. Then they will be tracked frame by frame with the system processor and extract the same result for every frame. These results can be considered by those companies that research in the autopilot field and aviation calibration area.

As the previous experiment shows in high resolution image with small Region-Of-Interest (ROI) in pixel, the probability of false detection will be increased. One of the major solutions for this problem that has been used in this work is using camera geometry to find ROI based on the pose of PAPI. Assuming that the precise aircraft position has been estimated, a ROI location for the PAPIs can be calculated using the registered PAPI position and current aircraft pose. Extracting the ROI makes it possible to reduce the search region of Light [2].

The final goal of this project is to find a way to detect PAPIs with a camera in real time without substantial memory requirements. There are different deep learning methods that have been proposed to reach the goal. With mention to this point that the diameter of light starts to increase in images respect to decrement of distance. The advantages of this work are:

- Decrease the workload of Pilots during inspection.
- Data will be reused as references for postprocessing. After any calibration of PAPI lights we can process the data with any change in parameter without need to repeat the mission.
- Improvement of decision making. Aggregating of 30 or more weak learner (Frames) for decision will improve the calibration performance.

PROPOSED PAPI LIGHTS DETECTION

In this section, we explain the datasets which will be used to train and evaluate the Convolutional Neural Network (CNN)-based object detection models. The object detection and classification procedure can be described as follows:

- Determine the search region: A Region-Of-Interest (ROI) is extracted from the captured image by knowing the position of PAPI light. This extraction is calculated by camera geometry and coordination Transformation.

- Extract candidate objects: rectangular objects are extracted from the search region as candidate PAPIs. For finding the precise ROI in each image, the best algorithm that optimize comprising computation and time should be considered.

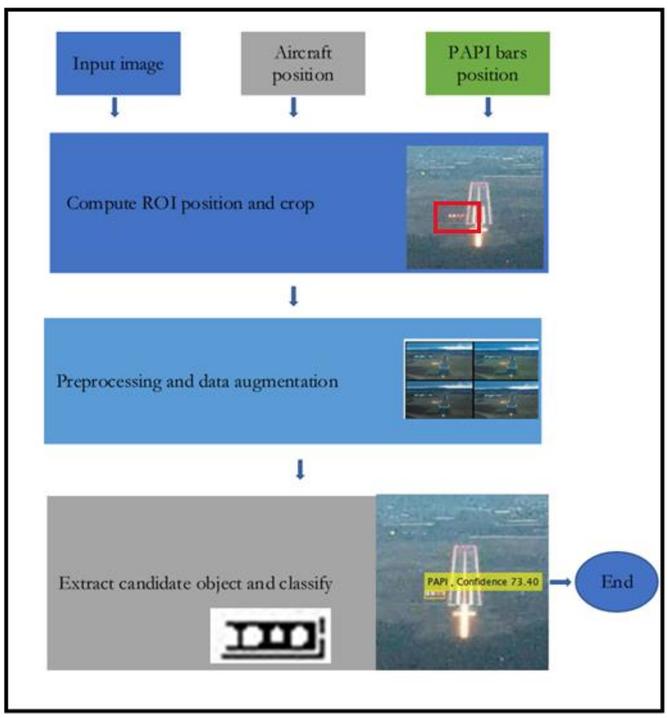


Figure 1. The Flowchart of the Proposed Method

- Classify the state of the candidates: identification, detection, and classification of PAPI colors should be done by simple color filtering or machine learning algorithms individually.

