Low Cost, High Performance Automatic Motorcycle Helmet Violation Detection

Aphinya Chairat, Matthew N. Dailey,
Somphop Limsoonthrakul, Mongkol Ekpanyapong
AI Center, Asian Institute of Technology
Khlong Luang, Pathum Thani 12120, Thailand
aphinya@ait.ac.th, mdailey@ait.ac.th,
Somphop.Limsoonthrakul@ait.ac.th, mongkol@ait.ac.th

Dharma Raj KC
University of Arizona
Tucson, Arizona 85721
kcdharma@email.arizona.edu

Abstract

Road fatality rates are very high, especially in developing and middle-income countries. One of the main causes of road fatalities is not using motorcycle helmets. Active law enforcement may help increase compliance, but ubiquitous enforcement requires many police officers and may cause traffic jams and safety issues. In this paper, we demonstrate the effectiveness of computer vision and machine learning methods to increase helmet compliance through automated helmet violation detection. The system detects riders and passengers not wearing helmets and consists of motorcyclist detection, helmet violation classification, and tracking. The architecture of the system comprises a single GPU server and multiple computational clients that cooperate to complete the task, with communication over HTTP. In a real-world test, the system is able to detect 97% of helmet violations with a 15% false alarm rate. The client-server architecture reduces cost by 20-30% compared to a baseline architecture.

1. Introduction

According to the World Health Organization, every year, 1.35 million people die because of road traffic crashes [27]. There are many causes of road traffic fatalities, especially human error or bad behavior by drivers. Many of these crashes involve motorcycles. A motorcycle rider, by wearing a helmet, can reduce his or her risk of a fatal injury by 42% and head injury by 69% [27]. As an example, in Thailand, motorcycle accidents kill around 5,500 people per year, and only 20% of people riding as a passenger on the back of a motorcycle wear helmets, according to the ThaiRoads Foundation [22]; this is one of the major causes of road fatalities in the country. Under such extreme conditions, helmet law enforcement would be one of the most effective ways to reduce fatalities. Indeed, police forces already have traffic officers in place to penalize offenders. But strict and ubiquitous enforcement would require many police officers everywhere and would incur additional difficulties such as dangerous pursuit of offenders and traffic bottlenecks. As an alternative, sensor technology, especially computer vision, can enable advanced solutions enabling us to improve the situation through automated helmet violation detection. In this paper, we consider the problem of monitoring for motorcyclists riding without a helmet using computer vision and machine learning based video analysis combined with an automated law enforcement information system to help to solve the problem of helmet law violations and thereby reduce traffic fatalities.

Methods for object detection have advanced rapidly in recent years. Classic methods such as Haar feature-based cascades [23] and histograms of oriented gradients (HOG) [2] have been supplanted by CNNs. Region-based CNNs (R-CNNs) [6] combine region proposals with convolutional neural networks (CNNs). The fast region-based convolutional neural network method (Fast R-CNN) [5] builds on R-CNNs to increase the speed of training and testing. Faster R-CNN [18] builds further on this work. Rather than running a separate selective search algorithm, it uses the same CNN for region proposals and region classification. In our work, we use YOLO [17] to detect motorcycles, as it is more effective than classic methods and is much faster than Faster R-CNN.

For tracking, classic methods likewise are not very accurate under natural conditions, leading to high false positive and miss rates. Yilmaz *et al*. [28] categorise tracking methods into three categories: point tracking, kernel tracking, and silhouette tracking. Tang *et al*. [21] identify characteristic patterns of occlusions. The method detects "double-persons" image patches in which two people occlude each other by 50% or more. The detector builds on the deformable part models approach of Felzenszwalb *et al*. [4]. Kong *et al*. [10] propose a system to count pedestrians in crowded scenes. The system uses a perspective transfor-

mation (homography) computed between the ground plane and the image plane for the region of interest (ROI). A density map is used for feature normalization. Linear models and neural networks are used during training to find the relationship between the features and the number of pedestrians in the image. Kristan *et al*. [11] propose an approach to track the position and scale of an object in a tracking-by-detection framework. The approach learns discriminative correlation filters from a scale pyramid representation to localize the target in each frame.

In this paper, we use the Kristan *et al*. [11] method as implemented in Dlib along with the YOLO detection result to match candidate detections with tracks. A new track is created whenever the detector finds an instance of the object in the first frame or for objects that do not match existing tracks. We classify each object in the track and take an average of the classifier's confidence score. If a track's average score is more than 0.5, we consider it a helmet violation. When a track does not get updated for some number of frames (five in our implementation), we delete that track.

