A DEEP LEARNING APPROACH FOR REAL-TIME DETECTION OF MOTORCYCLISTS WITHOUT HELMETS USING CONVOLUTIONAL NEURAL NETWORK AND YOLOv2

Mary Ann E Telen[1], Aljoe Espinosa[2]
[1,2] University of Science and Technology of Southern Philippines, C.M. Recto Ave., Lapasan, Cagayan de Oro City, Philippines
*For Correspondence; Tel. +639265918458, Email: maryann.telen@ustp.edu.ph

ABSTRACT - *Motorcycle transportation is gaining popularity in many countries due to various social and economic factors. However, safety protocols set by the government require the use of helmets, which is often disregarded by riders. Accidents can be fatal if the rider is not wearing a helmet. To ensure the safety of riders on the road, an automated system is necessary for the detection of helmeted and non-helmeted motorcyclists. In this study, a convolutional neural network (CNN) was used for the automatic detection of helmet and non-helmet motorcyclists. Deep learning-based models have shown promising performance in object detection, and YOLOv2 is a popular model that combines classification and object detection in a single architecture. The main objective of this study was to achieve real-time detection of motorcyclists without helmets. The system operates by processing post-video motorcycle and helmet detection using YOLOv2 algorithm and Dark flow for implementation. The accuracy percentage of the algorithm was evaluated and compared with the standard detection of YOLOv2 on VOC2007. All extracted data from the system were stored in a Google Drive account for cloud storage. The results showed that the proposed algorithm was effective for helmet detection, achieving an accuracy percentage of 83%. The detection results were evaluated and verified. This study contributes to the development of an automated system for motorcycle safety and can be used as a basis for further research.*

Keywords: — Motorcycle transportation, Helmet detection, Convolutional neural network, YOLOv2 algorithm, Deep learning-based models

INTRODUCTION

Motorcycles have become one of the most popular modes of transport, resulting in a significant increase in their use and a corresponding rise in the number of road accidents. While motorcycles provide riders with mobility and flexibility on the road, riding without a helmet poses a serious danger. Motorcyclists lack the structural support that cars provide to keep drivers safe, making it crucial for riders to take extra care to protect their bodies. To address this issue, governments have made it mandatory for motorcycle riders to wear standard protective helmets while driving. Wearing a helmet significantly increases a rider's chances of survival in the event of an accident. However, current video surveillance-based techniques for detecting non-helmet-wearing riders require significant human involvement and are passive in nature. Consequently, governments have resorted to manual strategies to catch violators. [1] According to the study of Zheng, they adopted a deep convolutional neural network (CNN) and attained a high degree of accuracy in detecting helmets, reaching 90% accuracy on one of the datasets. This finding supports the effectiveness of the proposed model for motorcycle helmet detection [2]. According to the study of Xu, they employed a convolutional neural network (CNN) and an advanced version of the YOLO algorithm for helmet detection. Their system yielded an accuracy of 94.3% on a motorcycle helmet dataset, indicating the efficacy of deep learning in motorcycle safety through helmet detection. [3]

The objective of this study is to develop an automated image processing system capable of detecting the number of motorcyclists riding without helmets in a specific road area. The system is designed to operate in two steps. The first step involves detecting two-wheel vehicles, whether they are motorcycles or not. The second step involves detecting motorcyclists without helmets, which is then captured by the camera and sent to the cloud. The use of this system will assist the Land Transportation Office (LTO) in accurately counting the number of motorcyclists violating the helmet law. The system's automation is highly advantageous because it allows for more precise and robust monitoring of these violations while significantly reducing the amount of human resources required.

METHODOLOGY

A. Conceptual Framework

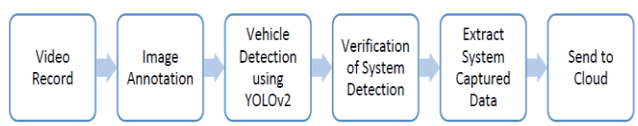


Figure 1. Conceptual Framework of the System

The conceptual framework of the system shown in Figure 1 was set in into six process, (1) Video Record, (2) Image Annotation, (3) Vehicle Detection using YOLOv2, (4) Verification of System Detection, (5) Extract System Captured Data and (6) Send to Cloud. The video recording was recorded and converted into images in order to annotate them for training purposes. The vehicle detection using YOLOv2 is used in the testing of video data. Bounding boxes was set into each classified model with their classification. Verifying that the actual classified model inside the bounding boxes with classification will be in high confidence level. Extracting the captured images from the system with the classification required and then the captured images are sent to cloud.

