

Detection and classification of road signs in raining condition with limited dataset

Mohamed Jamil Khan Mohamed Bysul Khan

University of Malaya

Noraisyah Mohamed Shah (✉ noraisyah@um.edu.my)

University of Malaya

Norrima Mokhtar

University of Malaya

Research Article

Keywords: Road sign recognition (RSR), Policy Augmentation, Automatic White Balance (AWB), Image-to-Image-translation (I2IT), MobileNetV2

Posted Date: September 22nd, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-2071755/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Abstract

The road sign recognition (RSR) system is used to complete two tasks: localizing the traffic sign in an image and then classifying it according to the image features. Some of the applications of such system are incorporated into the Advanced Driver Assistance System (ADAS) and autonomous vehicles. However, the accuracy of the model decreases when changes in lighting or weather occurs, and the lack of training samples taken in rainy condition causes the model to be sub-optimal. The research was conducted with the focus on solving two problems, the accuracy of an object detection model decreases when changes in lighting or weather occurs and the lack of training data, especially images taken under adverse condition. In this work, we compare and analyze three methods; automatic white balance (AWB), policy augmentations and Image-to-Image-translation (I2IT) technique on their performance to detect traffic signs in raining conditions. All methods were built upon the pre-trained SSD-MobileNetV2 model and using the TensorFlow2 framework. The images from the Malaysia Traffic Sign Dataset (MTSD) are used to train all models. Finally based on the result of the three model a final combination model is proposed that achieved the best performance in rainy condition. Experimental results showed that AWB was not that effective in detecting road sign in raining condition, while the other two techniques were highly effective. The final proposed model was implemented by combining policy augmentation and I2IT, obtained an mAP of 0.7967 in clear images and mAP of 0.7160 when rainy images was added to the testing dataset. These corresponds to mAP at 50% IoU of 0.8921mAP@0.5 during clear weather and 0.8340mAP@0.5 in raining images, which outperformed other models. Thus, a road sign detection and classification system that can perform well in rainy condition with limited training dataset has been successfully developed.

1 Introduction

Over the past decade, many researchers have attempted to develop automated traffic sign detection systems to advance the technology. These kind of detection systems have become increasingly complex to achieve higher efficiency and robustness. Recently many papers have proposed the use of machine learning to deal with this problem. For instances, [1] used support vector machine (SVM), [2] used convolutional neural network (CNN), [3] experimented with You Only Look Once (YOLO), and [4] implemented artificial neural network (ANN) to perform the task.

To tackle the specific problem mentioned in the previous part, a few researchers proposed some innovative techniques. Firstly, Zheng et. al. [5] found that the image detection model performs better when the image undergoes an auto exposure or automatic white balance (AWB) filter in the pre-processing stage. As the detection model is trained using images that were taken in perfect lighting condition, altering the lighting of input image using a filter to generate an image that looks as if it is taken under canonical light resulted in a higher detection accuracy. [5] proposed to use data augmentation technique with their learned policies to tackle the issue. They found that using that specific augmentation policy had performance marginally better than just using random augmentation. Besides, the best

augmentation policy obtained from the COCO dataset, when used in PASCAL-VOC had better improvement in the detection accuracy as compared to random augmentation.

Finally, [7] found that using image-to-image translation (I2IT) technique to generate training images in rainy condition improved the model's detection accuracy. What this technique essentially does is that, it creates artificial rain on the images taken in clear condition so that when a model is trained on this images, it would learn the features necessary to detect the object in this adverse condition even when the original dataset did not have such examples.

2 Research Methodology

As a first step traffic road sign image dataset along with its respective labels were extracted from the Malaysian Traffic Sign Dataset (MTSD) [8]. From the dataset, only five classes with the highest number of images were chosen: 'No Stopping', 'No Entry', 'Hump', 'Traffic Light', and 'Give Way'. The pre-processing stage consist of random cropping around the traffic sign and downscaling. Subsequently, classification of the five image classes using the pre-trained model, SSD-MobileNetV2 was tested in four different configurations: (i) the baseline without any modification, (ii) with automatic white balance (AWB) as proposed by [5] and (iii) with policy augmentation incorporated as proposed [6] and (iv) with image-to-image translation (I2IT) as proposed by [7], in its training dataset.

The trained model was tested in two conditions, firstly with only clear images in the testing dataset and secondly with clear and some rainy images (obtained manually from YouTube videos) added into the testing dataset. All the three techniques were first implemented and tested separately to obtain their respective optimal parameter. Performance metrics such as the mean average precision (mAP), average recall (AR) and training image generation time was used to undermine the effectiveness of each method.

2.1 Baseline model implementation

The training for the baseline model begins by using all the bounding boxes provided by MTSD, however both the mAP and AR values are low due to the very small bounding box relative to the size of the training images. Zhu et. al. stated that training a model to detect small bounding box would require a huge dataset with each image having a very high resolution [9]. However, the requirement of huge dataset, is not possible in our case due to the limitation of MTSD and the training time limit set by Google Colab. Thus, bounding boxes with area less than 5% of the image are removed from the dataset, resulting with the removal of around 70 bounding boxes from the initial dataset to facilitate the training. The final number of bounding boxes per class is shown in Table 1 below

Table 1

Number of remaining bounding boxes for each class after the removal of small bounding boxes

Class	Number of bounding boxes
No entry	38
No stop	47
Give way	52
'Bumper' (Hump)	67
Traffic light	53
After the removal, the baseline model is trained for 2500 steps after which the loss started to stagnate.	

2.2 AWB implementation

For AWB, histogram clipping with 5%, 10% and 20% clipping percentage as well as Contrast Limited Adaptive Histogram Equalization (CLAHE) were performed on all training images. Histogram clipping is done by first calculating the cumulative distribution function from the histogram. Then, a transformation function which uses the histogram clipping threshold percentage as input to produce a new image which has a flat histogram and a linearized cumulative distribution function (CDF) across its value range. Based on the CDF, the transformation function reduces the magnitude of the grayscales that are higher than the threshold and vice versa. As a result, the final image will have a more balanced histogram and the CDF of the new image will be more linear. The effectiveness of clipping threshold of 5%, 10% and 20% were evaluated to find the optimal value.