For an object classification, classic methods feed an image patch directly to a neural network, SVM, or other classifier. HOG vectors and other feature vectors can be classified by backpropagation neural networks, SVMs, or other methods. Convolutional neural networks (CNNs) offer the benefit of automatically generating high-level features, offering better performance than feature-based methods, so long as a sufficient amount of training data is available. LeNet-5, CNN introduced by LeCun *et al*. [13], is trained by back-propagation. The network was designed for recognizing hand-written digits. In 2012, AlexNet [12] was introduced. It is a large high-performance network similar to LeNet but with more layers and more filters in each layer. In 2014, GoogLeNet [20] won the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). The architecture consists of a 22-layer deep CNN with a reduced number of parameters. More recently, in ILSVRC 2015, He *et al*. [8] introduced the residual neural network (ResNet). This network introduces skip connections. We find that GoogleNet provides the best tradeoff between accuracy, time, and resources.

To utilize detection, tracking, and classification in a practical system, we need high performance processing of multiple camera streams. Wang *et al*. [25] propose a method to perform large-scale video surveillance. Video from many cameras is processed by intelligent surveillance components (ISCs) and visualization surveillance components (VSCs). A wide area is monitored by a single system. Pouyanfar *et al*. [14] focus on intelligent video processing in terms of software, hardware, and applications. They connect cameras to a DVR/NVR and process video on the cloud. Qiu *et al*. [15] also use fixed camera views and processing on the cloud. Kim *et al*. [9] propose multi-camera based local position estimation for moving object detection. Each camera is connected to a separate DARKNET model. Our approach is most similar to that of Wang *et al*. [25]. We focus on a specific surveillance domain (motorcycle helmet law enforcement) and consider how to scale the ISCs that require GPU processing.

Finally, we consider the problem of helmet violation detection. Gualdi *et al*. [7] detect helmets on pedestrians at construction sites. The pedestrian detection method uses a LogitBoost classifier. Silva *et al*. [19] propose a system for automatic detection of motorcyclists without helmets. Local binary patterns, histograms of oriented gradients (HOG), and Hough transform descriptors are used for feature extraction. Desai *et al*. [3] demonstrate a system performing automatic helmet violation detection on public roads. They use background subtraction and optical character recognition for classification of license plate recognition, and they use background subtraction and the Hough transform for detection. Dahiya *et al*. [1] propose automatic detection of motorcyclists without helmets in surveillance video in real time. The proposed approach detects motorcycle riders by first performing background subtraction and object segmentation. For each motorcycle, they determine whether the rider is wearing a helmet or not using visual features and a binary classifier. Vishnu *et al*. [24] detect motorcyclists without helmets in video using convolutional neural networks. Like Dahiya *et al*., they use adaptive background subtraction on video frames to restrict attention to moving objects. Then a convolutional neural network (CNN) is used to select motorcyclists from among the moving objects. Another CNN is applied to the upper one-fourth of the image patch for further recognition of motorcyclists driving without helmets. Wonghabut *et al*. [26] describe an application using two different CCTV cameras mounted at different angles. One camera is used for detecting motorcyclists involved in violations. The other captures images of the violating motorcyclist. Haar cascades are used as a descriptor for extracting object features. Raj KC *et al*. [16] propose a helmet violation processing system using deep learning. The system comprises motorcycle detection, helmet versus no-helmet classification, and motorcycle license plate recognition.

Our work is the continuation and extension of Raj KC *et al*. [16]'s system. In this paper, we call the Raj KC *et al*. [16] system the baseline system. We use YOLO for motorcycle detection, Kristan's method for tracking, GoogleNet for classification, and a specific system architecture for multiple camera processing for helmet violation detection. The contributions of this paper are

1) Comparison of methods for classifying motorcycle riders as to whether they are wearing helmets or not. We find that a GoogleNet CNN trained from scratch performs well.

2) Analysis of speed, resource, and accuracy tradeoffs for classical and deep learning based detection of motorcycles in real world images across multiple scenes and conditions.

3) Design of a low cost, high performance CPU-GPU system architecture for motorcycle helmet violation detection. We analyze the application, and we find that the most time-consuming activity is YOLO-based motorcycle detection. We improve throughput using a hybrid structure for the surveillance components. We use one GPU machine as a server and another machine as a compute client. The client machine sends requests and images to the server. The server performs object detection in those images and replies with bounding boxes of the object instances.

Our application requires a graphics processing unit (GPU). One computer with an Intel Core m5-6Y57 Dual-Core processor, 4 GB of 1866 MHz LPDDR3 memory, and Integrated Intel HD graphics currently costs 617.44 USD. The Nvidia Jetson TX2 costs 479 USD. The Nvidia GTX 1080Ti with 11 GB GDDR5X 352 bit memory costs about 825.40 USD. On the cloud, Amazon provides Amazon Elastic Graphics, which requires 0.4 USD/hour with 8 GB graphics memory. Deploying a complete system for one junction requires many concurrent processes. Based on the prices mentioned above, we aim to minimize the cost of the system when we need to deploy many applications.

4) Analysis of accuracy and efficiency of the entire system for real-world processing of traffic violations.

