Helmet Violation Processing Using Deep Learning

Dharma Raj KC[1], Aphinya Chairat[2], Vasan Timtong[2], Matthew N. Dailey[2], Mongkol Ekpanyapong[2]
Asian Institute of Technology
Khlong Luang, Pathum Thani 12120, Thailand
dharma@ait.asia, aphinya.chairat@ait.asia, vasant@ait.asia, mdailey@ait.asia, mongkol@ait.asia

***Abstract*—Road accidents are one of the major causes of human deaths. Among the different types of road accidents, motorcycle accidents are common and cause severe injuries. The helmet is the motorcyclist's main protection. Most countries require the use of helmets by motorcyclists, but many people fail to obey the law for various reasons. We present the development of a system using image processing and deep convolutional neural networks (CNNs) for finding motorcyclists who are violating helmet laws. The system comprises motorcycle detection, helmet vs. no-helmet classification, and motorcycle license plate recognition. We evaluate the system in terms of accuracy and speed. The system has been deployed at several locations in Bangkok and Phuket, Thailand since 2016. Early reports indicate that compliance with motorcycle helmet laws is increasing.**

Keywords—deep learning; convolutional neural network; helmet violation.

I. INTRODUCTION

One of the leading causes of unnatural deaths today is road fatalities, primarily motorcycle accidents. Particularly, among all road fatalities, motorcycle accidents accounted for 9% in Europe, 20% in the United States, and 34% in western Pacific and southeast Asian countries [1]. Helmets are the main protection device for motorcyclists, so where compliance is low, automated processing has the potential to dramatically increase compliance, thereby saving human lives.

Until very recently, most of the methods used for object detection and object classification used methods such as Haar, HOG, local binary patterns (LBP), the scale invariant feature transform (SIFT), or speeded up robust features (SURF) for feature extraction and then support vector machines (SVM), random forests, or AdaBoost for the classifier.

Silva et al. [7] use methods such as histograms of oriented gradient (HOG), LBP, and the wavelet transform (WT) for feature extraction for classifying motorcyclists with helmets and without helmets. They use multiple combinations of the base features such as HOG+LBP, HOG+WT, LBP+WT, and HOG+LBP+WT, obtaining seven possible feature sets. They combine the seven features with six classifiers, namely SVM, radial basis function network (RBNF), multilayer perceptron (MLP), naive Bayes, random forest, and k-nearest neighbors (KNN). On a test set with 255 images (151 with helmets and 104 without helmets), the MLP classifier using HOG descriptors showed the best result, with 91.3% accuracy. As convolutional neural networks (CNNs) have recently outperformed custom feature-based methods in many domains, there is evidence that the use of CNNs could increase the accuracy of helmet/no-helmet classification.

The origin of automatic license plate recognition systems goes back at least to the 1990s, when Gonzalez and Herrera [9] published their methods for U.S. license plate detection and recognition. They use transition detection system for license plate detection, edge finding, contour extraction and validation for segmentation, and structural analysis techniques such as convex hulls, and bays and holes for character classification.

In recent years, CNNs performing both automatic feature extraction and classification have outperformed previously dominant methods in many problems. Advances in graphical processing units (GPUs), along with the availability of more training data for neural networks to learn, have recently enabled unprecedented accuracy in the fields of machine vision, natural language processing, and speech recognition. Nowadays, all state-of-the-art methods for object classification, object detection, character classification, and object segmentation are based on CNNs. See for example the methods used in the ImageNet large scale visual recognition challenge [2].

Li and Shen [10] use a deep convolutional neural network and long-short term memory (LSTM) for the license plate recognition and character extraction process. They use a CNN for license plate detection. They use two methods for segmentation and recognition. The first is character segmentation based recognition using image binarization, connected component analysis, and character recognition. The second is a sequence labeling based method using CNNs and recurrent neural networks (RNNs).