The Contrast Limited Adaptive Histogram Equalization (CLAHE) technique extends the idea of histogram clipping but instead of applying the equalization to the histogram of the whole image at once, the histogram of smaller blocks of the image called 'tiles' were equalized separately. To execute this, the RGB image array must be converted into LAB format which consists of L (lightness) channel, and A, B which is a dual color combination channel. CLAHE is applied only to the L-channel and takes in the clipping limit and tile grid size as its parameters. The result of AWB implementation is given in Fig. 1 below.

2.3 Policy augmentation implementation

Another technique implemented onto the pre-trained model is data augmentation. This technique will make the model more robust to changes such as tilting of sign, fading of color, and lighting. AutoAugment library developed by a group of researchers from Google was used to apply the transformation and generate the augmented images [10]. In this method, the three policies proposed along with its optimal parameters obtained by [6] were considered, namely, ImageNetPolicy, CIFAR10Policy and SVHNPolicy. The algorithm was implemented using the AutoAugment library provided by DeepVoltaire on GitHub [11]. The following Fig. 2 shows the sample results after each respective augmentation was applied.

Some of the augmentations such as the shearing and translation altered the position of the road sign in the augmented image, thus, the BBAug library in Python was used [12], to update the bounding box coordinate in the label file accordingly.

2.4 Image-to-image translation (I2IT) implementation

Finally, for I2IT, three cases were tested, namely, (i) with rain augmentation only, (ii) with rain and fog augmentation, and (iii) with rain and darken augmentation. To implement Image-to-Image Translation (I2IT) technique into the training dataset of the model, Auto Mold library in Python was used. It is a tool used to augment images to introduce various real-world scenarios that could reduce the accuracy of the model when testing. This include adding random shadow on the ground, apply random brightness to the images, add rain streaks, add fogs, add solar flares, and add motion blur to the image.

The following Fig. 3 are examples of images generated using the I2IT technique. It should be noted that the augmentations are shown separately in this figure. However, in the actual testing, the augmentations were combined and referred to as Case 1: Rain Augmentation, Case 2: Rain and Fog Augmentation, and Case 3: Rain and Darken Augmentation.

For this project, different combination of this random transformation was tested, and its performance was evaluated to find the best combination of transformations that would result in the best detection accuracy in the testing dataset.

Once the optimal parameter of each of the three techniques described in 2.2 to 2.4 were obtained, combinations of those techniques were then investigated to study the performance

3 Results And Discussion

In this section, the results obtained using the pre-trained model is discussed. As mentioned previously, a convolutional neural network (CNN) based pre-trained model, SSD MobileNetV2 was used to detect and classify the five classes of traffic sign. Three metrics were used to assess the model: mean average precision, average recall, and training image generation time. Firstly, the baseline model was built, and some initial optimization was made regarding the bounding boxes. Then, automatic white balance technique (AWB) with two different approaches was implemented by changing their respective parameters to obtain the optimal performance for each of them. Policy augmentation and I2IT was implemented together with the combination of those techniques. Up to this stage all experiments were conducted with batch size of 4. Finally, hyperparameter tuning of the training process was done to get the best of the model by testing different training batch sizes.

3.2 Evaluation of the baseline model

The baseline model was tested with both clear and rainy images, with the result given in Table 2. As expected, the mAP is reduced when tested for clear plus rainy images compared to only clear images.

Table 2
Performance evaluation of the baseline model

Model	Image condition	mAP	AR
Baseline	Clear	0.6276	0.6013
	Clear + rainy	0.4721	0.4890

3.3 Evaluation of the model performance with various AWB parameters

Figure 4 below shows the performance of the model when histogram clipping percentage (5%, 10%, 20%) and CLAHE technique was implemented on the model. It was found that 10% histogram clipping gave the best result for this technique and improvement in both the mAP and AR (not shown). However, the model only performed marginally better when compared to the baseline. The image generation time for the different clipping percentage were 9.21, 9.38, 9.99 and 14.37s for histogram clipping percentage of 5%, 10%, 20% and CLAHE respectively.

3.4 Evaluation of model performance for various image augmentation policies

Figure 5 below shows the mAP for the three augmentation policies when tested using clear and rainy image conditions. The CIFAR10 policy provides the best performance for both image condition followed by ImageNet and SVHN. This can be attributed to the fact that the CIFAR10 policies had a more balanced augmentation in terms of physical and colour augmentation. Conversely, ImageNet and SVHN policy focused mainly on colour augmentations. Furthermore, the image generation time for CIFAR10 was relatively low (11.2s) compared to ImageNet (15.9s). Although SVHN had the lowest image generation time (10.20s), the low mAP and AR score made CIFAR10 the best choice.

3.5 Evaluation of model performance for various I2IT augmentations

The object detection and classification model were evaluated when its training images included the images generated using the I2IT technique. A total of three cases were evaluated, namely, Case 1 which used rain augmentation only, Case 2 which used rain and fog augmentation, and Case 3 which used rain and brightness (darkening) augmentation. The purpose of this technique, as mentioned previously, is to generate training images in rainy condition so that the model would be able to learn the features required to detect the road sign in those condition.

Figure 6 shows the mAP result for the different cases. Case 2, which uses rain and fog augmentation achieved the highest performance in which both the mAP and AR for both test image condition had the

highest values. The darkening augmentation, when implemented together with the rain augmentation made the performance even worse when compared to just rain augmentation. The downside of Case 2 is that it took significantly longer time to generate the images compared to Case 1 as generating the fog in the images is computationally heavy. From these results, it can be deduced that rain augmentation in combination with fog augmentation helps the model the most to learn to detect the road signs in rainy condition.

Table 3
Evaluation result for selected methods

Method	Image generation time (s)	Test image condition	mAP	AR
AWB 10%	09.38	Clear	0.6380	0.6273
		Clear + Rainy	0.4963	0.4692
CIFAR10	11.20	Clear	0.7904	0.7547
		Clear + Rainy	0.6664	0.6581
I2IT Case 2: rain + fog	18.86	Clear	0.7545	0.7287
		Clear + Rainy	0.6907	0.6920

Table 3 gives the result for the mAP, AR and image generating time for selected best result of different methods, for comparison.