B. Research Setup

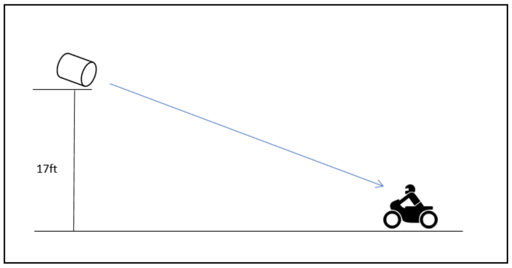


Figure 2. Setup of the Data Gathering

The camera is mounted as setup in Figure 2 and continuously tracks the flow of a moving motorcycle vehicle over a set period of time. The video clip is a post-process object detection that has gone through multiple steps after training to assess if the vehicle is a motorcycle or not and detecting whether or not a motorcyclist is wearing a helmet.

C. Research Variables

For the independent variables, an iSight camera (iPhone camera) capable of 1080p at 60fps was to be installed on the overpass where the motorcycle vehicle would pass through. The camera's angle should be sufficient to oversee the motorcyclist since it affects the classification of the motorcycle and helmets. For the dependent variable, in detection, the distance between the motorcycle determines the accuracy of detection. Motorcycle and helmet detection are dependent on environmental conditions and the velocity of the motorcycle.

D. Research Setting

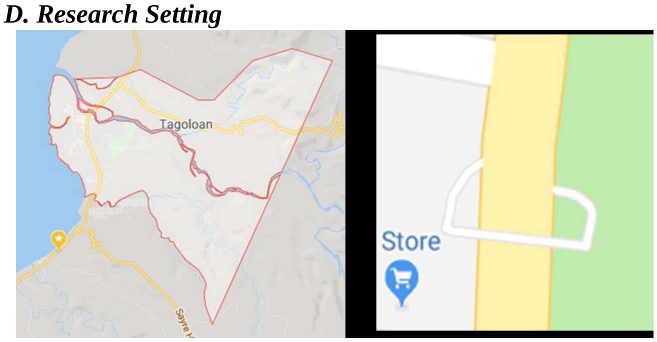


Figure 3. Research Setting

The tests were carried out at the Tagoloan, Misamis Oriental overpass shown in Figure 3. The location was chosen primarily for the ease of access and protection of the hardware used to collect data. The study focused on collecting data at various times of day, from morning to afternoon, where there are many motorcycles and motorcyclists without helmets can be seen. This study focuses on the motorcyclist to test the level of accuracy of the system. The first module shown in Figure 4, the researchers trained the images and videos, tagging the images to train frame by frame for a custom dataset to classify, then it will feed it to the Darkflow (YOLOv2 for Windows). Darkflow applies a single neural network on the image, then running linear regression to create class probabilities and bounding boxes, and finally using Non-Maximum Suppression (NMS), a technique to filter the predictions of object detectors, to eliminate the points that do not lie in important edges of the images [4]. The system then detects the train weights or custom datasets that were trained which annotates the qualified motorcyclist and helmet. In the second module, the system will detect each moving motorcyclist that is captured frame by frame. If the system does not detect a motorcyclist, it will continue to scan until it does then the system will forward it to the third module, for helmet detection. In the third module, the program will search for the helmet of the motorcyclist and will based its detection on the custom datasets provided by the researchers throughout the training process. If there is a helmet of the motorcyclist then discard the bounding box of the helmet motorcyclist and if there is none, then the program will pass it to the next module. On the fourth module, the program would remove the bounded box of the non-helmet motorcyclist in the system's program and save it to the system's cloud backup folder before sending it to the cloud.

E. System Design

.

