DATA AUGMENTATION FOR HAAR CASCADE BASED AUTOMOBILE DETECTION

MACIEJ PAWEŁCZYK[1]

Warsaw University of Technology, Institute of Aviation

Summary

The article describes recent object detection methods with their main advantages and drawbacks and shows results of application of machine learning Haar Cascade algorithm for automobile detection.
The article underlines problems related to the feature dataset generation and presents an overview of current dataset augmentation methods such as image mirroring, cropping, rotating, shearing and color modification. New methods fot image dataset augmentation, such as utilization of CAD models and Deep Learning solutions, are also proposed.
In order to ensure low cost, real time detection machine learning based Haar Cascade detector has been proposed and tested on a custom dataset specifically created for dataset augmentation methods evalutation. Article provides all input parameters for detector training process, along with a brief description of object detection metrics.
Finally the article presents results of the baseline detector and augumented calssificator created based on vertical image mirroring technique, for different dataset configurations. Algorithms performance for real time detection on high resolution images was also evaluated.

Keywords: Automobile Detection, Object Detection, Haar Cascade, Data Augmentation

1. Introduction

Traffic cameras are usually low FPS, medium resolution CCTV cameras. While there is a wide range of object tracking methods one of the biggest challenges is object detection itself, which includes object identification and localization on the given picture frame for any size of the object in question. In order to allow real time application of object detection a Single Shot Detector (SSD) has been proposed with an intent of general automobile detection.

Automobile detection has a long history, usually related to highway toll and security based license plate screening. Those systems usually do not work real time and require substantive resources to install and maintain. Furthermore, highway toll fee management is often driven by additional RFID hardware, which has to be installed on forward windshield, reducing customer experience and obstructing line of sight.

[1] Warsaw University of Technology, Faculty of Power and Aeronautical Engineering, Institute of Aviation, Nowowiejska 24, 00-665 Warszawa, Poland, e-mail: mpawelczyk@meil.pw.edu.pl

Vehicle monitoring cost could be severely decreased if it could be handled by low power appliances such as standard CCTV cameras. That in turns requires effective model, which will enable high accuracy even in cases of background clutter and object occlusion. There have been multiple models proposed and used for vehicles (automobile) detection purposes. Two examples of Haar Cascade utilization for vehicle detection are presented in Figure 1 (stationary) and in Figure 2 (for UAV applications).

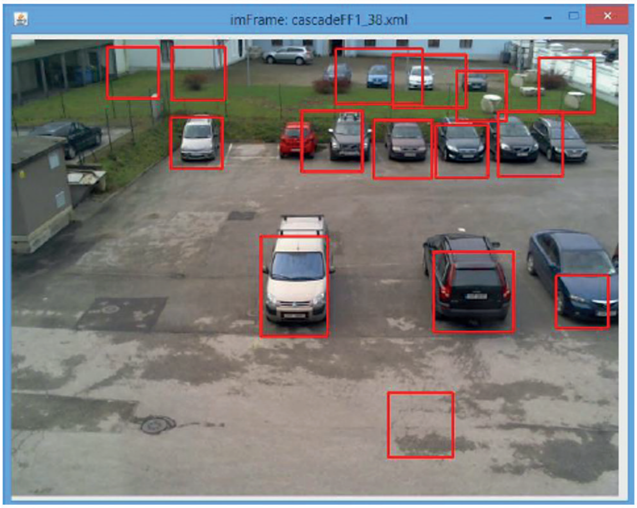

Fig. 1. Haar Cascades can be used for other objects, such as automobiles, which resulted in multiple industry application even considering low algorithm performance, [9]