3.6 Evaluation of model performance with combination of the best techniques

In this part, the previous three techniques were combined to observe their effectiveness when implemented together. Three cases were developed to study these. Combination Case 1 (CC1) is when using CIFAR10 with the rain and fog augmentation (I2IT). Combination Case 2 (CC2) is CIFAR10 + rain + fog + AWB (10%). Finally, Combination Case 3 (CC3) is CIFAR10 + rain + fog + CLAHE.

Similar performance metrics as previous parts were recorded and tabulated in Fig. 7 below. Both AWB and CLAHE does not improve the performance of the model when used in combination with CIFAR10 and I2IT of rain and fog augmentation. In fact, it worsened the performance (in both mAP and AR). This can be deduced that as the model is already familiar with the features of the rainy condition due to the augmentations in the training process, correcting the lighting to make the rainy image look as if it is taken under canonical lighting is unnecessary. Thus, from the study in this part, it can be concluded that AWB and CLAHE does not need to be implemented when the model has already been trained on the policy augmented images as well as the I2IT augmented images.

In the next investigation the training batch size of 4, 8 and 16 is tested and the resulting performance metrics were evaluated to find the optimal value.

Batch sizes larger than 16 was not tested due to the hardware limitation, whereby an “insufficient memory” error was encountered. The training time also increased significantly as the batch size was increased. The best model obtained from the previous evaluations (CIFAR10 + rain + fog I2IT) were used for this study. Table 4 below shows the performance metrics for different batch sizes. training batch size of 8 provided the best performance in terms of mAP and AR albeit taking just slightly longer training time compared to batch size of 4. Batch size of 16, however took much longer time (almost 3 times longer) and the increase in size is very minimal and, in some cases, (such as when tested with rainy images) had worse performance.

Table 4
Evaluation result for different batch sizes

Training Batch size	Training time	Test image condition	mAP	AR
4	00:42:40	Clear	0.7768	0.7427
		Clear + Rainy	0.6842	0.6777
8	01:00:11	Clear	0.7967	0.7627
		Clear + Rainy	0.7190	0.6995
16	02:52:48	Clear	0.8114	0.7760
		Clear + Rainy	0.6841	0.6717

3.6 Further evaluation of the final model

The purpose of this part is to provide a clearer picture of the performance improvement gained from implementing each of the technique. Only the mean average precision (mAP) metric will be considered in this part as both the mAP and average recall (AR) of all the models were close in value. Figure 8(a) shows the mAP of the models when tested against images taken in clear condition only. Figure 8(b) on the other hand, shows the mAP when both the clear and rainy condition images were used. Figure 8(c) gives the image generation time for various techniques used.

From the Fig. 8(a), the limitation of the MobileNetV2 in detecting small bounding boxes can be seen clearly. There is a big jump in performance (28.9% mAP) when the smaller bounding boxes were removed from the dataset. This is counter-intuitive, considering that the traditional assumption that more images available in a dataset would produce better results. This also agrees with the findings obtained by researchers in University of Bucharest, Romania that one of the limitations possessed by MobileNetV2 is that it is unable to detect small bounding boxes as the boxes itself contain little information for the model to recognize it [13].

CLAHE, when applied to the testing set, only achieved very minimal performance improvement (less than 1%) while histogram clipping achieved 1–2% improvement. The real jump in performance improvement was achieved when implementing the policy augmentations and I2IT into the models. The combination

of CIFAR10 and I2IT (rain and fog) with training batch size of 8 has the best performance in both image conditions. When compared to the baseline (excluding small bounding boxes), it had achieved 16.9% and 24.7% mAP increase in clear and rainy conditions, respectively.

Another observation from Fig. 8 above is that in clear condition, policy augmentation achieved higher mAP compared to I2IT. Whereas, when rainy images was added to the testing set, I2IT had higher mAP compared to policy augmentation. This highlights the ability of I2IT that targets specifically performance improvement in rainy conditions as opposed to policy augmentation which is a more of a general performance improvement. When combining these two techniques, the mAP achieved by the model had a good tradeoff and balance in detecting images in both clear and rainy images.

The training image generation time for the various techniques was also investigated. Histogram clipping had the lowest generation time (9.38s) while CLAHE took longer time (14.37s). This can be attributed to the way both algorithm works, histogram clipping applies the histogram linearization to the whole image at once, while CLAHE applies it to multiple portions of the image separately. CIFAR10 took only 11.2s to generate the images while I2IT, which involves generating rain streaks and fog particles in the images took 18.86s. The main reason for I2IT taking much longer time is due to the generation of fog particles which is computationally heavier as compared to traditional image color augmentation that is performed by policy augmentations. The computation time, when combining both of these methods is just the sum of the time taken for each of the methods separately (30.06s) as it is implemented together. The generation time stated is relatively low as the size of dataset used is relatively small. The generation time would scale linearly as the size of image dataset increases.

Table 5 Output comparison between the baseline and the best model



To visualize and compare the model output between the baseline and the obtained best model, the test images along with its bounding boxes are shown in Table 5. From the images, the best model can detect the road signs more effectively in both clear and rainy condition compared to the baseline model.

The overall mAP of the best model, obtained by this thesis in clear condition is 0.7967 and 0.7190 in rainy condition. These corresponds to mAP at 50% IoU of 0.8921mAP@0.5 and 0.8340mAP@0.5 respectively. In contrast to this, the model developed by Mohd-Isa, et al (2020), reported in [14], which was trained on the same MTSD dataset only achieved 0.825mAP@0.5 in clear condition. It can be clearly seen that our model along with the techniques implemented in this thesis, has outperformed the previous technique in clear condition. The performance of our model even when tested in rainy condition images had higher mAP@0.5 than the result obtained by [14] that was only tested in clear condition.