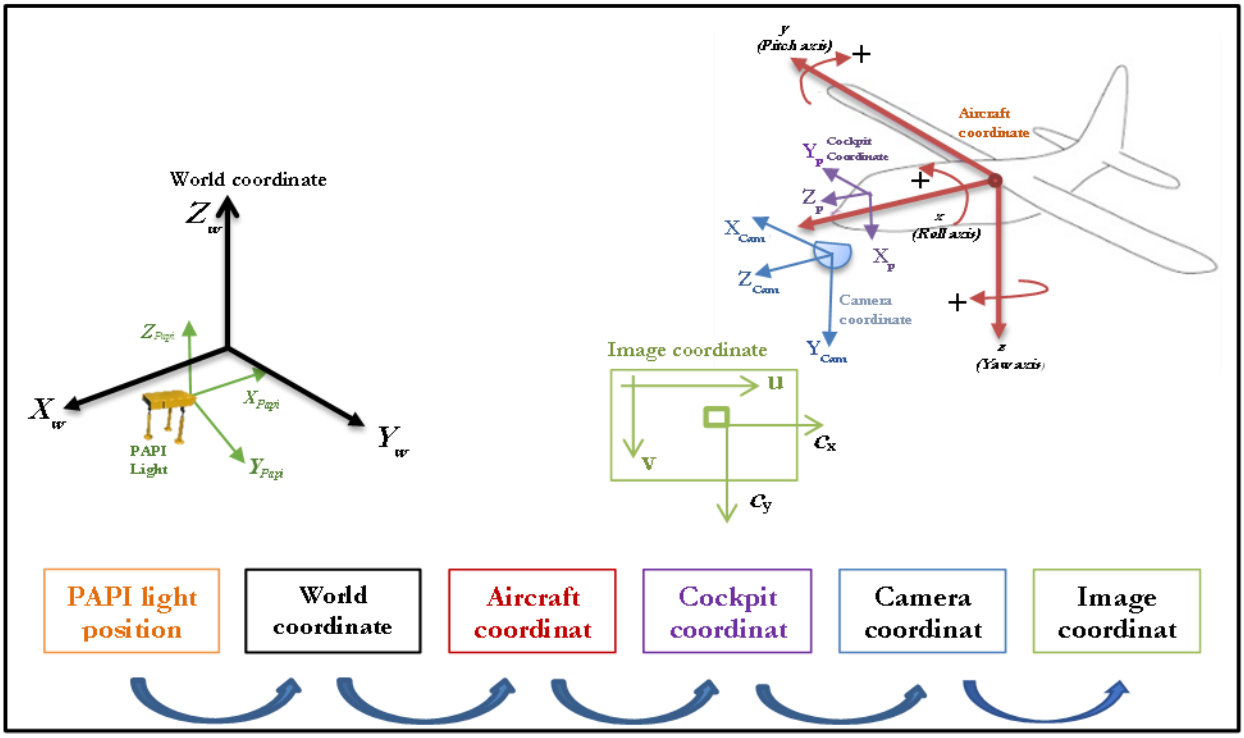


Figure 2. Coordinate System Considered in this Work.

The flowchart of the proposed method is shown in Figure 1.

<u>Lights detection with goal to calibration</u>

In this method an aircraft or flyable device simulate the real flight for calibrating the angle of each light. So, any light bar should be detected, separately and independently. Common procedure for light bars detection in calibration are Levelrun and sine like approach. In Levelrun procedure, flyable agent in proper altitude and distance far from light bars start to close to the light, then any transition of each light from white to red will be marked. Then, the angle adjusted for any light will be computed. Hence, in level run method we gather only one data (angle) per procedure but for entire light bars. In sine like approach procedure, we increase redundancy to reach better accuracy. In this procedure, the agent alternately crosses the angular conceptual line outside of any light and every time computes the angle of crossing point. Here, we have more data, but in one procedure we can check only one light bar.

<u>ROI Estimation</u>

With ROI cropping, the area for searching the proposed region will be decreased. The algorithm of detection will run faster and false positive and negative will be decreased. By knowing the real diameter of the PAPI light and calculating the distance between aircraft and light bars, we can have guesses of diameter of expected blob, simultaneously, eliminate unwanted detected blob. Each image cropping with a rectangular shape when the position of PAPI bars in image is detected as a ROI. The location of the ROI can be calculated based on the current pose and the position of PAPIs. Figure 2, illustrates the coordinate systems considered in this work. The PAPI position in the world coordinate $\boldsymbol{p}_w = [x_{papi}, y_{papi}, z_{papi}, 1]$ is converted to $\boldsymbol{p}_{Aircraft} = [x, y, z, 1]$.