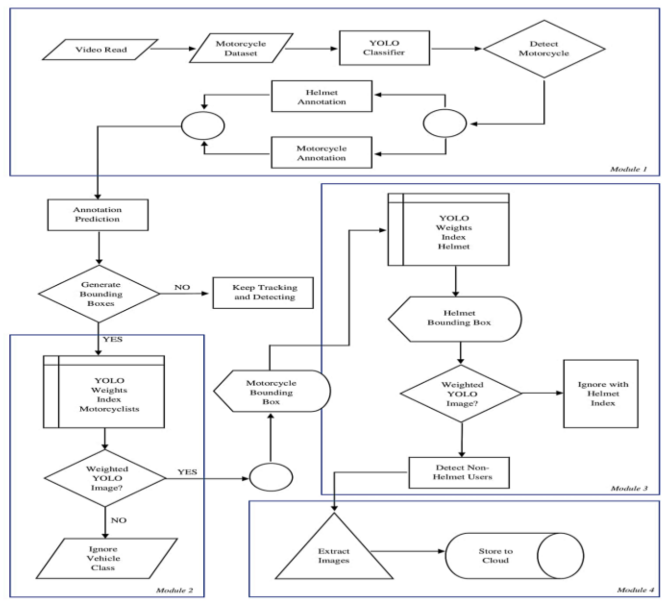


Figure 4. System Flowchart

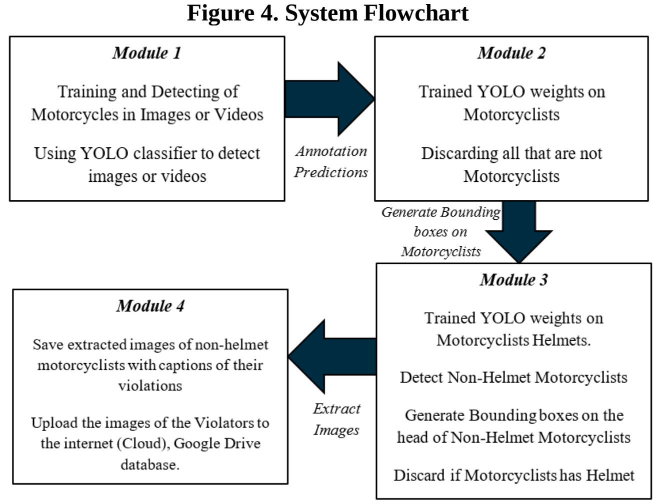


Figure 5. System Module Flow

To define the four modules in the Figure 5 scheme. The first module is the stage where video acquisition begins and the tagging, detecting, and tracking of motorcycles is initiated. It will create bounding boxes based on the coordinates of the motorcycle and helmet that correspond to the annotated region of the image, ensuring the confidence level of the annotated region. The second module is where classifying the image using weights to determine whether or not it is a motorcycle, and discarding negative images of a motorcycle. The third module is to detect if it is a motorcycle, it will then extract a picture of the motorcyclist with their motorcycle and it will save the image to the database, local storage. The fourth module determines that the motorcyclist is not wearing a helmet, if so, it will extract a picture of the motorcyclist with their motorcycle and save it to the database, connected to the cloud.

F. Instrumentation and Modelling

The video of the scene was captured using a camera module. In terms of software, YOLO [5] was designed using the Tensorflow systems algorithm. The You Only Look Once (YOLO) system is a cutting-edge, real-time object detection system. On COCO test-dev, it handles images at 67 frames per second and has a mAP of 76.8 percent on a Pascal Titan X. YOLOv2 is a very fast and accurate program. In mAP at.5 IOU, YOLOv3 is comparable to Focal Loss but around 4x faster. It applies the model to a picture at various sizes and locations. The researcher uses a single neural network to process the entire image. For one-time object detection, YOLO with Darkflow, an open-source implementation, was used. Python3 was used as a programming language. [6]

G. Research Dataset Training

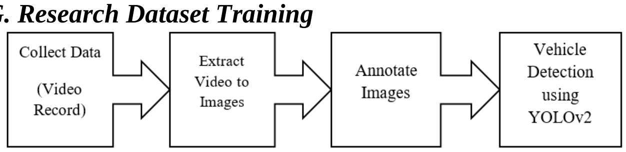


Figure 6. Training Process Flow

The researchers developed a training dataset method for motorcycle and helmet detection, as well as data extraction to the cloud.

he training phase flow is depicted in Figure 6 above. The Training Process Flow will start at the training of the datasets first, by training the datasets in various degrees and angles of the camera will determine the likelihood of the bounding boxes to have a higher confidence level.