4 Conclusion

In conclusion, the preprocessing of the MTSD dataset has been successfully completed and is ready to be used for training purposes. Some fine adjustment on the dataset such as checking the labels of each images, extracting the five classes from the whole dataset, and removing relatively small bounding boxes to facilitate the training were done.

AWB technique on its own was found to increase the performance of the model slightly. Besides, policy augmentation and I2IT technique was much effective in increasing the model performance in clear and rainy condition. It was also found that the AWB technique only had very minimal impact on the performance when combined with the other two techniques. On the other hand, the combination of policy augmentation and I2IT was found to give the best performance in both clear and rainy condition. Finally, the hyperparameter tuning of the best model was performed to get the final best model. The final model achieved an mAP of 0.7967 in clear condition and 0.7190 in rainy condition.

Declarations

Ethical Approval Not applicable

Competing interests No competing interest.

Author's contributions Noraisyah M.S. and Mohamad Jamil Khan M.B.K wrote the manuscript and Norimma M prepared the figures. All authors reviewed the manuscript.

Funding This research was not funded.

Availability of data and materials The image database and algorithms used in this research are listed in the reference list

References

1. T. Guofeng, C. Huairong, L. Yong and Z. Kai, "Traffic sign recognition based on SVM and convolutional neural network," *2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2017, pp. 2066-2071, doi: 10.1109/ICIEA.2017.8283178.
2. Palak and A. L. Sangal, "Traffic Signs Classification using Convolutional Neural Networks: A review," *2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC)*, 2021, pp. 450-455, doi: 10.1109/ICSCCC51823.2021.9478172
3. W. Yang and W. Zhang, "Real-time Traffic Signs Detection Based on YOLO Network Model," *2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, 2020, pp. 354-357, doi: 10.1109/CyberC49757.2020.00066.
4. Islam, Kh Tohidul, and Ram Gopal Raj. "Real-Time (Vision-Based) Road Sign Recognition Using an Artificial Neural Network." *Sensors (Basel, Switzerland)* vol. 17,4 853. 13 Apr. 2017, doi:10.3390/s17040853

5. Z. Zheng, H. Zhang, B. Wang and Z. Gao, "Robust traffic sign recognition and tracking for Advanced Driver Assistance Systems," 2012 15th International IEEE Conference on Intelligent Transportation Systems, 2012, pp. 704-709, doi: 10.1109/ITSC.2012.6338799.
6. Zoph, Barret, Ekin Dogus Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens and Quoc V. Le. "Learning Data Augmentation Strategies for Object Detection." ECCV (2020), doi: 10.48550/arXiv.1906.11172
7. M. Hnewa and H. Radha, "Object Detection Under Rainy Conditions for Autonomous Vehicles: A Review of State-of-the-Art and Emerging Techniques," in IEEE Signal Processing Magazine, vol. 38, no. 1, pp. 53-67, Jan. 2021, doi: 10.1109/MSP.2020.2984801.
8. Madam, Ahmed and Rubiyah Yusof. "Malaysian traffic sign dataset for traffic sign detection and recognition systems." Journal of Telecommunication, Electronic and Computer Engineering 8 (2016): 137-143
9. Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li and S. Hu, "Traffic-Sign Detection and Classification in the Wild," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2110-2118, doi: 10.1109/CVPR.2016.232
10. E. D. Cubuk, B. Zoph, D. Mané, V. Vasudevan and Q. V. Le, "AutoAugment: Learning Augmentation Strategies From Data," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 113-123, doi: 10.1109/CVPR.2019.00020.
11. DeepVoltaire. (23 July, 2021). AutoAugment. Retrieved from GitHub: https://github.com/DeepVoltaire/AutoAugment/blob/master/AutoAugment_Exploration.ipynb
12. Sahota, H. (30 September, 2020). BBAug. Retrieved from GitHub: <https://github.com/harpalsahota/bbaug>
13. P. Soviany and R. T. Ionescu, "Optimizing the Trade-Off between Single-Stage and Two-Stage Deep Object Detectors using Image Difficulty Prediction," 2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2018, pp. 209-214, doi: 10.1109/SYNASC.2018.00041.
14. W. -N. Mohd-Isa, M. -S. Abdullah, M. Sarzil, J. Abdullah, A. Ali and N. Hashim, "Detection of Malaysian Traffic Signs via Modified YOLOv3 Algorithm," 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), 2020, pp. 1-5, doi: 10.1109/ICDABI51230.2020.9325690.