Jaderberg et al. [3] have shown the use of CNNs for text detection and recognition provides significant improvement over existing methods.

II. HELMET VIOLATION PROCESSING USING DEEP LEARNING

To the best of our knowledge, our system is the first to use CNNs for helmet violation and no-violation detection. The system performs well even when there are multiple persons on the motorcycle and one of them is not wearing a helmet.

A. Overview

A schematic overview of our helmet violation detection system is shown in Figure 1. Video feeds are first run through a motorcycle detector incorporating a linear SVM for histograms of oriented gradient (HOG) feature vector classification. The

978-1-5386-2615-3/18/$31.00 ©2018 IEEE

region expected to contain the rider's head is passed to a CNN for helmet and no-helmet classification. Motorcycle detections

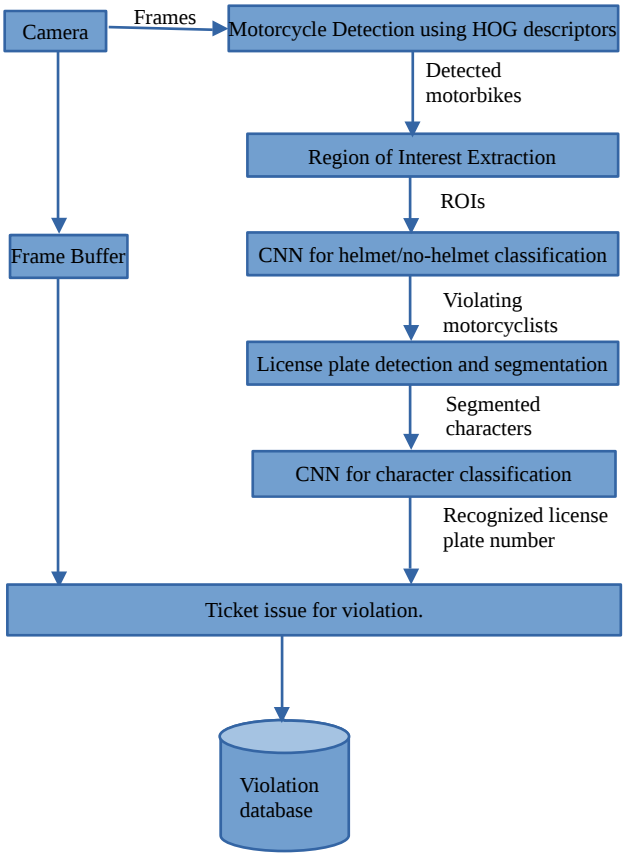

Fig 1. System data flow.

classified as helmet violations are further processed for location of the license plate using a Haar cascade detector. The license plate is segmented using thresholding and connected components, then the characters in the license plate are classified. An image of the violating motorcyclist, the motorcyclist's license plate, and a short video clip of the violating motorcyclist are automatically uploaded to an automated ticketing systems from which police operators can issue a legal violation citation and send the citation to the offender in the email.

B. Helmet/no-helmet classifier

We have developed several different CNN models for the helmet/no-helmet classifier. Compared to an earlier version of the classifier incorporating a head/helmet detector and HOG descriptor classifier, the CNN for helmet/no-helmet classification improves accuracy from 80% to 98%. Though the more accurate classifier requires a high performance GPU to run in real time, we have also developed a classifier that obtains 90% accuracy and runs in real time on a CPU.

C. License plate character classifier

We have experimented with several different CNN models for character classification of motorcycle license plates. An early version of the system had an accuracy of 85.86% on a test set of 3105 manually cropped characters. The CNN for character classification improved the accuracy to 97.75% on same dataset.

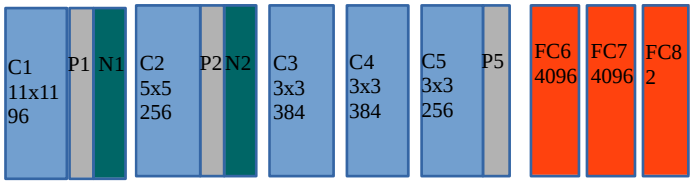

Fig 2. AlexNet CNN architecture.