1. *video recording:* the video record used as the custom dataset for training;
2. *extract video to image:* the collected video will be extracted to images to manually annotate the image;
3. *image annotation:* the setting of the images in annotating it what it is, also used to train data and to have XML files in order to have a classification of the data.

vehicle detection using YOLOv2: the testing of the data from the video and bounding boxes are set into each classified model with their classification. To annotate files, *labelImg* was used as a software tool. Annotations are stored as XML files in the PASCAL VOC format, which is compatible with the YOLO format. Automatic image annotation is the mechanism by which a computer system automatically assigns metadata to a digital image in the form of captioning or keywords.

H. Collection of Custom Datasets

The system was trained using a custom dataset in the analysis. The researchers collected videos at the Tagoloan, Misamis Oriental overpass in three separate time settings. Figure 7 below depicts a sample custom dataset gathered from various time settings. The recorded videos were converted into images at 2 frames per second for image tagging and annotation. The photographs collected included every angle of a motorcycle passing through the overpass, as well as the motorcyclist, whether he or she was wearing a helmet or not.
4.

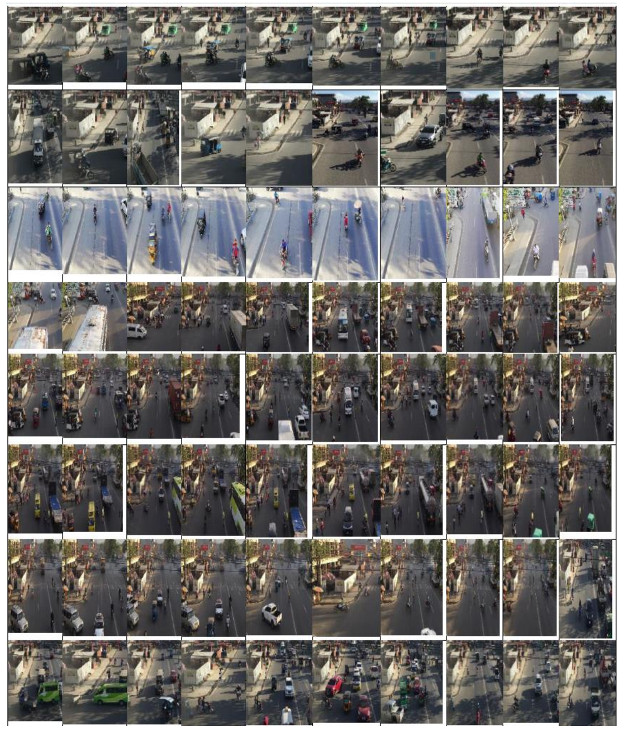


Figure 7. Sample Extracted Images

Figure 8. Sample of Image Annotation

Figure 8 depicts a sample of motorcycle and helmet annotation training data for the dataset. For image annotation, the researchers used *labelImg*. A directory containing the photos to be labelled was opened. *RectBox* was designed to enable you to pick a region to annotate. Tagging images must be compatible with what the mark to be annotated should be. The label list must not be changed when processing a list of images. When an image is saved, classes.txt is updated, but previous annotations are not.

I. Object Detection Algorithm using YOLOv2

The study used YOLO, or You Just Look Once, for an object detection algorithm that was very different from the region-based algorithms [7]. In YOLO, a single convolutional network predicts the bounding boxes as well as their class probabilities. It splits an image into a SxS grid and then takes bounding boxes within each grid. The network outputs a class likelihood and offset values for each of the bounding boxes. The bounding boxes with a class probability greater than a certain threshold value are chosen and used to locate the object within the image.