Figures

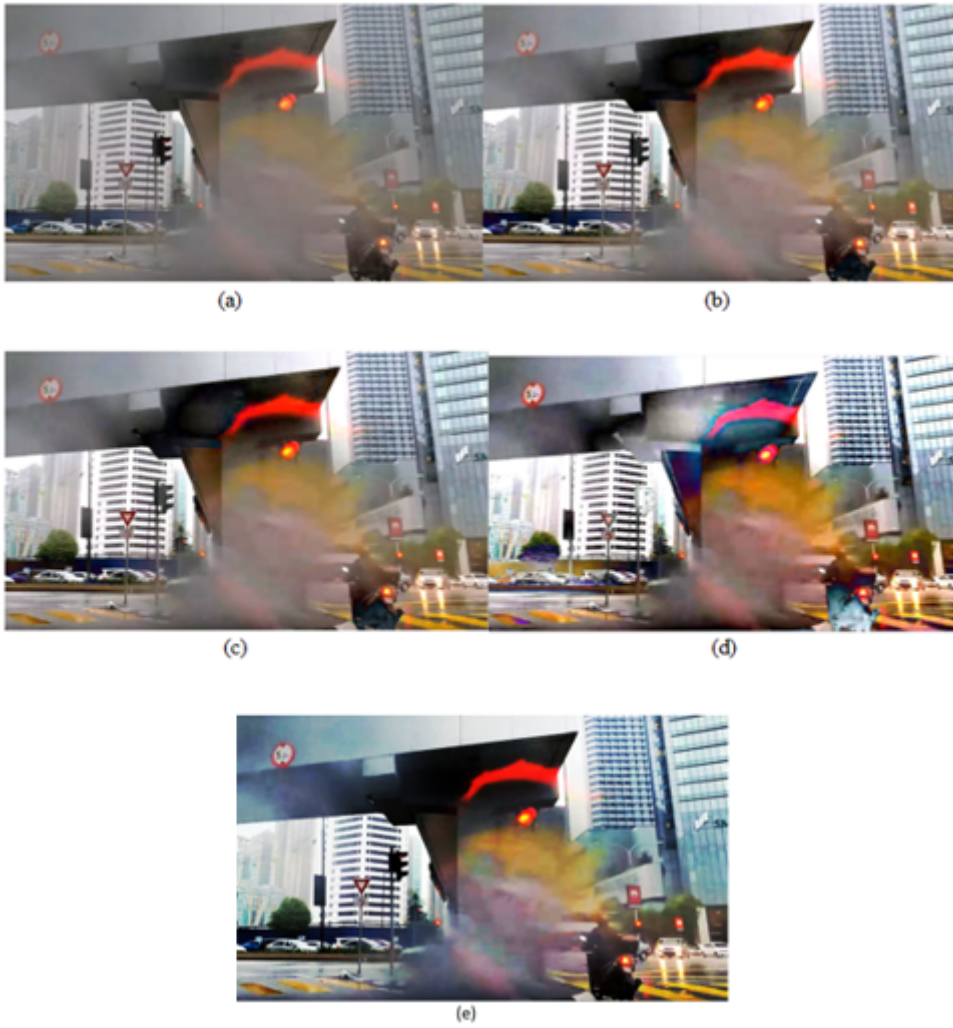


Figure 1

Images with AWB applied with different clipping percentage, (a) Original, (b) 5%, (c) 10%, (d) 20% (e) CLAHE