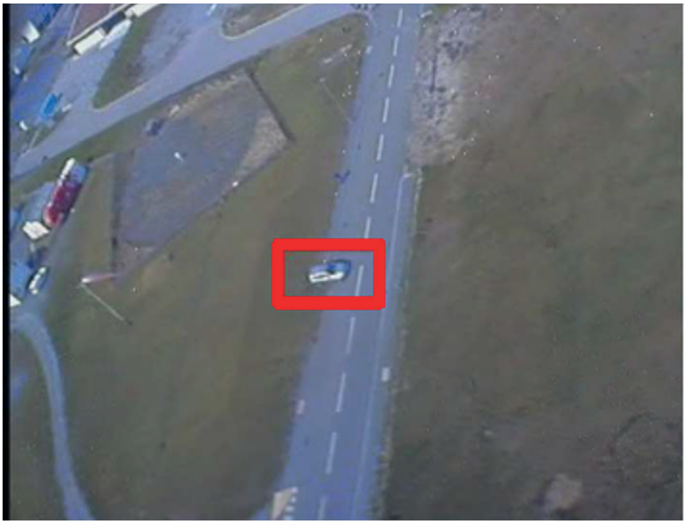

Fig. 2. Similar Haar Cascade applications for vehicle detection were used on UAVs/drones themselves, [2]

Luo-Wei Tsai et al. [10] proposed a detection utilizing color and edges and tested relationships between colors of the road, background and vehicles themselves. Model utilized Bayesian network in order to identify vehicle colors from background and a wavelet feature to process vehicle images. Proposed model shown 89.9% precision and average accuracy of vehicle detection of 94.5%.

Hsu-Yung Cheng et al. [4] presented automatic vehicle detector, which utilized vehicle color extraction, Canny edge detector and dynamic Bayesian network for classification. The system thus utilized pixel wise relationship and allowed up to 92.31 hit rate under different camera angles.

Ekrem Başer et al. [1] have presented results of vehicle detection via Haar Cascade in order to detect unallowed lane change, reaching the accuracy rate of 92%.

The technology enabling use of optical systems in practical applications has been emerging in the last two decades with significant progress made in the area of image recognition. Most typical example of this is a face recognition system, present in civilian appliances such as cameras or cell phones since late 2000s. This system is an example of machine learning Haar Cascade object detection system, proposed in 2001 by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" [11], although the algorithm itself can be traced back to 1973 [5].

The Viola Jones approach is often called "Haar Cascades" as the algorithm utilizes a simple feature set, which is reminiscent of Haar basis functions, which have been also utilized by Papageorgiou et al. [7] which in turn is related to work of Alfréd Haar [6]. The model utilized a user defined number of machine learning decision tree splits, specified number of detection stages and a set of predefined object features, optimized for a specific task. The picture is scanned via growing window frame with some objects quickly discarded due to required threshold being not reached or left for further analysis. If specific area is intersected by predefined number of sliding windows it is deemed an object in question. A such approach allows a high rate of detection with little computational effort and can be used real time in appliances with little computational capability (with the first application in Fuji camera in 2006). The algorithm, due to its hierarchical nature, is in fact very similar to convolutional neural network approach. While the algorithm is easy to apply and can be run in real time on modern hardware it is constrained by a limited number of available features, which hinders its generalization capability, reducing achievable accuracy. However, it is still widely utilized for face detection in cameras and smartphones due to its low computational requirements and general versatility. Haar Cascade OpenCV implementation does also not allow GPU acceleration.

More recent systems are derived from Artificial Neural Networks (ANNs) and are capable of much greater accuracies than Haar Cascades. Modern utilization of ANNs considers multiple hidden layers stacked on top of each other for better generalization effect. This approach, due to high complexity and utilization of multiple hidden layers, is often called Deep Learning. Deep Learning layers can be modified at will, which allows a higher generalization capability and resulting higher accuracy than that of Haar Cascade. The drawback is higher computational requirements, both for runtime and training. While ANNs are supported by GPU acceleration they still require a lot of computational effort for successful cost function minimization. The end result of the training is frozen computational graph, which can only run on GPU accelerated hardware. While the recent introduction of supporting hardware such as neural computing stick [15] can support versatile hardware implementation such as Raspberry Pi 3, processor based models are simply easier to implement.