III. Experimental Evaluation

We performed two groups of experiments. The first set concern helmet violation classification. The second set concerns character recognition.

A. Experiments for violation classification

We performed a series of experiments to find the best model for the task by changing the hyperparameters and number of training examples as well as applying data augmentation techniques. We have performed all these experiments using the Caffe framework, which is an open source framework for deep learning experiments [4].

Following are the experimental results for helmet/no-helmet classification.

1) Experiment 1: To find a fast and accurate model for helmet/no-helmet classification, we started from a simple model and increased the model complexity incrementally. The input image size was 227×227 with RGB color input channels.

Motorcycle images detected by the existing system were used to generate the data for helmet/no-helmet classification. The detected motorcycles were first manually classified into helmet and no-helmet classes. For training, we cut the upper one third of each motorcycle image as a region of interest extraction process, because this is the region most likely to contain the helmet/no-helmet information. We used a 7×7 convolutional filter with a stride of 2 for the first convolutional layer. We prepared 900 images (450 with helmets and 450 without helmets) for testing. The number of training and validation images were 931 and 200 for the positive class (non-violation) and 960 and 200 for the negative class (violation), respectively. We tried different hyperparameter values with learning rate of 0.01 and 0.001, batch size of 16 and 32, and stochastic gradient descent (SGD), and AdaGrad. The SGD model did not converge, but the AdaGrad optimization algorithm led to a validation accuracy of 64.25% and a test accuracy of 59.76%.

2) Experiment 2: We increased the number of training images for the positive class to 4227 and the negative class to 8481. The number of validation and test images were fixed, for purposes of fair comparison. We re-used the previous model and trained using different optimization algorithms

such as AdaGrad, AdaDelta, Adam, and RMSProp. AdaGrad gave the best performance, with a validation accuracy of 76.5% and a test accuracy of 74.11%.

3) Experiment 3: We increased the depth of neural network to include two convolutional layers, two ReLU layers, two pooling layers, and one fully-connected layer. The network consists of a first convolutional layer with a filter size of 7×7, a ReLU layer, a 3×3 max-pooling layer, a 5×5 filter convolutional layer, a ReLU layer, a 2×2 max polling layer, a fully-connected layer with 2704 units, and a final classification layer with two classes. We trained with different optimization algorithms. Among them, AdaDelta gave 87.10% validation accuracy and 93.11% test accuracy.

We tested the run time of the violation/non-violation CNN models on a GeForce GTX 1070 GPU from Nvidia with 8GB RAM and an Intel Core-i7 CPU with 8GB RAM. It was able to classify 40 images per second on a GPU, and 4.2 images per second on CPU.

4) Experiment 4: We increased the number of training images for the violation class and non-violation class to 10,055 and 7952, respectively. A new model was designed with 11 layers including the final softmax. It consists of three convolutional layers, three ReLU layers, three pooling layers, one fully-connected layer, and one softmax layer.

We performed data augmentation using rotation, skewing, random zooming, and shifting. The maximum value of rotation was five degrees in clockwise and counter clock-wise directions. The maximum translation on the x-axis was 0.1, the maximum translation on the y-axis was 0.02, the maximum shear angle was 0.1, the maximum zoom range was 0.2, and the remaining pixels were filled using the nearest neighbor. We thus produced about ten images for each original image. We trained the network using the augmented data and the AdaDelta optimization algorithm. We obtained 95% validation accuracy and 96.11% test accuracy.

This network was able to classify 37.5 images per second on the GPU, and 2.17 images per second on the CPU.

We tried changing the output of the first convolutional layer from 64 to 128, which did not result in much change in accuracy, suggesting the importance of depth over width of the neural network.