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} X_{papi} \\ Y_{papi} \\ Z_{papi} \\ 1 \end{pmatrix}$$

Where, $\boldsymbol{R}$ is a 3×3 matrix as rotation can be calculate of Euler angles, $\boldsymbol{t}$ is translation vector which is coordination of aircraft provided by GPS or in other words, the position of aircraft in world coordinate. A pixel position u, v is then calculated based

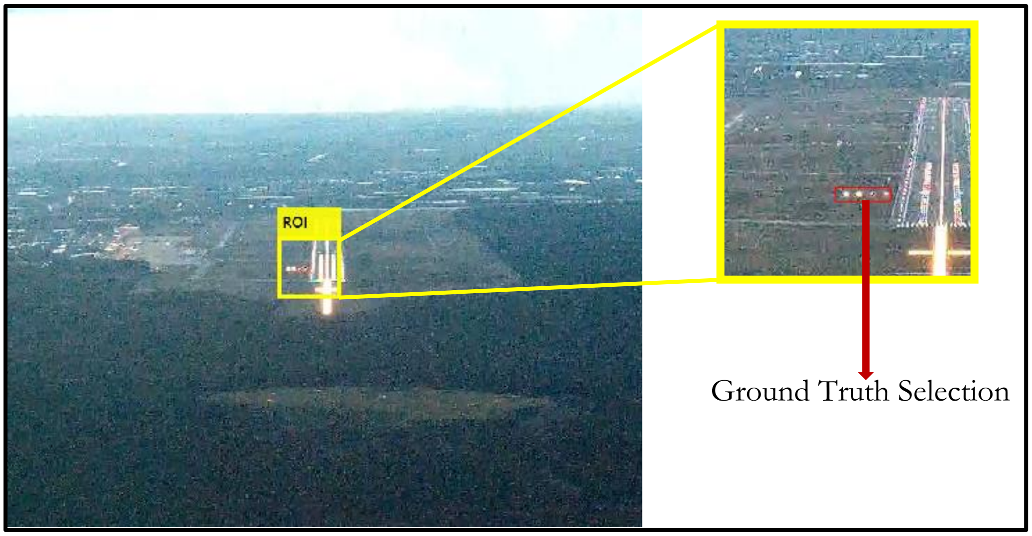


Figure 3. The ROI is Estimated by knowing the Position of PAPI and Aircraft. Ground Truth for First Level of Detection.

on the intrinsic parameters of camera [3]. Regarding to finding the *u* and *v*, we can calculate the ROI as a rectangle with a width w_{ROI} and a height h_{ROI} that their CenterPoint is pixel u and v.

$$w_{ROI} = k_{ROI}\frac{f_x s_s}{z_s}$$

$$h_{ROI} = k_{ROI}\frac{f_y s_s}{z_s}$$

Where f_x, f_y are camera calibration parameter or intrinsic parameters of camera $and\ k_{ROI}$ is constant scale parameter for determining the size of ROI and s_s is a size of PAPI [4]. The precise aircraft pose is known for each image by stamping the time to both image and position data, and the apparent motion of objects due to changes in roll, pitch, and yaw are straightforward to correct using the camera model. The algorithm for estimating the ROI in an image is,

- Convert all position to UTM coordinate Finding the matrix for converting the world coordinate to aircraft coordinate [5]. For finding this homogeneous matrix with 6 DOF, we should find the position of aircraft with GPS sensor and roll, pitch, and yaw angle with Gyro sensor.
- Finding a translation matrix with knowing offset between Gyro position and cockpit position.
- With another translation parameters, the world coordinate maps to camera coordinate.
- The calibration parameters or intrinsic parameters of camera, following by a translation matrix, can transform the camera to pixel coordinate.
- With knowing real distance between camera and PAPI position we can estimate the ROI parameters for clipping the image due to have a constrains on searching area. Figure 3 shows the ROI when is clipped based on camera geometry calculation.