Furthermore Artificial Neural Network model can be retrained at any moment on new data, which is usually recommended as model generalization is best reflected with the higher training dataset. At the same time Haar Cascade tends to overfit above certain threshold (different for a different application), with training time for new datapoints requiring completely new training runtime. As such Haar Cascade are usually used for lightweight, real time implementation, while ANNs are more likely to be used for offline analysis with almost guaranteed higher performance on the same dataset.

While object detection systems are very useful and can result in high automatization of almost any process, they also require structured, labeled data. In this instance the data will consist of hundreds of images, labeled with pixelwise information on where exactly is the object in question. High diversity of input dataset is critical so that both Haar Cascade and ANN would be able to generalize the problem enough to recognize new vehicle of any shape, color, orientation and background. The task gets even more complicated as not only does the system needs to detect the object in question, but it also has to determine where exactly in the picture it is.

Data acquisition for the image, usually begins with utilization of internet search engines such as Google. This intuition is however broken very fast as most popular phrases such as "car", "vehicle" or similar return marketing product pictures rather than an object in question. In fact typical commercial like picture usually results in overall lower accuracy of the model as the system is unable to recognize the object on road/nature/urban background (as neither is consistently one color, which is usually the case for commercial photographs).

It is critical to remember that with object localization not only does the model requires the image itself, but also an information about pixel box position of the object in question. As this task needs to be performed manually it is highly time consuming, resulting in increased turnaround time and model generation cost. The object detection model will achieve highest accuracy if it is trained on the dataset similar to its target. The dataset generation is a labor consuming task as every picture requires approximately 30 seconds to properly label, which can take weeks in case of machine learning or deep learning based approaches requiring thousands of training data samples. To reduce manual effort required (and corresponding cost) and to expand available dataset, it is possible to utilize other methods called data augmentation.

2. Data Augmentation Methods

The most typical data augmentation method is mirroring. This method assumes creation of vertical reflection of the baseline image as shown in Figure 3. For some objects horizontal reflection is also possible (for example kitchen fork detection), but in this particular application horizontal swap is extremely rare. In theory, this method could easily double the available data set and allows quick and easy modification of the bounding box position. In applications where most of the vehicles should be detected from larger distances, mirror image will not result in dataset expansion, but could in fact instill old features into the model resulting in the model overfitting (abstracting model to fit within provided data rather than generalizing it to fit new, unknown images). This has been shown in Figure 4.

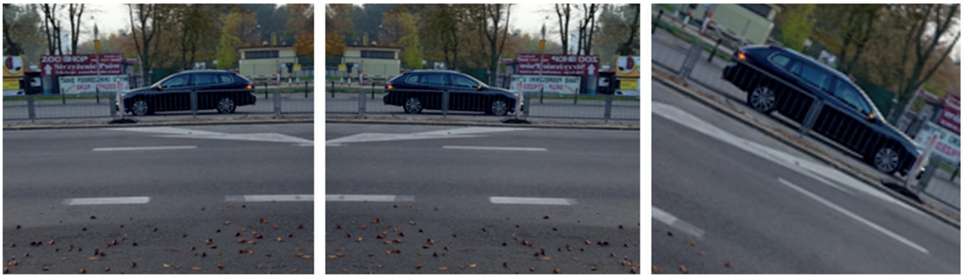

Fig. 3. Object image – baseline, mirrored and rotated/cropped. Author's own materials

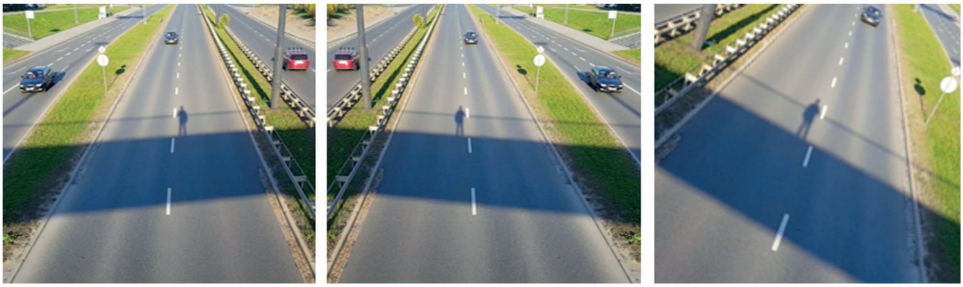

Fig. 4. Long range detection model application – little pixelwise change in object position resulting in little new information provided to the model. Author's own materials