Figure 9. YOLO detection System

Figure 9 shows how simple it is to process images with YOLO. Our method (1) resizes the input image to 448 x 448 pixels, (2) applies a single convolutional network to the image, and (3) weights the detected objects based on the model's confidence. (Redmon, Divvala, Girshick & Farhadi, 2016)

The researchers started training with PASCAL VOC results. The researchers used *predict.py*, which is responsible for predictions and drawing bounding boxes on annotated images. *helmet.py* is a Python script that automates image tection and transfer to a cloud folder. The researchers changed the *cfg file* under region layer to the four classes that have been trained.

[region]
anchors = 1.08,1.19, 3.42,4.41, 6.63,11.38, 9.42,5.11, 16.62,10.52 //reference points
bias_match=1
classes=4 // bicycle; motorcycle; helmet; withouthelmet coords=4
num=5
softmax=1

After that, Change filters in the [convolutional] layer (the second to last layer) to num * (classes + 5). In this case, num is 5 and classes are 4 so 5 * (4 + 5) = 45 therefore filters are set to 45.

[convolutional]
size=1
stride=1
pad=1
filters=45
activation=linear //YOLOv2 used linear regression.

Change labels.txt inside Darkflow for the classes name.

Motorcycle
bicycle
with helmet
without helmet

The training started using the command:

python flow --model cfg\tiny-yolo-voc-3c.cfg --load -1 --dataset C:\KMPlayer\Capture\IMG_0585 --annotation C:\KMPlayer\Capture\IMG_0585 --train --gpu 0.7.

After building the model during the training, Figure 10 shows the sample training loss process. The system run in an initial training epoch of 24, and has a checkpoint at step 3798. The Researchers used NVIDIA GEFORCE 940MX GPU, and it took about an hour to finish training process. An average loss error should be as low as possible to stop completely the training.

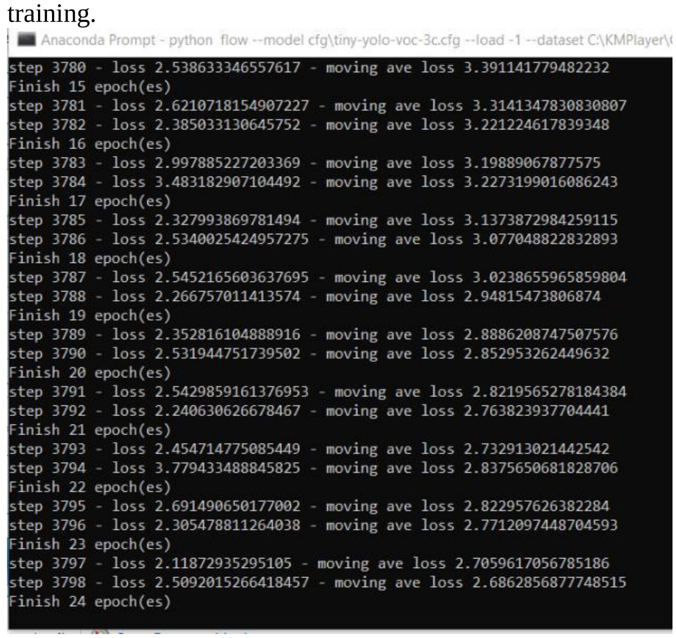


Figure 10: Training Loss

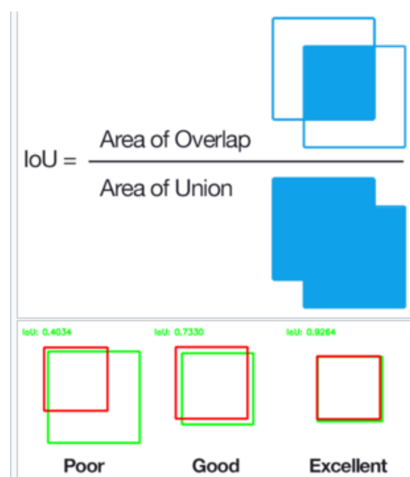
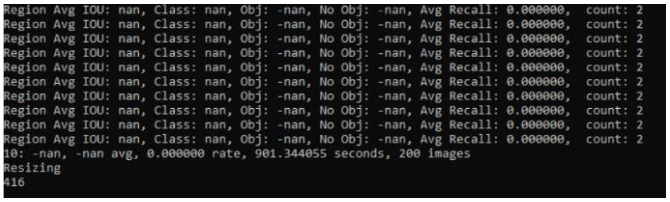