The result is a performance model that could be applied to other intelligent video processing applications.

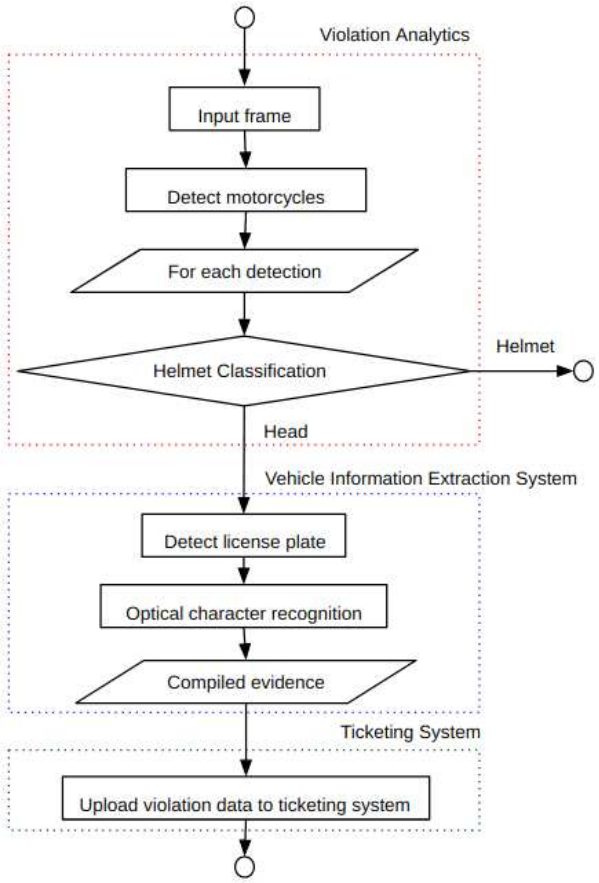

Figure 1. Overview of the system.

2. Motorcycle Helmet Violation Detection Methods

2.1. System Overview

The system detects helmet violations in real-world road traffic scenes. The system consists of three parts: analytic system (video processing system), vehicle information extraction system, and a cloud-based web application for ticketing, as shown in Figure 1.

2.2. Motorcyclist Detection

As already discussed, existing work on motorcycle detection uses a variety of approaches. The motorcycle detection module of Raj KC *et al*. [16] uses Haar cascades. In this paper, we compare HOG, Haar cascades, and YOLO.

In Experiment MD-1, we compare these three methods using training data from one location and test data from another location. The training data consist of 1,742 positive images and 1,508 negative images to create the HOG model and the Haar cascade model. Positive images consist of cropped motorcycles. Negative images are images without motorcycles. The CNN (YOLO) uses 1,255 images (full frame) from the same location as those for the HOG and Haar cascade models. Training uses full frame images and text files for each image. The text file has data on motorcycle locations. Example training images for Experiment MD-1 are shown in Figure 2. In the test phase of Experiment MD-1, we use 630 motorcycle images to check accuracy. Examples of testing images for Experiment MD-1 are shown in Figure 3.

Based on the best method from MD-1, we performed Experiment MD-2, in which we built a new CNN model using a larger dataset. We compare the three methods using training data from one location and test data from another location. We improve the motorcyclist detection dataset by collecting more data from many locations. We used 5,323 images to train the final model. To check the accuracy of the model, we used 11,276 motorcycle images that are completely unseen data. An example test image used in Experiment MD-2 is shown in Figure 4.

2.3. Helmet Violation Classification

To build a baseline model for helmet violations, we obtained 960 images in the "violation" class and 931 images in the "non-violation" class. We use the top half of the mo-

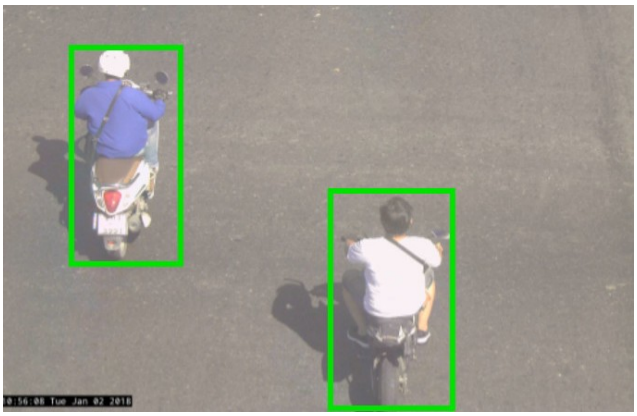

Figure 2. Example training image for Experiment MD-1. Rectangles are locations of motorcycles, identified by humans.