Another data augmentation method is object rotation. The entire picture is rotated, usually around its center point and resulting image discrepancies are cropped to provide a consistent image as shown in Figures 3 and 4. This is a much more effective technique than image mirroring as entire image is changed. The biggest disadvantage of this technique is resolution reduction as cropped image will have lower resolution than its baseline. The level of resolution reduction will depend on rotation angle used. Considering that all pictures are usually down sampled anyway, this will not result in any problems in model training. The more important challenge comes from the bounding box transition. As bounding box has to contain an entire object, it may have completely different definition after object rotation (excluding the obvious fact that it need to be rotated as well).

Random cropping is a technique often used to quickly generate new datapoints. As the name implies, it considers copying input image and randomly cropping it as shown in Figure 5.

The end result is however simple copy of the input, which means that no new feature will be learnt by object detection/localization algorithm and that old features will overfit. At the same time bounding box position change can be easily automated to fit new picture.

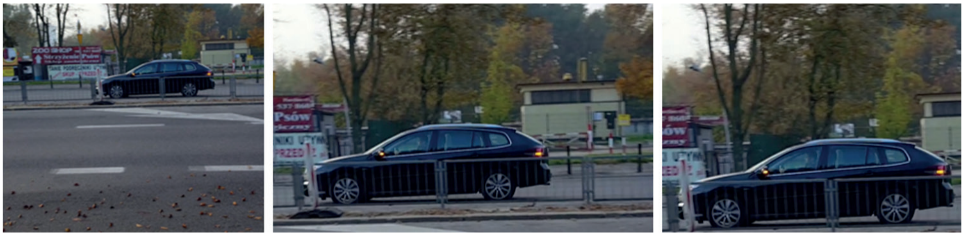

Fig. 5. Random cropping on target object image. Autor's own materials

Shearing applies a trigonometrical 2D transformation on the image, which results in sheared image. Shear angle can be adjusted for better effect. High shear angle results in high distortions, which makes resultant object almost nonrecognizable. The drawback of this method is the fact that sheared image has to be cropped, which results in information loss (resolution reduction). An additional challenge is bounding box transformation to ensure consistency in object localization.

Swirling a non-linear image deformation that creates a whirlpool effect. Whirlpool parameters can be adjusted to control the end effect. Swirl effect is adjustable, which ensures consistency and value added of output image. The bounding box is not affected, while resulting picture can add completely new features to the model. Shearing/swirling effect has been shown in Figure 6.

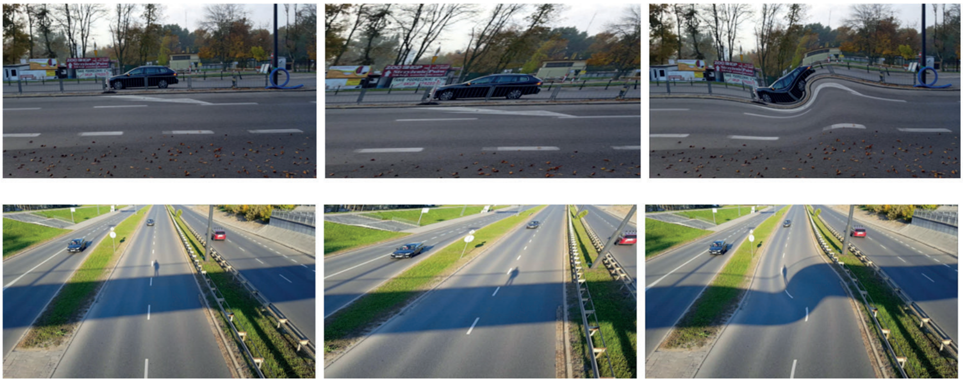

Fig. 6. Object image – baseline, sheared and swirled. Author's materials

Warping is a combination of shearing, swirling, translation, rotation and cropping according to the user requirement. Warping as a combination of methods mentioned above, is usually done locally, which ensures that bounding box remain in known position.