5) Experiment 5: We used the augmented data set from Experiment 4 with the reference model AlexNet provided with the Caffe framework. It is an implementation of the model from Krizhevsky et al. [5].

With SGD optimization, and a learning rate of 0.001, we obtained validation accuracy of 98% and a test accuracy of 97.22%. This model was able to classify 33.33 images per second on the GPU, and 0.98 image per second on the CPU.

Fig. 2 shows the architecture of AlexNet. C denotes a convolutional layer, e.g., C1 for the first convolutional layer. P denotes a pooling layer. N denotes a local response normalization (LRN) layer. The number below C e.g., 11×11 for C1, denotes the size of the kernel used, and the number below kernel size e.g., 96 in C1, denotes the number of outputs from that layer. FC denotes a fully-connected layer, and the number below it denotes the number of outputs from that layer. Each convolutional layer and fully connected layer is followed by ReLU layer, except FC8. FC6 and FC7 are followed by dropout layers.

TABLE I. VIOLATION AND NON-VIOLATION EXPERIMENTS COMPARISON

Experiment	Validation accuracy	Test accuracy	Optimization algorithm
1	64.25	59.76	SGD
2	76.5	74.11	AdaGrad
3	87.1	93.11	AdaDelta
4	95	96.11	AdaDelta
5	**98**	**97.22**	**SGD**
6	97	98.33	SGD

6) Experiment 6: We used the same augmented dataset with GoogleNet as the model, which is the implementation of Szegedy et al. [6]. We obtained a validation accuracy of 97% and a test accuracy of 98.33%. This model was able to classify 12 images per second on the GPU and 0.25 images per second on the CPU.

Table 1 provides an overall comparison of the violation/no-violation classification experiments.

We choose AlexNet over GoogleNet because of its higher speed. We would recommend our HelmetNet from Experiment 3 or Experiment 4 as the best solution for CPU mode because they are able to process 4.2 and 2.17 images per second on the CPU, respectively. They have a fewer parameters compared to the bigger AlexNet and GoogleNet models.

B. Experiments for character classification

In this section, we describe the use of CNNs for Thailand license plate character and digit classification. The model's output consists of 48 classes, 10 classes for digits (0-9) and 38 classes for characters (ก – ฮ without six unused characters).

We performed a series of experiments to find the best model for motorcycle license plate characters. The training set consists of 39,497 images, and the test set consists of 3105 images.

1) Experiment 1: We started with a simple model comprising one convolutional layer, one ReLU layer, one pooling layer, and one fully connected layer. We resized the input images to 25×25. We used 3×3 convolutional filters and 2×2 max pooling. We obtained a test accuracy of 96.94% with the AdaGrad optimization algorithm and a learning rate of 0.01.

2) Experiment 2: We increased the complexity of the network to two convolutional layers, two ReLU layers, and

one fully connected layer. This model achieved a test accuracy of 97.26%.

3) Experiment 3: We added one more fully-connected layer to the model from Experiment 2. This model achieved a test accuracy of 97.36%.

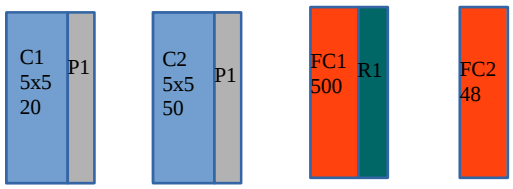

Fig 3. LeNet architecture.

4) Experiment 4: In this experiment, we trained the reference model LeNet [7]. We used 28×28 gray-scale images as the input. We obtained a test accuracy of 97.52%. We also tried different input image sizes. The same network with a gray-scale input image size of 20×20 gave a test accuracy of 97.36%. The same network with an input image size of 24×24 gave a test accuracy of 97.75%. We choose this network as our character classifier.

Fig. 3 shows the structure of the LeNet model provided by Caffe.