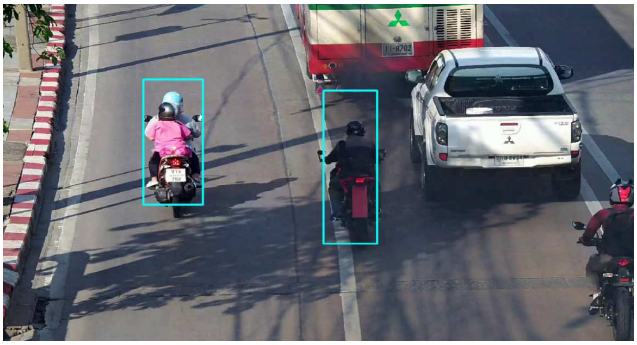

Figure 3. Example test image for Experiment MD-1. Rectangles are locations of motorcycles detected by the YOLO model built in Experiment MD-1.

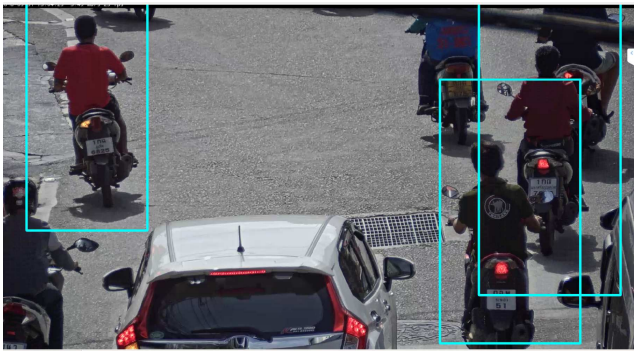

Figure 4. Example test image for Experiment MD-2. Rectangles are locations of motorcycles detected by the YOLO model built in Experiment MD-2.

torcycle bounding box for training and testing. We consider a motorcycle with two riders, one wearing a helmet and another not wearing a helmet as a violation. Examples of training images in the "violation" and "non-violation" classes are shown in Figures 5 and 6.

We trained a GoogleNet CNN from scratch using a batch size of 64 and a learning rate of 0.001. We use GoogleNet

Figure 5. Example training images in violation class.

Figure 6. Example training images in non-violation class.

because this model consumes the least amount of GPU memory among the best image classifiers introduced in recent years and is also quite accurate. In Experiment HC-1, we tested the hypothesis that a similar number of training items for each class would give good results. We used 4,900 images in the violation class and 4,852 images in the non-violation class. In Experiment HC-2, we increased the data set size to 7,899 training images, while the training data for the non-violation class was unchanged, at 4,852 images.

2.4. Tracking

The typical approach to tracking for helmet violation detection applications, including Raj KC *et al*. [16], considers the simple overlap between a candidate in the previous frame and a candidate in the current frame to determine whether it is the same object. We implemented this basic approach as a baseline. When the overlap area is more than a threshold, it is considered the same object. If the overlap area is less than the threshold area, the two detections are considered to represent different objects, and we wait to check overlap with candidates in the next frame. If the candidate does overlap with a candidate in the next frame, we continue tracking. If it fails to overlap again, we end the track.

In Experiment T-1, we validate an improvement to this basic tracking method in which we consider the similarity of two regions as well as their overlap. We increased the number of allowable misses before a track is deleted to 2. That is, if the track is not updated for more than two frames,

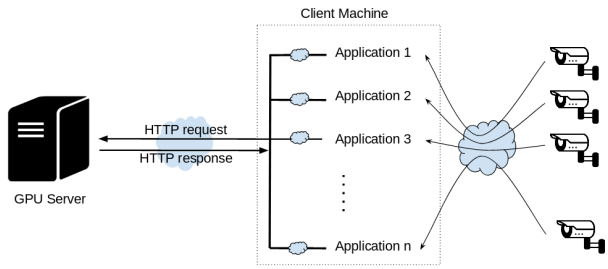

Figure 7. Overview of multi-cameras processing sharing one GPU model.

that track will be deleted. This helps when a motorcycle is lost for one or two frames then is re-detected, with a small penalty in terms of memory utilization. We found that even with appearance modeling, we still observe a large number of ID switches.

To address this, we further implemented an improved tracking method using Kristan's method, which uses a classification confidence measure instead of appearance similarity in order to decide whether or not to continue a track.

2.5. Performance

Here we explain the method and experiments used to test the performance of the overall system.

At one location, there can be many lanes and many cameras. In a first version of the application, we paired one detection model with each camera and one camera with each lane. We found that this required a large amount of GPU memory. We then implemented the motorcycle detection model on a server using one GPU. The server was an Intel Core i5 CPU 650 @ 3.20GHz $\times$ 4 with 7.7 GiB RAM and a GeForce GTX 1060 6GB/PCIe/SSE2 GPU. The client machine was an Intel Core i5-7200U CPU @ 2.50GHz $\times$ 4 with 7.7 GiB RAM and a GeForce 920MX/PCIe/SSE2 GPU. We use Boost.Asio version 1.58 to handle threads. Communication between the client and server utilizes JSON over HTTP. The server and client are assumed to be on the same subnetwork. An overview of the system is shown in Figure 7,and a sequence diagram describing the dynamic execution of the system is shown in Figure 8. We ran one application with one model to check how many frames per second the application would run. Then we increased the number of applications and experimented with different-sized image frame buffers: 1, 5, 10, 50, 100, 200 and 500 images.