Figure 11: YOLO Subdivision Output

Figure 11 above shows IOU (Intersection-Over-Union) a great metric to determine how accurately the model detected a certain object. At 100% it has a perfect detection: a perfect overlap of our bounding box and the target. It's clear that it needs to optimize this parameter. [8]



```
Region Avg IOU: nan, Class: nan, Obj: -nan, No Obj: -nan, Avg Recall: 0.000000,  count: 2
Region Avg IOU: nan, Class: nan, Obj: -nan, No Obj: -nan, Avg Recall: 0.000000,  count: 2
Region Avg IOU: nan, Class: nan, Obj: -nan, No Obj: -nan, Avg Recall: 0.000000,  count: 2
Region Avg IOU: nan, Class: nan, Obj: -nan, No Obj: -nan, Avg Recall: 0.000000,  count: 2
Region Avg IOU: nan, Class: nan, Obj: -nan, No Obj: -nan, Avg Recall: 0.000000,  count: 2
Region Avg IOU: nan, Class: nan, Obj: -nan, No Obj: -nan, Avg Recall: 0.000000,  count: 2
Region Avg IOU: nan, Class: nan, Obj: -nan, No Obj: -nan, Avg Recall: 0.000000,  count: 2
Region Avg IOU: nan, Class: nan, Obj: -nan, No Obj: -nan, Avg Recall: 0.000000,  count: 2
Region Avg IOU: nan, Class: nan, Obj: -nan, No Obj: -nan, Avg Recall: 0.000000,  count: 2
Region Avg IOU: nan, Class: nan, Obj: -nan, No Obj: -nan, Avg Recall: 0.000000,  count: 2
10: -nan, -nan avg, 0.000000 rate, 901.344055 seconds, 200 images
Resizing
416
```

Figure 12: YOLO Batch Output

Figure 12 above, shows the YOLO batch output in dataset training. The entire iteration/block represents one *batch* of images, divided according to the *subdivisions*.

- 10 = indicates the current training iteration/batch.
- -nan = the total loss.
- -nan avg = is the average loss error, which should be as low as possible. As a rule of thumb, once this reaches below 0.060730 avg, stop the training.
- rate = represents the current learning rate, as defined in the *.cfg* file.
- 901.344055 seconds = represents the total time spent to process this batch.
- 200 images = the total amount of images used during training.

J. Extracted images to Cloud (Google Drive)

Cloud storage involves stashing data on hardware in a remote physical location, which can be accessed from any device via the internet. Clients send files to a data server maintained by a cloud provider instead of (or as well as) storing it on their own hard drives. After several training and testing, the detected image of a motorcyclist without a helmet was captured and saved to cloud storage. A google drive app was set up on a desktop with 15GB storage used as a cloud and a google drive account was created where the captured images were stored.

Figure 13. Extraction Process of Detected Motorcyclist Without Helmet

Figure 13 depicts the extraction procedure of a detected motorcyclist who is not wearing a helmet. Images extracted from the device were saved to a Google Drive folder on the desktop, which immediately synced to a Google Drive account that served as a cloud storage.

RESULTS AND DISCUSSION

The study utilized twelve (12) videos to test the motorcycle detection system, with an accuracy rate of 87.6 percent achieved in the morning setting. In the afternoon, however, the system encountered several variables that affected processing, such as weather conditions, resulting in an accuracy percentage of 88 percent. On the other hand, the evening setting produced the lowest accuracy percentage at 58.4 percent, as shown in Table 1. The table also indicates a positive result for the motorcycle detection system, with creasing accuracy from morning to afternoon settings. In Figure 14, samples of detected motorcycles during the training phase were presented, and the four videos gathered from various time settings demonstrated successful results. The confidence level of the detection system increased with repeated training for each time setting. However, the evening setting produced the lowest accuracy percentage due to factors such as heavy traffic that covered some motorcycles and poor detection confidence level during system training. For helmet detection, the study used three (3) videos, with one for each environment. Table 2 showed an increase in the percentage of helmet detection accuracy with repeated training for each time setting. Upon checking each video, the researchers found an increasing accuracy percentage for each time environment. The study recommends using the system during the morning and afternoon time settings due to better environmental conditions and less congested traffic. The researchers also suggest utilizing the YOLOv2 algorithm for object detection, as it can predict detections for over 9000 different object categories in real-time applications.