Color shifting/multiplication is a simple technique, which processes pixel wise definition on an image depending on the user input. Usually image is saved in RGB format, where each pixel is represented by a number, usually as an integer from 0 to 255 (recently as float from 0 to 1). Modification of the integer will result in significant changes in the picture as typical RGB to Grayscale conversion will be affected by this change. The end result has been shown in Figure 7.

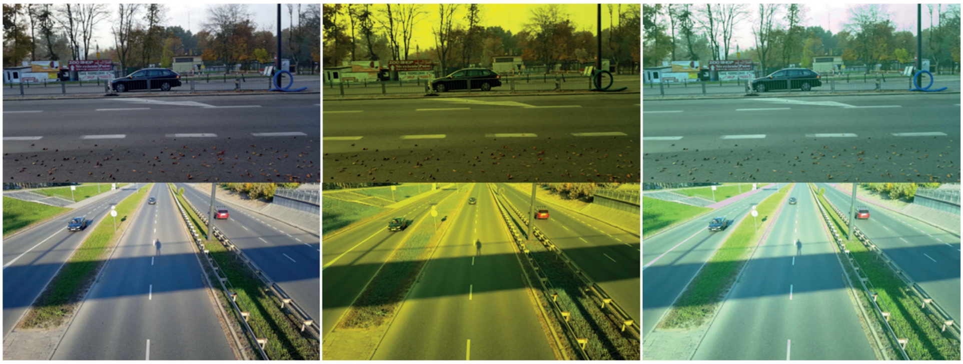

Fig. 7. Object image – baseline, with color multiplication (blue channel removed) and with color shifted ([+10,+50,+25]). Author's materials

Computer generated image utilization is another technique, that is recently gaining a lot of attention. While 3D CAD model takes a lot of time to create the end result can be adjusted parametrically according to designers will. Parameters like object width, length, color, feature size, inclination and background can be easily adjusted creating infinite number of potential datapoints. One of the drawback of this approach is that the images have no noise associated with typical CCTV camera operation (vibration, lightening, haze etc.). Those, however, can be added later on. While each position change of the model will have to be reflected by appropriate bounding box position change entire process can be easily automated to facilitate the dataset generation. Example of automatically generated and rendered 3D CAD model image has been shown in Figure 8, in this example for UAV object detection. All the parameters of the CAD model, such as texture, propeller geometry, object shape and size can be easily adjusted along with its background.

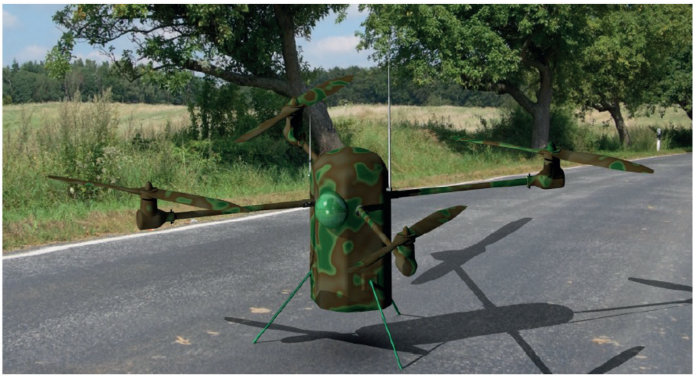

Fig. 8. Computer generated image of quadcopter generated by CAD software (Catia) rendering. Author's materials. [8]

There are also multiple ensemble techniques, which combine solutions mentioned above. One of more popular one is multi crop, which consist of combination of mirroring and cropping. It's typical implementation is called 10-crop, where one input image is cropped and mirrored to create 10 new images as shown in Figure 9.