3. Results

Here we describe the experimental validation of the improvements to the three modules, motorcyclist detection, violation classification, and tracking, described above.

Table 1. Test set comparison of detection methods in Experiment MD-1. Training and testing data were from different locations.

Method	Precision	Recall	F1 Score
HOG	0.98	0.65	0.81
HAAR	0.98	0.66	0.82
CNN (YOLO)	**0.99**	**0.73**	**0.86**

Table 2. Test set comparison of detection methods in Experiment MD-2. Training and testing data were from different locations. Note that this dataset is more challenging than that used in Experiment MD-1.

Method	Precision	Recall	F1 Score
HOG	0.94	0.45	0.69
HAAR	0.96	0.45	0.70
CNN (YOLO)	**0.96**	**0.61**	**0.78**

3.1. Motorcyclist Detection

We performed Experiment MD-1 to find the best method for detection, and then in Experiment MD-2, we further improved the best model with an increased amount of training data. The resulting system still makes mistakes, but the error rate is dramatically decreased.

We built several versions of the motorcycle detection system, based on HOG, Haar, and CNN (YOLO). We found that detection using HOG is not flexible when the environment changes. This occurs, for example, when the training and testing locations are different or the shape of the object changes somewhat. Table 1 shows the results of the comparison in Experiment MD-1.

While the HOG and Haar cascade models can in principle be used at different locations, the accuracy results under transfer are not very convincing. The CNN model, which was trained with less data, is nevertheless better than the HOG and Haar cascades. If we add more training data, the model may further improve. Hence, we conclude that while HOG and Haar cascades are overly sensitive to scene view changes, the CNN (YOLO) model is much less affected.

In Experiment MD-2, we attempted to further improve the motorcycle detection rate by collecting more data from many locations and training another HOG model, Haar cascade, and CNN (YOLO). Results are shown in Table 2. The YOLO CNN performs best. The recall is lower than in Experiment MD-1 because the test data are more difficult. The data set includes motorcycles in which we cannot see the license plate because of overlap with other motorcycles.

3.2. Helmet Violation Classification

The ground truth image set for violation classification contains 1,498 images: 556 violation images and 942 non-

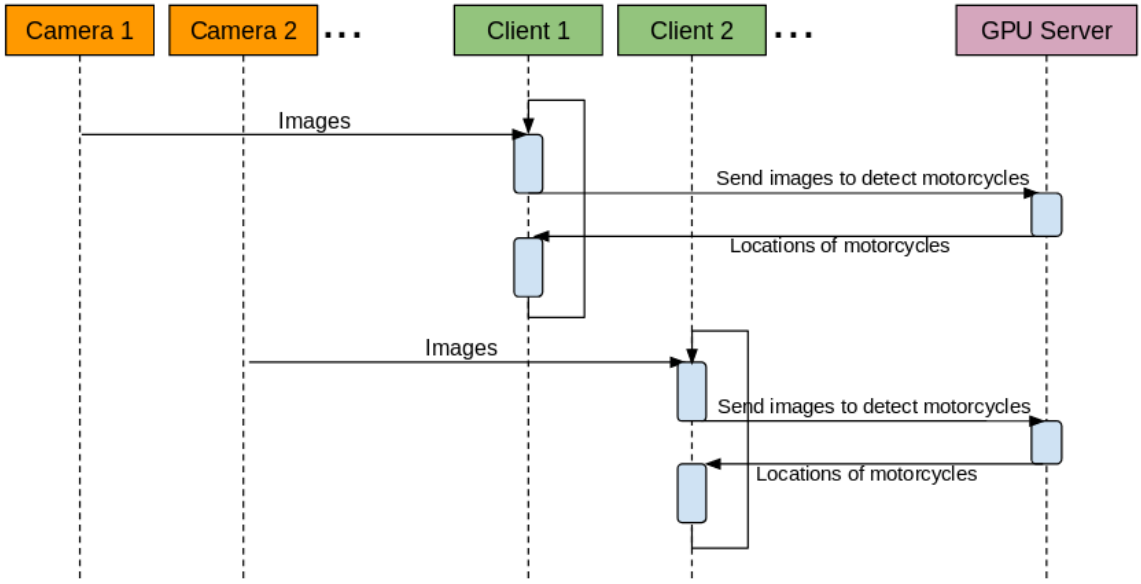

Figure 8. Client-Server communication in the multi-camera processing system diagram.