<u>First level of detection and evaluation</u>

After cropping the images using camera geometry, first step of detection is implemented. In this level the ground truth is bounding box of area of all 4 lights in the image. Figure 3 shows the ground truth selection in first step of detection. After augmentation data, and estimate the anchor boxes in size and number, using a YOLO-V3 detector the area of PAPI lights is detected and cropped [6]. In this level with estimating the 6 anchor boxes, we achieve a 99% in area detection. Figure 4 is a plot of precision-recall with average precision in title. For evaluation of a model in the field of machine learning, Precision means the percentage of model predictions that are relevant, but recall refers to the percentage of total predictions that are correctly categorized by the model. Precision is equal to dividing the number of items detected by the model correctly by the

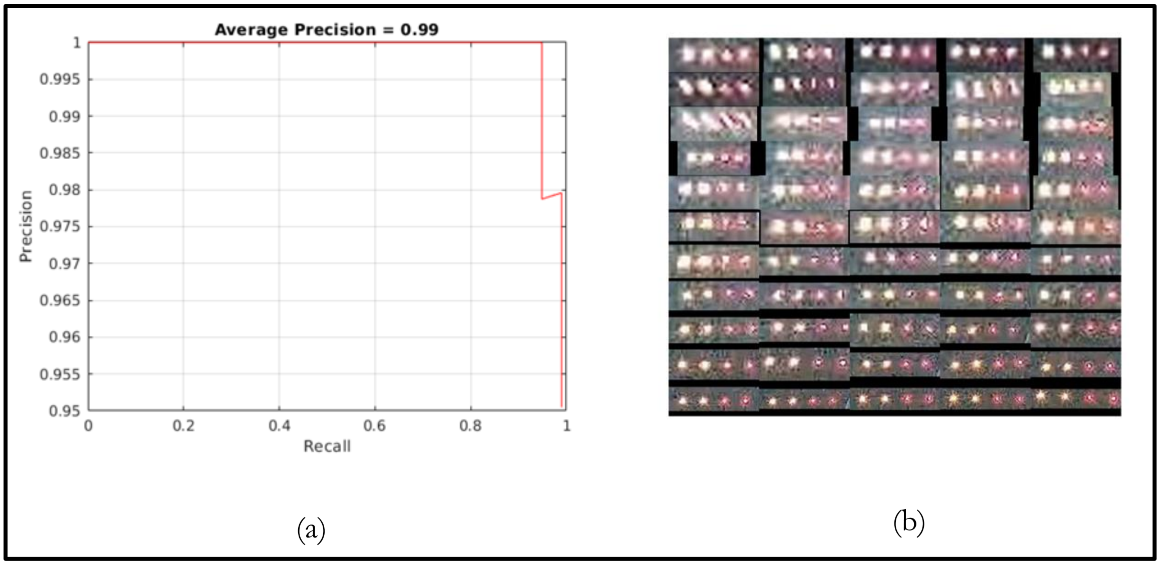


Figure 4. The Results of First Level of Detection. The Recall vs Precision Graph (a). The Example Images has been Cropped from ROI that are used for Next Level of Detection (b).

total number of items created by the model, and Recall is equal to dividing the number of items detected by the model correctly by the number of items detected correctly.

$$Precision = \frac{True\ Positive}{True\ positive + False\ Positive}$$

$$\boldsymbol{Recall = \frac{True\ Positive}{True\ positive + False\ Negative}}$$

Precision is a good tool for examine a model that the costs of False Positive is high. The mean average precision (mAP) is a single number that help detector to make correct classifications (precision) and finding all relevant objects (recall). The precision-recall curve demonstrates how precise a detector is at varying levels of recall. Ideal PR graph is happened when the precision is 1 at all recall levels. Another criterion for evaluate a network is F-Value. F-Value measurement is the harmonic mean of recall and precision. This is a parameter to search a balance between Precision and Recall. The goal of apply this parameter is focus on both False Negative and False Positive. F-Value is calculated based on this formula,

$$F - value = 2 \times \frac{Recall \times Precision}{Recall + Precision}$$

Intersection over Union (IoU) measures the overlap between 2 boundaries, ground truth and predicted boundary. It defines with this formula,

$$IoU = \frac{Area\ of overlap}{Area\ of\ union}$$

Sometimes, the detector will predict many boundaries for one object in image. For solving this problem, in some datasets, we can predefine an IoU threshold in classifying whether the prediction is a true positive or a false positive.