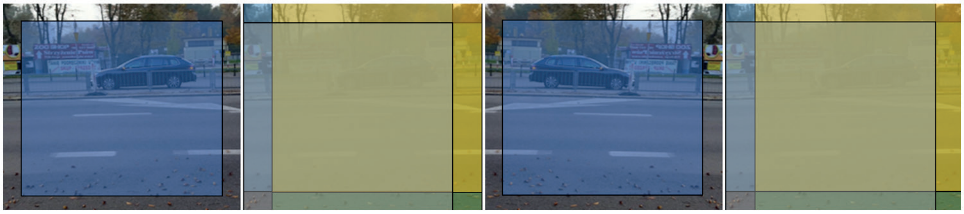

Fig 9. Multi crop applied on object image. From the left – baseline image cropped regularly, baseline image cropped 4 times from the corners, mirrored image cropped regularly, mirrored image cropped 4 times from the corners. Author's materials

While applications mentioned above rely on straightforward pixel modification future data augmentation will heavily rely on convolutional neural networks. These networks can train on real life examples to mimic typical background modification schemes such as day vs night conversion [11]. More detailed expansion of this work can show picture change in reference to Generative Adversarial Network (GAN) style transfer, which can mimic object transfiguration, season transfer, photo enhancement and collection style transfer. Example of GAN application for image-to-image translation has been shown in Figure 10 (season-to-season translation) and in Figure 11 (image foreground extraction).

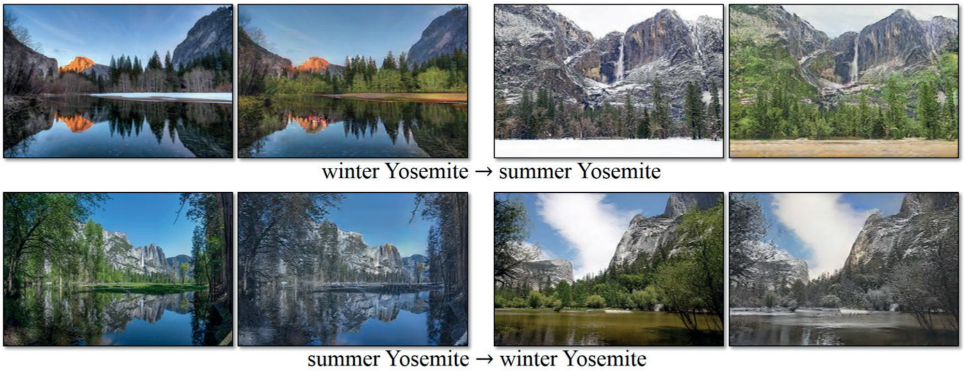
winter Yosemite → summer Yosemite
summer Yosemite → winter Yosemite

Fig. 10. Image-to-Image Translation Example, [12], [13]

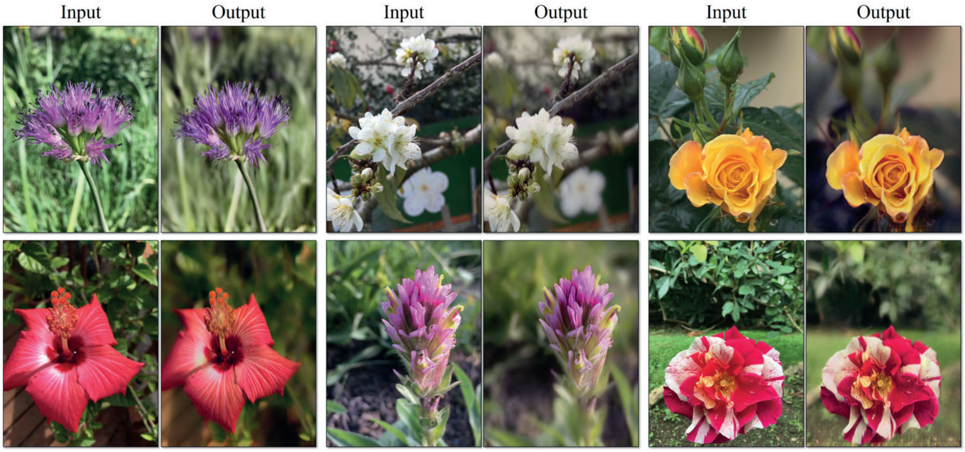

Fig. 11. Image-to-Image Translation Example; [12], [13]