Table 3. Results of helmet violation classification from initial baseline model, (GoogleNet).

	Predict: Helmet	Predict: Head
Actual: Helmet	823	119
Actual: Head	59	497
Total accuracy:	88%	

Table 4. Results of helmet violation classification in Experiment HC-1 (balanced data, GoogleNet).

	Predict: Helmet	Predict: Head
Actual: Helmet	899	43
Actual: Head	32	524
Total accuracy:	94%	

violation images. The accuracy of the baseline model was 0.88. In Experiment HC-1, we used a similar number of training items for each class, and we obtained an accuracy of 0.94. In Experiment HC-2, using a large positive training set, we obtained an accuracy of 0.95. The details are shown in Tables 3, 4, and 5. The remaining problem is that the model is confused in some cases; for example, black helmets and heads are similar. Some riders wear a cap or cover their head with a scarf or other clothing, and some complex cases also cause errors. Examples of wrong classification results are shown in Figure 9.

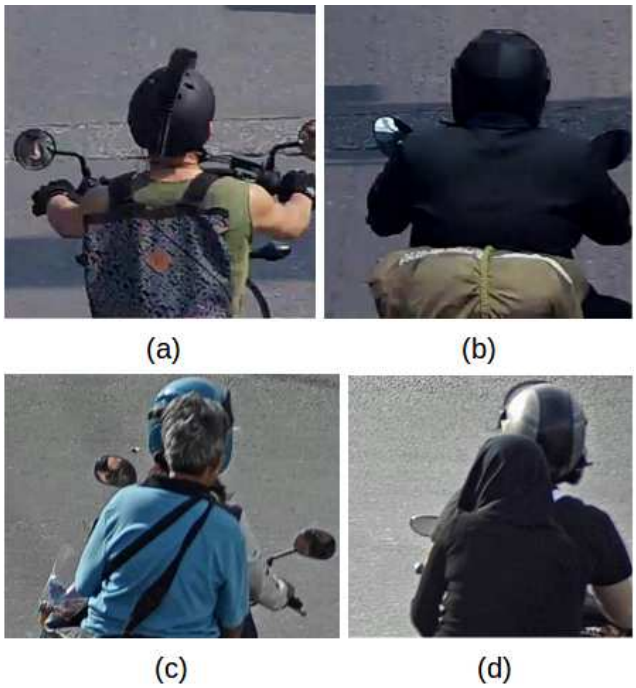

Figure 9. Examples of wrong classification. (a,b) Helmets predicted as heads. (c,d) Heads predicted as helmets.

3.3. Tracking

The baseline tracking method sometimes gives duplicate violations. This type of error occurs most frequently when the motorcyclist drives very quickly, so that the bounding box of the motorcycle does not overlap sufficiently in suc-

Table 5. Results of helmet violation classification in Experiment HC-2 (large dataset, GoogleNet).

	Predict: Helmet	Predict: Head
Actual: Helmet	899	43
Actual: Head	29	527
Total accuracy:	95%	

Table 6. Results of Experiment T-1. MT 30%: More than 30% tracked. LT 30%: Less than 30% tracked. MAT: Tracks that are mixed with one or more other track.

GT	MT 30 %	LT 30 %	MAT	Miss
164	108	7	37	8

Table 7. Results of tracking with Kristan *et al*. method in Experiment T-2. The Dlib implementation of the tracking method is superior to histogram matching.

GT	MT 30 %	LT 30 %	MAT	Miss
164	125	31	0	8

cessive frames. When a motorcycle is missed, a tracking-by-detection approach will not help. Another problem was that when a motorcycle enters a crowded area, the tracks can become confused.

Although duplicate violation declarations will eventually be screened when the license plate is recognized, resource utilization will be improved if fewer duplicate candidates are submitted for license plate recognition. The results of Experiments T-1 and T-2 are shown in Tables 6, and 7, showing dramatically reduced errors.

3.4. Architecture Evaluation

The baseline system performs detection, classification, and tracking on one machine. The motorcycle detector in the baseline system runs at approximately 35 frames per second. The client-server architecture places the detection module on a separate GPU server machine. Classification and tracking are performed on the client machine, which does not require a discrete GPU. In the client-server architecture, the server application uses 1.760 GB of GPU memory for one YOLO model. We trained this detection model using YOLO version 3. When we run the client-server architecture with one application on the server, we found that increasing the buffer size increases throughput in terms of frames per second but also increases latency for the client.

When we increase the number of client applications, the buffer size must be decreased. Client machines can run more applications when the buffer size is small. A comparison of time usage for different buffer sizes and number of applications is shown in Figure 10.