<u>Second level of detection</u>

We implemented YOLO_V3 based on MATLAB for object detection. The main reason of selection of this network is that YOLO_V3 by adding detection at multiple scale can perform better than YOLO_V2 in smaller object detection [7]. Both accuracy and training time are the main part of a network's performance. A tradeoff between decreasing training time and maintaining a high accuracy. By adjusting the hyperparameters of network we can optimize the network's performance or

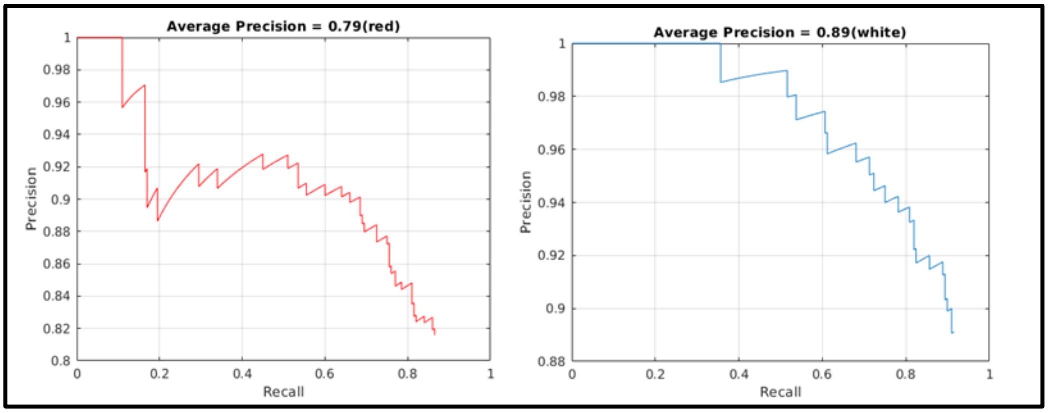


Figure 5. The Graph Precision-Recall (PR) for both Red and White Class. Ideally, Optimum PR Graph is happened when the Precision is 1 at all recall levels. The Text in Title shows mAP.

prevent to overfitting. The learning process will be ended when the training parameter reached to any constrains based on options for training. The parameters adjustment was done in this way:

- Network: The YOLO_V3 network uses SqueezeNet with the addition of two detection heads at the end. The input size of SqueezeNet as a feature extractor net is [227 227], so we scale our images to this size. For better detection on smaller object the second detection head is twice the size of first detection head [8].

The learning rate: this parameter depends on training result observation. During training process time If the learning rate is too high, maybe the results do not in common with ratio index 0.1. If the learning rate is too low, then the convergence of optimization loss function takes long time. Learning rate for this work is 0.001. The number of iterations set to 2000. The number of iterations that the learning rate exponentially increased during these iterations is called Warm up period. This parameter helps the network during the learning, to stabilize the gradients at higher learning rates. The formula for calculating this parameter is,

$$LearningRate = \left(\frac{iteration}{warmup\ period}\right)$$

- Number of epochs: The full pass of the training algorithm over the entire training dataset defined as an epoch. Mini-batch size is positive integer that indicates maximum number of epochs to use for training. An iteration in training process is one step that in the gradient descent algorithm or other optimization algorithm is taken for the minimizing the loss function using a Mini batch. Selecting the optimal mini batch size, help to the network to reach to optimal position in decrement of loss function. When we select the mini batch size and see the loss function at the end of training process still decreasing so should increase the mini batch size.
- Solver: The optimizer has been used in this work is the stochastic gradient descent with momentum (SGDM) [9].