Table1. Motorcycle Detection Results

Video ID	Manual Count	System Count	Accuracy Percentage
VID_M001	33	29	
VID_M002	40	35	
VID_M003	26	23	
VID_M004	38	33	
Total	**137**	**120**	**87.6%**
VID_A001	30	27	
VID_A002	41	35	
VID_A003	32	28	
VID_A004	39	35	
Total	142	125	**88%**
VID_E001	57	30	
VID_E002	42	23	
VID_E003	40	25	
VID_E004	51	33	
Total	190	111	**58.4%**

Figure14. Sample Images of Detected Motorcycles

Table 2. Helmet Detection Results

Morning Settings Video ID	Manual Count	System Count	Morning Accuracy
VID_M001	22	17	77.3%
VID_M002	18	16	88.9%
VID_M003	9	8	88.9%
TOTAL	**49**	**41**	**83.7%**
Afternoon Settings Video ID	**Manual Count**	**System Count**	**Afternoon Accuracy**
VID_A001	23	12	52.2%
VID_A002	18	15	83.3%
VID_A003	20	17	85.0%
TOTAL	**61**	**44**	**72.1%**
Evening Settings Video ID	**Manual Count**	**System Count**	**Evening Accuracy**
VID_E001	33	12	36.4%
VID_E002	24	11	45.8%
VID_E003	29	12	41.4%
TOTAL	**86**	**35**	**40.7%**

As a result, repeated training for each time setting was successful in achieving the YOLO's minimum accuracy percentage of at least 76.8 percent mAP on VOC 2007. The other accuracy figure, on the other hand, did not reach its minimum mAP but improved on it by more than 50% from the minimum percentage, and the rest was due to human error and hardware limitations.

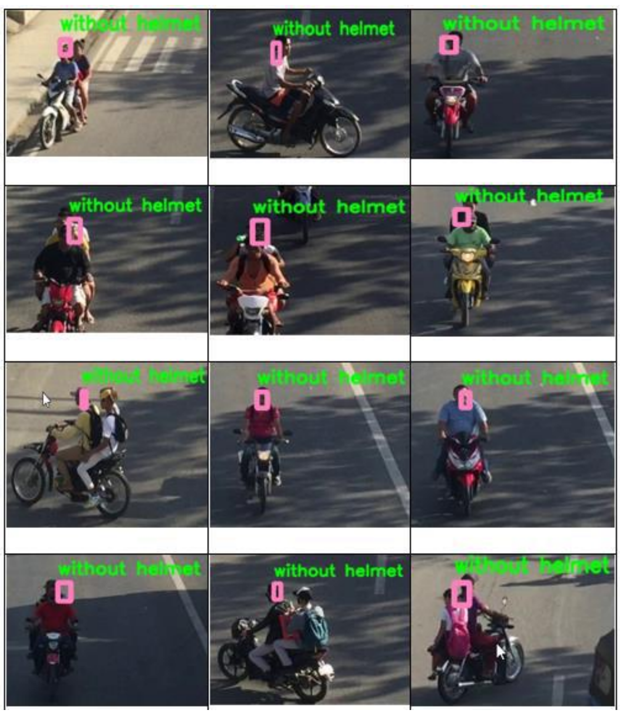


Figure 15. Sample Images of Detected Images Without Helmet

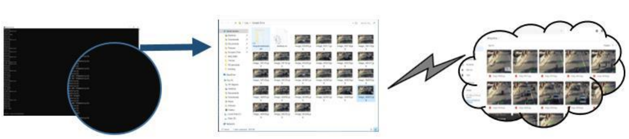


Figure 16. Extracted Images from the System to the Cloud Storage.