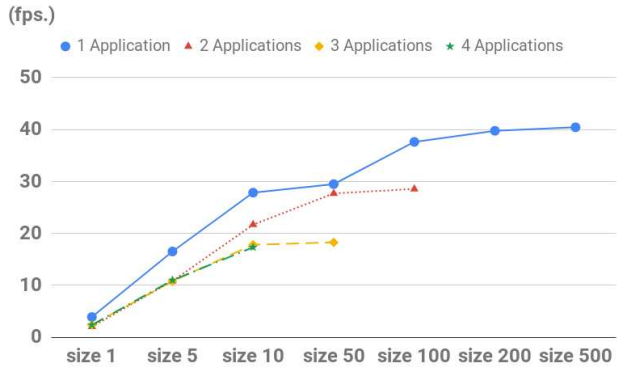

Figure 10. Frame rate for different buffer sizes, with client and server on separate machines on the same LAN. Frame rate is averaged over all applications. The GPU is the GTX 1060.

The baseline system has lower latency because the system does not waste time on network communication. The baseline uses one GPU machine, but the number of applications that can be run on it depends on the GPU memory of the machine. The client-server method has higher latency because the system has communication overhead between server and client. One GPU machine is needed to run the server, and another machine (no need for a GPU) runs the client. The clients' main constraint is CPU capacity. If we want to scale with minimal resources, this method is best.

3.5. Performance Model

To install the system at one junction with 12 cameras, we estimate compute costs using an Intel Core i5-7400 CPU @ 3.0GHz $\times$ 4 with 8 GB RAM and one GPU machine with a GTX 1060 with 6 GB, which currently costs 721.84 USD at our location. A suitable machine without a GPU costs 482.95 USD.

The baseline would require at least 6 GTX GPUs. The total cost for one junction would therefore be approximately 4,330.25 USD. If we use the client-server architecture, the total cost is much lower at 3,377.35 USD. The system with two servers and three client machines further only costs 2,891.61 USD, representing a 20-30% reduction from baseline with no decrease in accuracy. The architecture of the recommended system is shown in Figure 12. The cost estimates are shown in Table 8.

The complete helmet violation system was evaluated on one continuous hour of video of a real-world scene not used for training or testing, and multiple detections of the same violation were discarded. The result is shown in Table 9.

We also evaluated the helmet violation detection system during the night time using a video from a different real-world scene not used for training or testing. The result is shown in Table 10. To achieve these results, it was nec-

Figure 11. An example view at night time.

Table 8. System design with frame buffer size recommendation and cost estimates. SM: Number of server machines. CM: Number of client machines. FB: Frame buffer size. Cost units are USD.

	SM	CM	Cameras	FB	Cost
Baseline	6	0	12	-	4,330.25
Arch 1	2	4	12	40-50	3,377.35
Arch 2	2	3	12	20-30	2,891.61

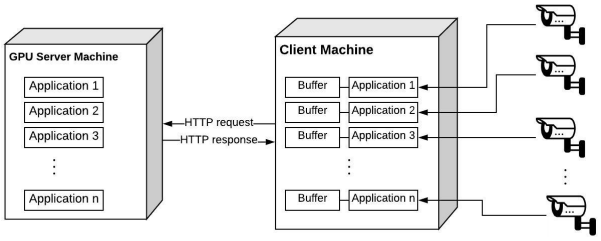

Figure 12. Architecture of client-server system.

Table 9. Final helmet violation evaluation during daytime. VA: Violations from application. TP: True positives.

GT	VA	TP (Recall)	False Alarms	Misses
153	178	150 (98%)	28 (15%)	3

Table 10. Final helmet violation evaluation in night time. VA: Violations from application. TP: True positives.

GT	VA	TP (Recall)	False Alarms	Misses
125	95	84 (67%)	11 (11%)	41

essary to install sufficient extra lighting of the intersection with over head floodlights. An example view at night time is shown in Figure 11.

4. Conclusion

We have developed a high performance, low cost helmet violation detection system and a performance model for the violation detection system. The system comprises motorcycle detection and helmet violation/non-violation classification. The YOLO convolutional neural network (CNN) is more effective than histograms of oriented gradients (HOG) or Haar cascades. For violation classification, a GoogleNet CNN trained from scratch on 12,751 images performs well. We find that tracking using a tracking-by-classification framework improves performance. We developed a performance model for the CPU-GPU architecture in which we have multiple clients on separate machines feeding data to a GPU server that performs object detection. The connection between client and server uses JSON over HTTP between two separate machines on the same LAN. The buffer size and number of clients affect the time usage of each application. If the size of the buffer is small, we can run more applications, but processing time is increased. The GPU server is an Intel Core i5 CPU 650 @ 3.20GHz $\times$ 4 with 7.7 GB RAM and a GeForce GTX 1060 6GB/PCIe/SSE2. The client machine uses an Intel Core i5-7200U CPU @ 2.50GHz $\times$ 4 with 7.7 GB RAM and a GeForce 920MX/PCIe/SSE2. This modest platform can detect and process helmet law violations in four camera streams. The GPU server requirement scales more slowly than the CPU client requirement.