These results are with manually segmented characters. To determine the trained model's performance on automatically segmented characters, we used an existing character segmentation algorithm based on binarization and connected component analysis, and compared the LeNet model with HOG-SVM. On a test set of 537 motorcycle license plates, the HOG-SVM gave an accuracy of 46.65%, and the CNN model gave an 51.48% accuracy. The accuracy per plate, where each plate contain 4-7 characters. The low accuracy is mainly due to weak segmentation, which is correct on only 66.91% correct over 537 images. We plan to further improve the segmentation accuracy in future work.

We measured runtime performance of these segmentation and character classifiers on Core-i7 with 8GB RAM and a GeForce 940MX GPU with 2GB RAM. The HOG-SVM model, which runs in CPU mode only, took 10.57 ms per license plate including segmentation. The CNN model took 8.92 ms per plate on the CPU and 8.08 ms on the GPU including segmentation. Speed could be improved with a faster GP, but the CPU mode speed is already fast enough for most applications.

IV. Discussion and Conclusion

We have deployed the system at one intersection in Bangkok (Din Daeng) and five intersections in Phuket. The Royal Thai Police have begun issuing tickets based on the output of the image processing system. Anecdotal reports indicate that helmet compliance is improving at these sites.

From these experiments on CNNs, we can conclude that depth of CNN matters more than the width and the small models can be effective even without GPUs.

V. Recommendation

Although we obtain good accuracy for violation/no-violation classification, we observe errors when a motorcyclist is wearing a hat, when the front rider is wearing a helmet and the second person is not. This issue can be solved by increasing the training data belonging to these classes. i.e., more data containing hats, and more data where the first rider is wearing a helmet and the second is not. The accuracy may be further increased by using recent CNN models like deep residual networks [11] or densely connected convolutional networks [12]. For character classification, the accuracy can be improved by increasing the number of characters from the classes with fewer examples in the existing training data and using bigger networks. The system can be more robust if we skip the segmentation step, as it introduces errors that cannot be corrected by the classifier.

VI. Acknowledgments

This research was supported by the Thailand Research Fund, the Safer Road Foundation of the U.K., and by a graduate fellowship from the Asian Institute of Technology (AIT) to Dharma Raj KC.

References

[1] World Health Organization, "GHO by category road traffic deaths data by country", 2013, http://apps.who.int/gho/data/node.main.A997/.

[2] O. Russakovsky et al., "ImageNet large scale visual recognition challenge", International Journal of Computer Vision (IJCV), Dec 2015, 115(3), 211–252.

[3] M. Jaderberg, K. Simonyan, A. Vedaldi and A. Zisserman, "Reading text in the wild with convolutional neural networks", International Journal of Computer Vision (IJCV), 2016, 116(1), 1–20.

[4] Y. Jia et al., "Caffe: Convolutional architecture for fast feature embedding", In Proceedings of the ACM International Conference on Multimedia, 2014, pp. 675–678.

[5] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", In Advances in Neural Information Processing Systems (NIPS), 2012, pp. 1097–1105.

[6] C. Szegedy et al., "Going deeper with convolutions", 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 1-9.

[7] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition", in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov 1998.

[8] R. Silva, K. Aires and R. Veras, "Helmet detection on motorcyclists using image descriptors and classifiers", In Graphics, Patterns and Images (SIBGRAPI), 2014 27th August, SIBGRAPI Conference on (pp. 141-148), IEEE.

[9] R.C. Gonzalez and J. A. Herrera, "Apparatus for reading a license plate", US Patent 4,817, March 28, 1989.

[10] H. Li and C. Shen, "Reading car license plates using deep convolutional neural networks and LSTMs", arXiv preprint arXiv:1601.05610, 2016.

[11] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition", In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.

[12] G. Huang, Z. Liu, K. Q. Weinberger, L. Van der Maaten and K.Q. Weiberger, "Densely connected convolutional networks", In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.