Figure 15 depicts the outcome of the caught motorcyclist who was not wearing a helmet. Figure 16 is a diagram that depicts the extraction of detected images. The system's detected images were extracted and saved to a Google Drive folder on the desktop, which automatically synced to a Google Drive account that served as a cloud storage.

CONCLUSION AND RECOMMENDATIONS

The study showed that the detection of non-helmeted motorcyclists is influenced by several factors, including the time of day, vehicle speed, lighting conditions, and the presence of shadows from different types of vehicles. The flow of traffic also plays a role in providing more data for detection. Despite the use of bounding boxes, there were instances where non-helmeted motorcyclists were not detected due to variables present in the video, such as the shadow of trees or street lights, which affected the video preprocessing. The researchers found that the accuracy of motorcycle and helmet detection varied depending on the time of day. The highest accuracy percentage rate of 83.7 percent was recorded during the morning period (9:30-10:00am) due to favorable environmental conditions for helmet detection and a higher number of motorcyclists riding without helmets during that time. The study conducted in Tagoloan, Misamis Oriental provided an appropriate environment for detecting a greater number of motorcycles and motorcyclists who are not wearing helmets due to fewer checkpoints in rural areas. The study findings revealed that the time setting has a significant impact on the system's detection accuracy percentage. The evening time (5:00-6:30pm) resulted in lower accuracy due to heavy traffic and busy roads, which led to several motorcycles and motorcyclists being covered by other vehicles. The researchers recommended that implementing automatic helmet detection for non-helmeted motorcyclists is more suitable in the morning and afternoon time settings due to less congested traffic and brighter surroundings. The YOLOv2 algorithm was found to be successful in detecting motorcycles and helmets, as it was useful at two separate stages of detection, resulting in increased system detection accuracy. Based on the experimental results, the proposed method exhibits superior performance compared to other existing methods. The motorcycle detection accuracy achieved 88 percent, while the helmet detection accuracy reached 83 percent, meeting the minimum accuracy requirement for YOLOv2. The researchers also found that the system's accuracy was affected during the evening environment due to heavy traffic and large vehicles obstructing motorcycles. Incorporating cloud storage in the method facilitated further analysis of the gathered data. Therefore, based on these conclusions, the following recommendations are suggested: first, utilizing YOLOv2 for object detection due to its capability to detect over 9000 object categories while still operating in real-time. Second, recording on flat highways with good vehicle visibility. Lastly, the proposed method is suitable for real-time applications and can be implemented.

ACKNOWLEDGMENTS

We would like to express our heartfelt appreciation to 3everyone who helped us complete this study. To the faculty of the Computer Engineering Department of the College of Engineering and Architecture at the USTP-CDO campus, who have all been patient and helpful during the research process. Thank you for your unflinching encouragement.

REFERENCES

[1] Pedrosa, L. P. (2017). Design and implementation of an automated system for detecting non-helmet use of motorcyclists. International Journal of Engineering and Technology, 9(4), 2974-2981.

[2] Zheng, Y., Li, X., & Wang, X. (2019). Helmet detection for motorcycle riders using deep learning. IEEE Access,7,28096-28105.

[3] Xu, X., Li, C., Li, X., Li, Z., Chen, Y., & Li, Q. (2020). Detection and analysis of motorcyclists without helmets based on deep learning. Journal of Intelligent & Fuzzy Systems, 38(3), 3001-3011.

[4] Wulff, J., 2017. Darkflow: Darknet Reinforcement Learning, OpenCV and TensorFlow. GitHub repository, available at: https://github.com/thtrieu/darkflow (Accessed: 2021, April 8).

77[5] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788). doi: 10.1109/CVPR.2016.91

[6] Python Software Foundation. (2020). Python 3.9.2 documentation. Retrieved from https://docs.python.org/3/index.html

[7] Park, J., & Kim, J. (2018). Object detection algorithm using YOLOv2 for improving real-time performance. Multimedia Tools & Applications, 77(20),27091-27106. doi:10.1007/s11042-018-6383-8

[8] Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., & Garcia-Rodriguez, J. (2018). A survey on deep learning techniques for image and video semantic segmentation. Applied Sciences, 8(12), 2272. doi: 10.3390/app8122272