3. Haar Cascade For Vehicle Detection

For purposes of vehicle (automobile) Haar cascade training the author has manually recorded the traffic on one of the busy roads in Warsaw, Poland. 1250 1920x1080 (HD) frames were obtained and manually labeled using Opensource software [14]. For motorbikes, cars, trucks and other vehicles only one label, "automobiles" was used. Each frame used for processing consisted of from 1 to 20 automobiles, with 4 objects on average. This consisted of "positives" dataset, that is – the dataset, which included object of interest. Haar cascade training additionally required "negatives" dataset – frames not consisting the object in question. Author experience shows that increasing the number of negatives as a proportion of positive images reduces the overall detection rate of the classifier. The dataset of 450 negative frames, usually consisting of typical the road related object such as street signs, bushes, trees, lamps, road crush barriers etc, was assembled. The analysis performed consisted of range of Haar cascade models with 450 negatives and 450-1250 positive frames with an increment of 100. This approach was compared to the second analysis performed with mirrored positive images additionally added to the dataset, increasing the number of positive images to 900-2500 (with an increment of 200), while maintaining constant negative dataset. 25 stages, Haar feature type, 24x24 frames, the 0.99 minimum hit rate and 0.5 max false alarm rate were used as Haar cascade parameters. All models were validated on 350 independent frames taken under different lightening conditions and for different vehicles than training model. The validation data set was not used in the training process.

True positive is a metric defining each object correctly identified as an object to be detected. False positive describes an incorrect identification of a picture area as the object. True negative would be a picture entire deprived of vehicles with no detections at all (not present in the results set as no pictures of the kind were used). A false negative is an object omitted by the classifier.

The results of the analysis are presented in Figures 12, 13 and 14. True positive number increases with a number of positive frames used with vertical mirror image augmentation successfully increasing the number by 11-57%. Unfortunately the number of false positive results increases even faster, by 47-110% or almost 4 times percentage wise. This is partially compensated by a reduction of the number of false negative by a range of 6-28%. Overall model accuracy change ranges from 25% decrease to 23% increase with an average reduction of 2%.

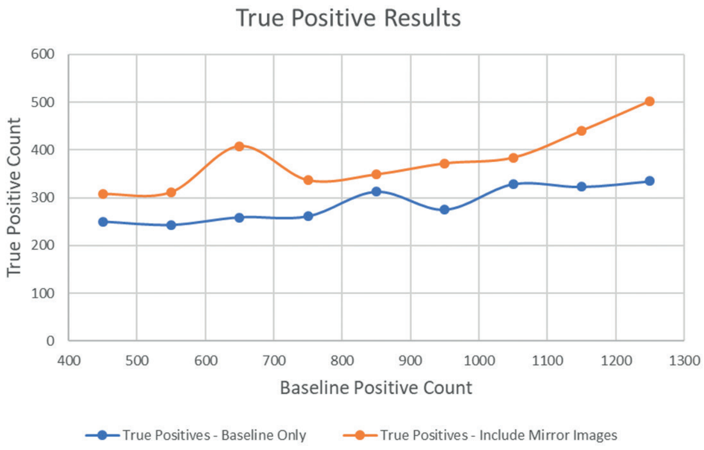

Fig. 12. Change in true positive count over the range of 450-1250 positive datapoints used for analysis with negative dataset constant at 450 for augmented and non-augmented dataset