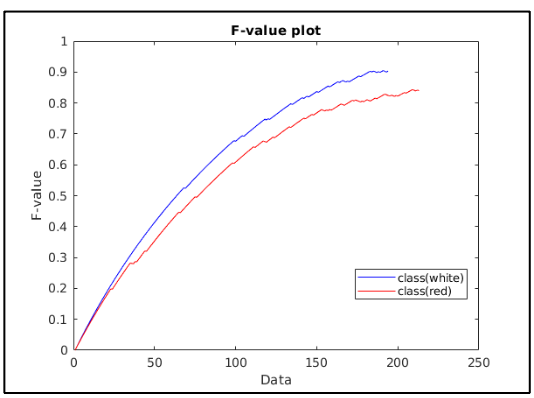


Figure 6 This figure shows F-Value curve for all data in the test data set for both class red and white.

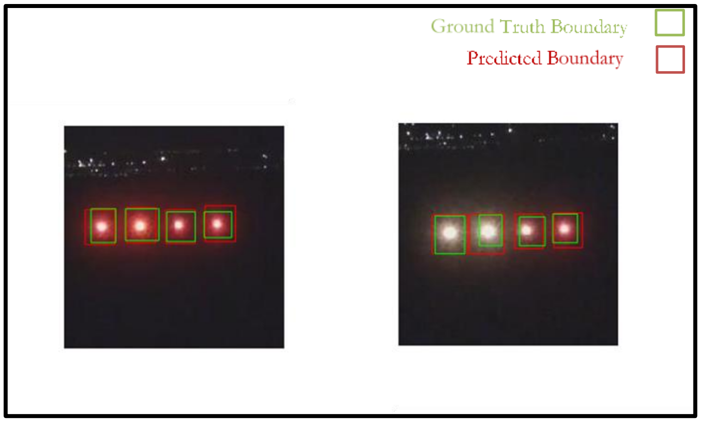
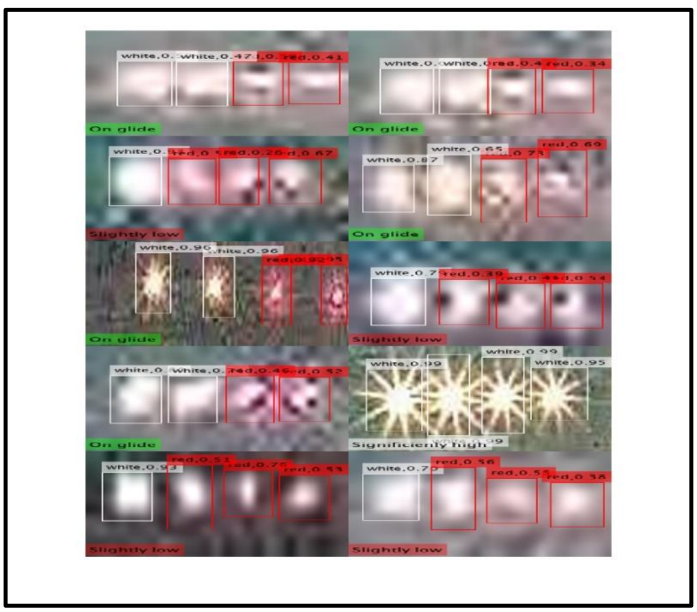


Figure 7. This Picture with Rectangles shows the IoU in two Test Images. The green Rectangles as Ground Truth Boundary and Red Rectangles as Detected Boundaries (night run).

- Train on GPU: GPU based processing is highly recommended for training the Deep Learning Neural Network (DNN), particularly when data set has huge number of images with high resolution [10]. Because of big amount of information during the training of network, with parallel computing toolbox and CUDA enable NVIDIA GPU the training time significantly decreased, otherwise based on CPU environment with big data set, in practice impossible to train the network. The computer used for the evaluation was on ubuntu 20.04 operating system, the CPU was core i5-8300H, the memory was 8 GB, and the GPU was NVIDIA GeForce GTX 1050i. The processing on the CPU was operated in a single thread.