5. Discussion

In this work, our goal is to get recall sufficient to deter violations at minimum cost. If we increase the resources available, recall may improve, or false positive rates may reduce, but at some point, we obtain diminishing returns and possibly obtain negative returns on the incremental investment. Our system has a 15% false alarm rate, but this is acceptable, as this is a human-in-the-loop system in which a human officer has to check each violation before a ticket can be issued. False alarms are therefore relatively easy to ignore or remove. To reduce false alarms, the motorcycle detection and helmet violation classification can be improved by increasing the training data set to include observed false alarms or by collecting more data from that specific location.

The main limitation of the system is the quality of the dataset used for training. Another is that although the system works well at night time, recall is lower because of unclear images. Night time accuracy can be further improved by adding additional lighting to make the images clear enough.

Acknowledgment

This research was supported by the Thailand Research Fund, the Safer Roads Foundation, and graduate fellowships for Aphinya Chairat and Dharma Raj K.C. from the Royal Thai Government and the Asian Institute of Technology.

References

[1] K. Dahiya, D. Singh, and C. K. Mohan. Automatic detection of bike-riders without helmet using surveillance videos in real-time. In *International Joint Conference on Neural Networks (IJCNN)*, pages 3046–3051. IEEE, 2016.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005.

[3] M. Desai, S. Khandelwal, L. Singh, and S. Gite. Automatic helmet detection on public roads. *International Journal of Engineering Trends and Technology (IJETT)*, 35.

[4] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1627–1645, 2010.

[5] R. Girshick. Fast R-CNN. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1440–1448, 2015.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

[7] G. Gualdi, A. Prati, and R. Cucchiara. Contextual information and covariance descriptors for people surveillance: an application for safety of construction workers. *EURASIP Journal on Image and Video Processing*, 2011.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[9] J. Kim, Y. Koo, and S. Kim. MOD: Multi-camera based local position estimation for moving objects detection. In *IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 642–643, 2018.

[10] D. Kong, D. Gray, and H. Tao. Counting pedestrians in crowds using viewpoint invariant training. In *British Machine Vision Conference (BMVC)*, pages 1–6, 2005.

[11] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, F. Porikli, L. Cehovin, G. Nebehay, G. Fernandez, T. Vojir, A. Gatt, et al. The visual object tracking vot2013 challenge results. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 98–111. IEEE, 2013.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[13] Y. LeCun et al. Lenet-5, convolutional neural networks. *URL: http://yann. lecun. com/exdb/lenet*, page 20, 2015.

[14] S. Pouyanfar, Y. Yang, S.-C. Chen, M.-L. Shyu, and S. Iyengar. Multimedia big data analytics: a survey. *ACM Computing Surveys*, 2018.

[15] H. Qiu, X. Liu, S. Rallapalli, A. J. Bency, K. Chan, R. Urgaonkar, B. Manjunath, and R. Govindan. Kestrel: Video analytics for augmented multi-camera vehicle tracking. In *Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 48–59, 2018.

[16] K. D. Raj, A. Chairat, V. Timtong, M. N. Dailey, and M. Ekpanyapong. Helmet violation processing using deep learning. In *International Workshop on Advanced Image Technology (IWAIT)*, pages 1–4, 2018.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[18] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[19] R. Silva, K. Aires, T. Santos, K. Abdala, R. Veras, and A. Soares. Automatic detection of motorcyclists without helmet. In *Latin American Computing Conference (CLEI)*, pages 1–7, 2013.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[21] S. Tang, M. Andriluka, and B. Schiele. Detection and tracking of occluded people. *International Journal of Computer Vision*, 110:58–69, 2014.

[22] ThaiRoads Foundation. Road safety situation in Thailand 2014-2015. `http://www.thairoads.org/`, 2015. Accessed: 2017-01-25.

[23] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[24] C. Vishnu, D. Singh, C. K. Mohan, and S. Babu. Detection of motorcyclists without helmet in videos using convolutional neural network. In *International Joint Conference on Neural Networks (IJCNN)*, pages 3036–3041. IEEE, 2017.

[25] Y.-K. Wang, C.-T. Fan, and C.-R. Huang. A large scale video surveillance system with heterogeneous information fusion and visualization for wide area monitoring. In *International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, pages 178–181, 2012.

[26] P. Wonghabut, J. Kumphong, T. Satiennam, R. Ungarunyawee, and W. Leelapatra. Automatic helmet-wearing detection for law enforcement using CCTV cameras. In *IOP Conference Series: Earth and Environmental Science*, volume 143. IOP Publishing, 2018.

[27] World Health Organization. Strengthening road safety in Thailand. `http://www.searo.who.int/thailand/areas/roadsafety/en/`, 2015. Accessed: 2017-01-25.

[28] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM computing surveys (CSUR)*, 38(4):13, 2006.