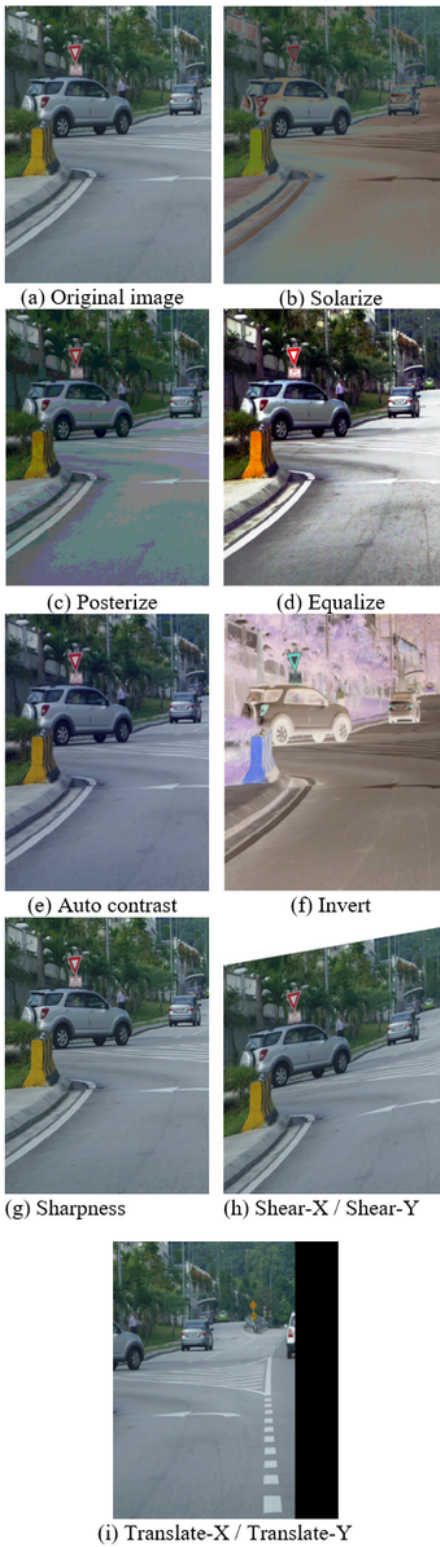


Figure 2

Sample results for various applied image augmentation operation

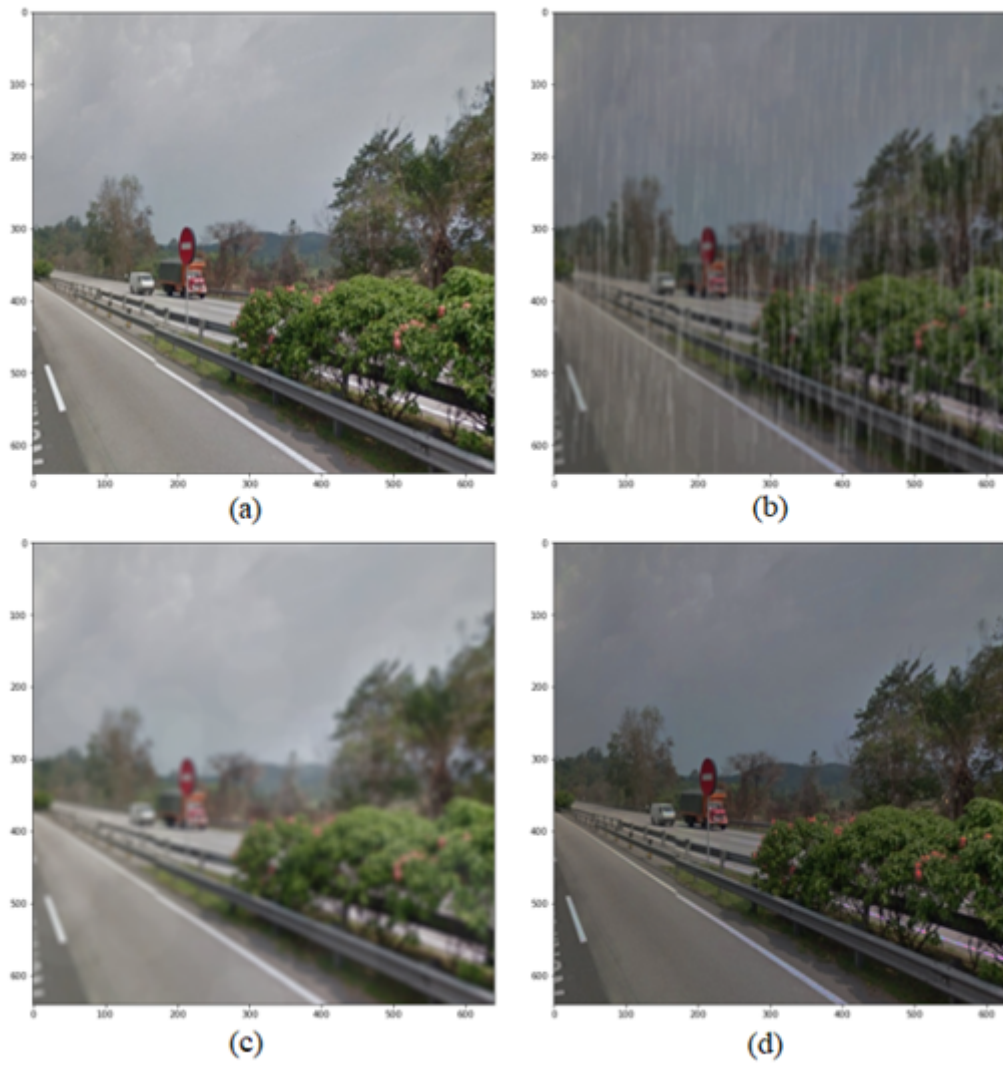


Figure 3

Sample image with various I2IT augmentation applied (a) original (b) rain (c) fog (d) darken