- **Figure 8. A collection of Image Outputs shows the Object Detection and Classification Performance of the Model.**

With monitoring the total loss vs iteration graph, we can determine those other hyper parameters such as mini-batch size, number of iterations, loss function solver and other factors, are selected in proper way or not. In Figure 7 Intersection over Union (IoU) for two test images has been shown. IoU measures the overlap between 2 boundaries, ground truth and predicted boundary. In this work, this threshold set to 0.5. For evaluation of performance of model, output image has been defined same as Figure 8. In this figure, an image from dataset is selected, then the behavior of model is visualized. In each picture the boundary box has been detected. Depend on class, white or red, the color of bounding boxes is changed. Then for each boundary box, the class and confidence score has been labeled to it. Finally, the array of light has been classified to significantly low, slightly low, on glide, slightly high and significantly high, then labeling to the images.

CONCLUSIONS

In this work, it has proposed a PAPI light recognition algorithm for inspection and maintenance purposes. In prepared dataset that includes objects with large and small pixel sizes of objects for PAPI lights provided in day and nighttime. The YOLO-V3 method can be processed in real time by the CPU, and this work verified that this method could recognize PAPI lights with high confidence and precision. The evaluations determine that with this algorithm achieve to mAP 79% in red class and 89% in white class. We solved the problem of finding the optimum ROI to enhance the CNN network performance, based on camera geometry. With knowledge of PAPI light position added by camera geometry solution for coordinate transformation, successfully be achieved to find limited ROI that significantly improve the network performance. With this method the small size of objects with a few pixels can be recognized. Moreover, camera geometry for limiting the ROI, in the environment that there are many similar lights, are intelligent solution to decrease false detection. we observed that YOLO is a fast, good for real-time processing. In [6] would be found graph and table for YOLO-V3 performance vs another object detector. Although we observed many cases that it is relatively difficult to determine the lighting color of the lamp due to the influence of the surrounding brightness, the model clearly could detect the object in these images. In addition, we observed such other objects can impact performance of network exclusive of object detector, like, feature extractors selection, localization loss function. input image resolutions, IoU threshold (how predictions are excluded in calculating loss), the number of proposals or predictions, data augmentation, training dataset, use of multi-scale images in training or testing (with cropping), deep learning software platform used and training configurations including batch size, input image resize, learning rate, and learning rate decay.

FUTURE WORK

- Many objects can have effect on performance of network. For optimizing the results, these future works are recommended:
- Other algorithm would be applied with comparing the results.
- Another machine learning algorithm for improving performance should be examined.
- Real time investigation for practical purposes and implementation would be investigated.
- In different weather the model should be examined.

ACKNOWLEDGMENTS

we would like to thank Aerodata for helping us with this project by providing data and guidance. We would also like to thank University West for giving us this opportunity to work on this project.

REFERENCES

[1] "Annex 14 to convention on International Civil Aviation," in *Volume I,"Aerodromes"*, International Civil Aviation, July 2018, pp. 5-52.

[2] V. John, K. Yoneda, B. Qi, Z. Liu and S. Mita, "Traffic light recognition in varying illumination using deep learning and saliency map," *2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014,* pp. 2286-2291, 2014.

[3] K. Yoneda, A. Kuramoto, N. Suganuma, T. Asaka, M. Aldibaja and R. Yanase, "Robust Traffic Light and Arrow Detection Using Digital Map with Spatial Prior Information for Automated Driving," *Sensors,* vol. 20, no. 4, p. 1181, 21 2 2020.

[4] D. C. Brown, "Decentering Distortion of Lenses," *Photometric Engineering,* vol. 3, no. 32, pp. 444-462, 1966.

[5] J. P. Snyder, Map projections--A working manual, vol. 1395, US Government Printing Office, 1987.

[6] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[7] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767,* 2018.

[8] F. N. a. H. S. a. M. M. W. a. A. K. a. D. W. J. a. K. K. Iandola, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," in *ICLR 2017,* 2016.

[9] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," 22 12 2012.

[10] I. Goodfellow, Y. Bengio and A. Courville, Deep learning, 2016.