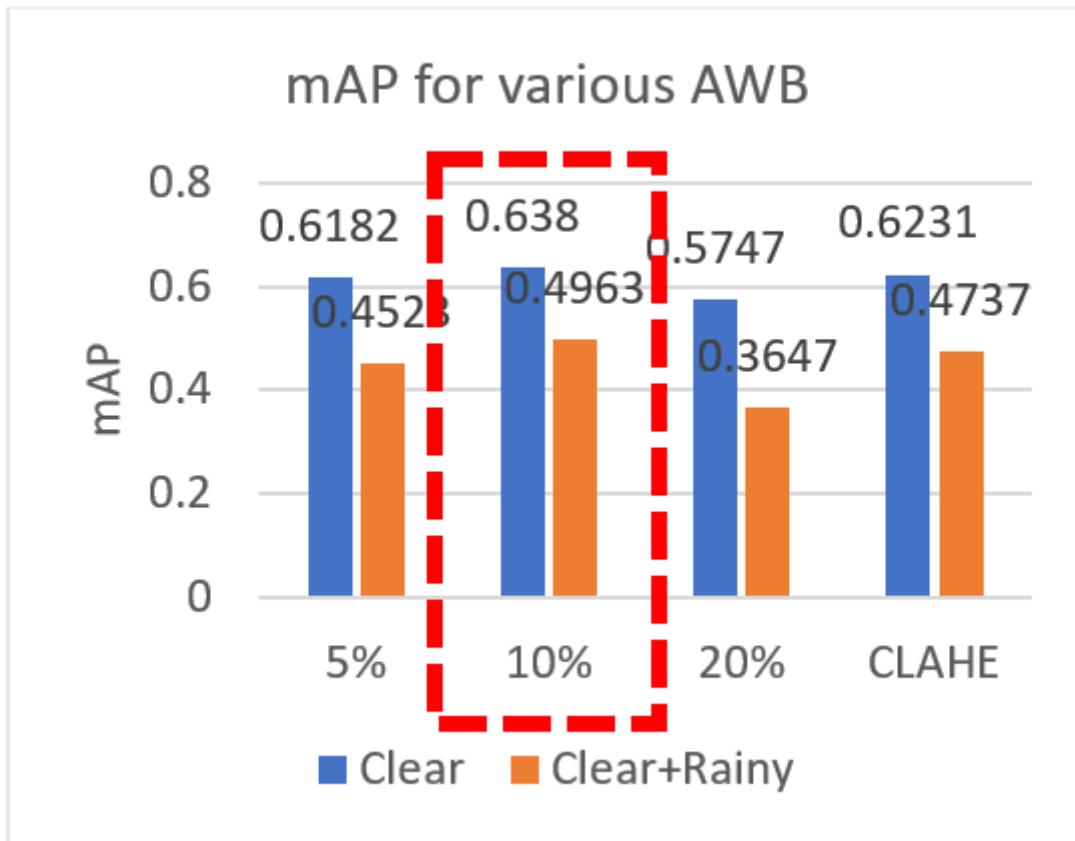


Figure 4

mAP for various AWB applied for clear, and clear plus rainy test images

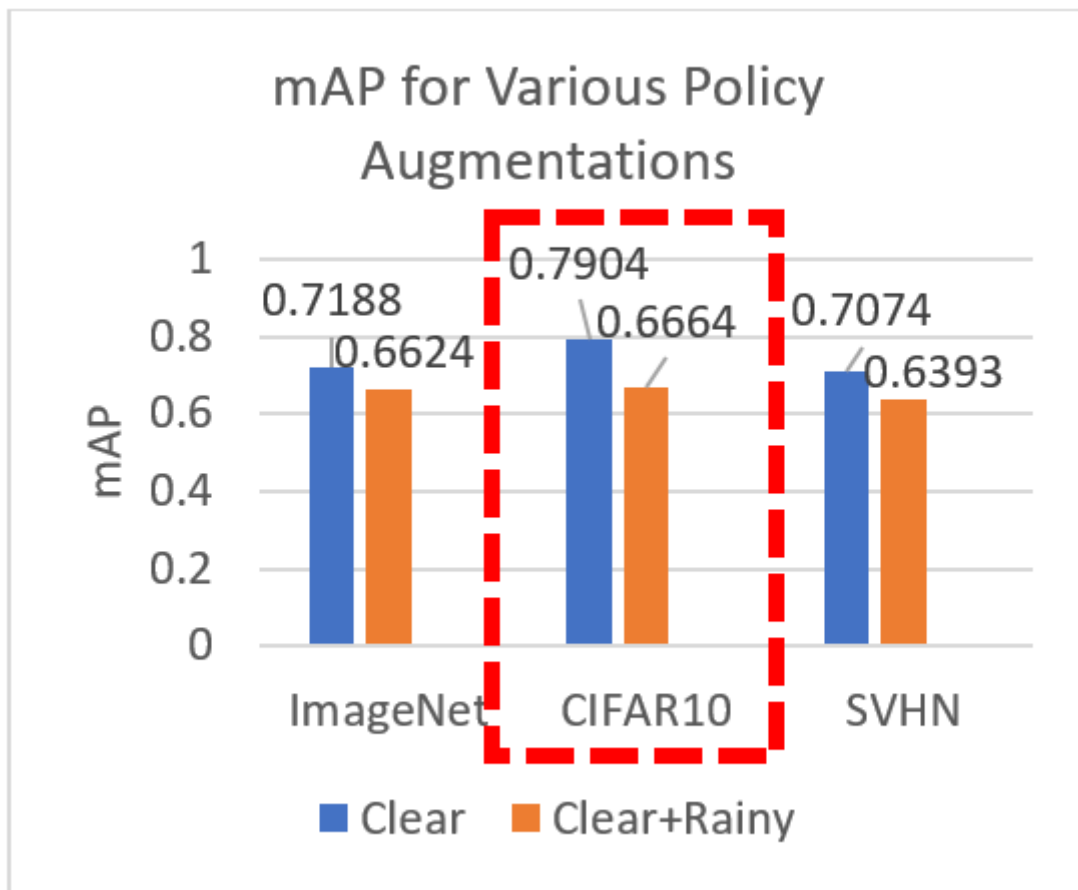


Figure 5

mAP for various policy augmentations applied for clear, and clear plus rainy test images

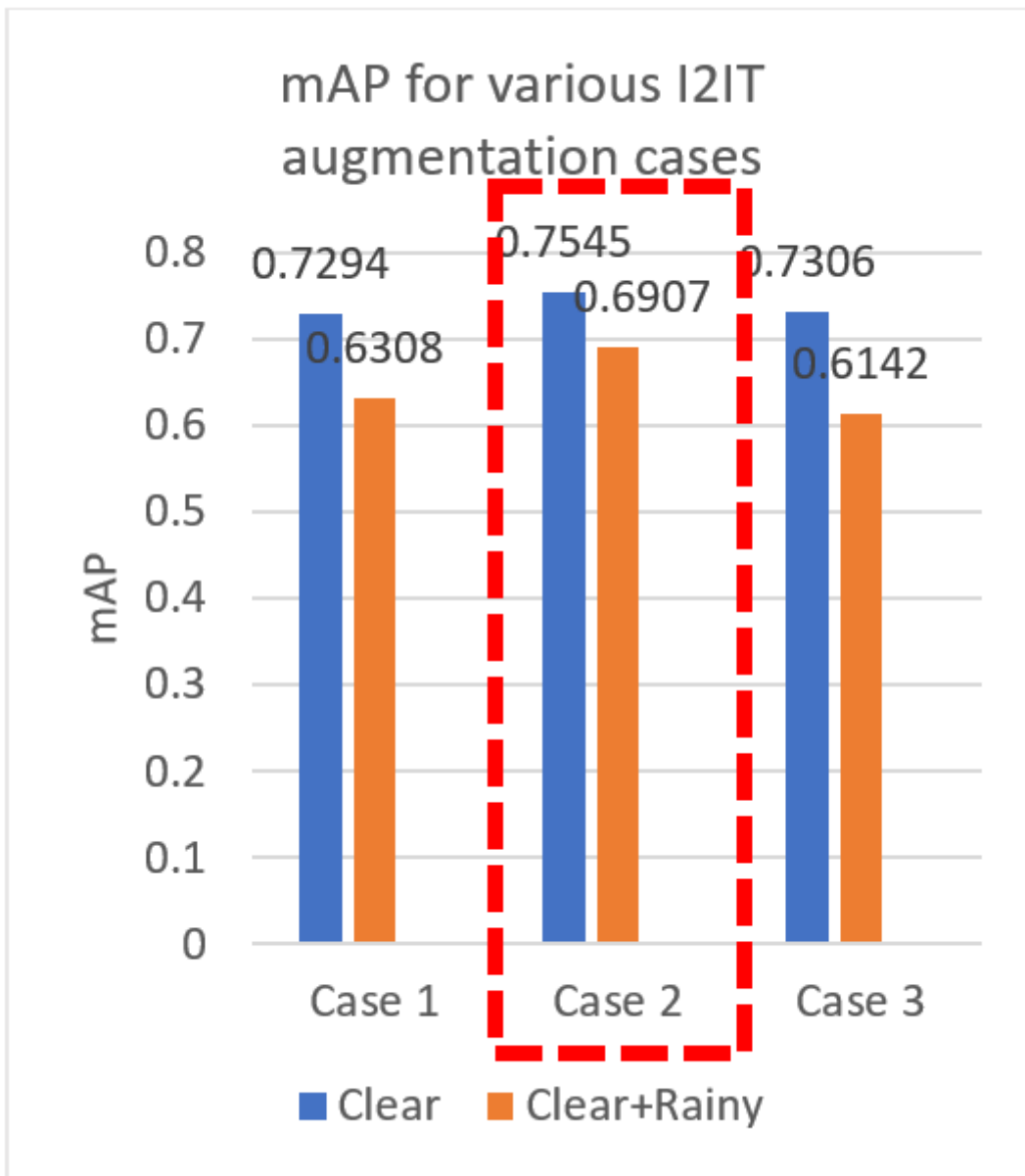


Figure 6

mAP for various I2It augmentations policy applied for clear, and clear plus rainy test images

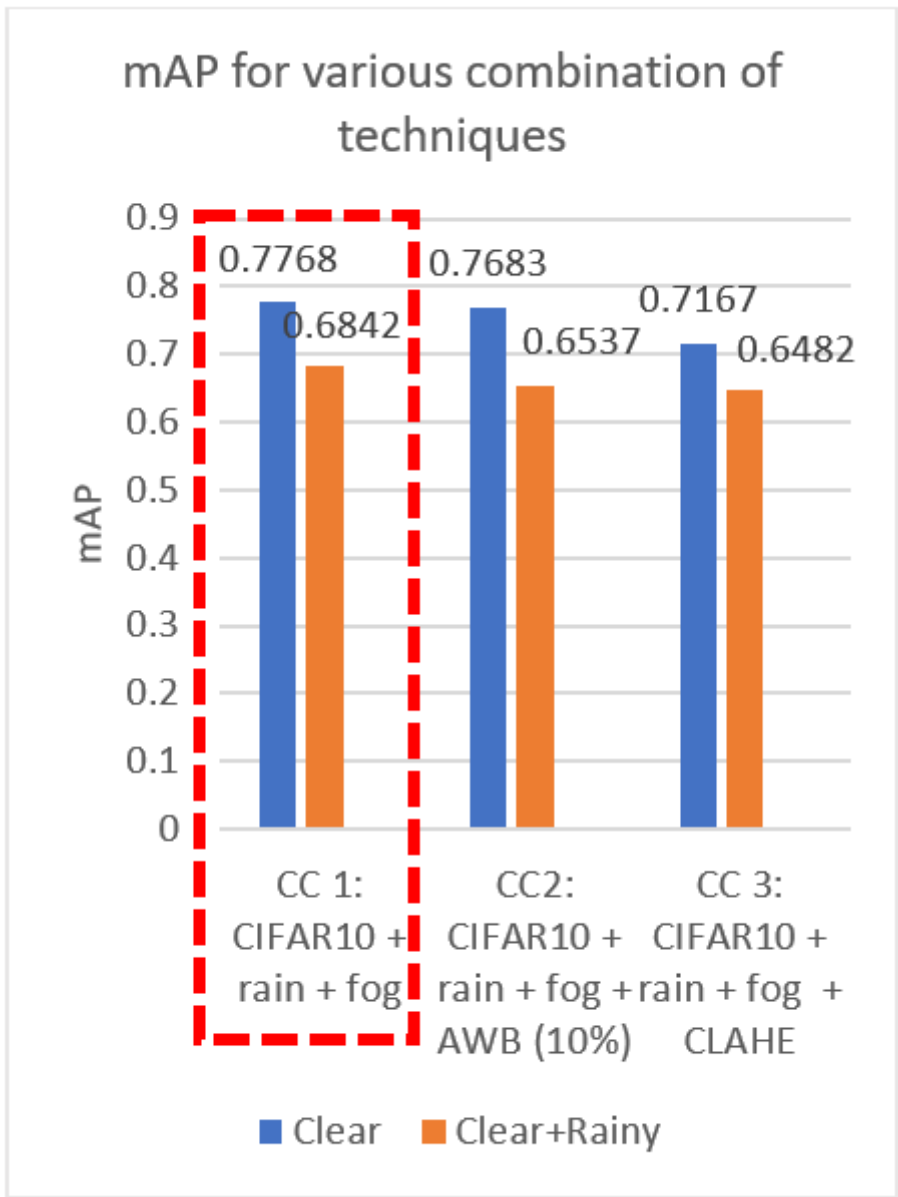
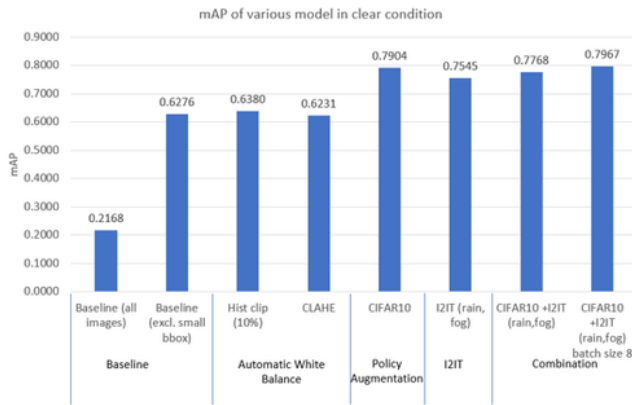
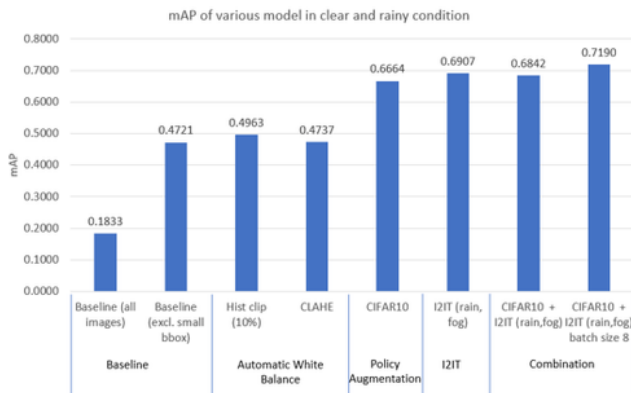


Figure 7

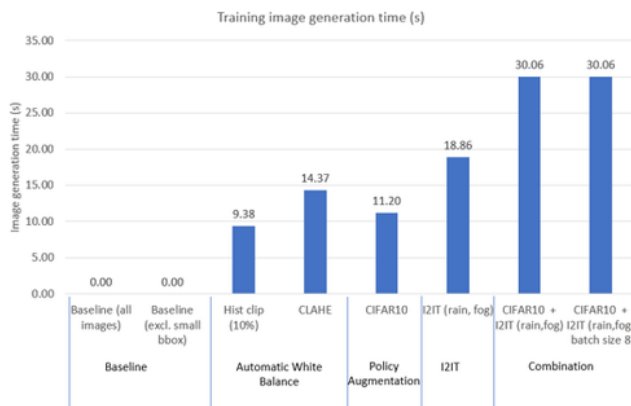
mAP for various combination techniques applied for clear, and clear plus rainy test images



(a) mAP of various models when tested in clear condition images



(b) mAP of various models when tested in clear and rainy condition images



(c) Training image generation time for various techniques

Figure 8

Performance metrics such as (a) mean average precision (mAP), (b) average recall (AR) and (c) training image generation time for various techniques