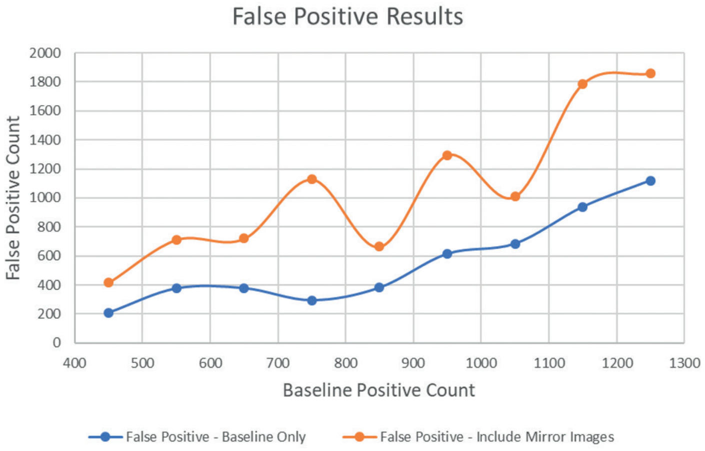

Fig. 13. Change in false positive count over the range of 450-1250 positive datapoints used for analysis with negative dataset constant at 450 for augmented and non-augmented dataset

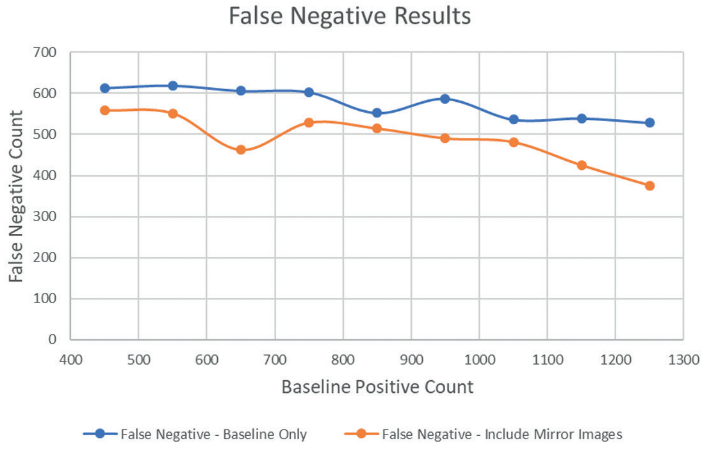

Fig. 14. Change in false negative count over the range of 450-1250 positive datapoints used for analysis with negative dataset constant at 450 for augmented and non-augmented dataset

At the same time dataset augmentation results in a higher number of training dataset volume (positive and negative dataset combined), which in turn results in more complicated classifier that ultimately results in longer detection time and Frames Per Second (FPS) count. Linear interpolation of obtained detection results have been presented in Figure 15, while typical "real-time" application is usually defined by 30 FPS or more. The low FPS rate is preliminarily driven by the high resolution of the analyzed frame (HD or 1920x1080), which drives the computation time. The low resolution 640x480 image achieve 30 and more FPS.

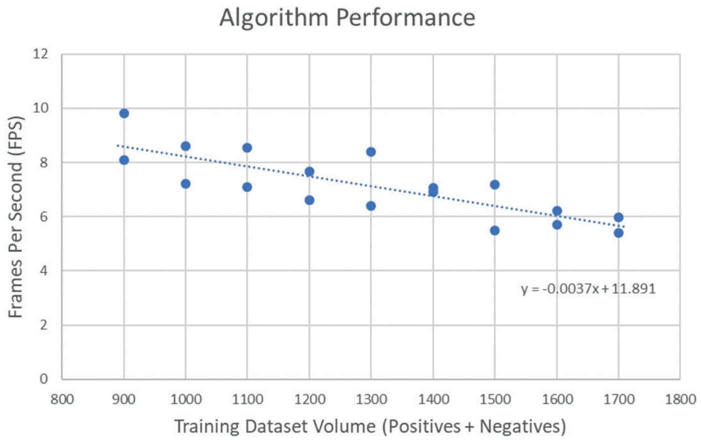

Fig. 15. Change in predicted FPS for model detection depending on the amount of datapoints used for model generation, including both augmented and non-augmented dataset

4. Conclusion

A data augmentation is a common technique for a dataset multiplication. While almost all techniques are applicable for object detection (classification, whether the object in question is in the picture or not) object localization (stating where exactly in the picture is the object) requires a more detailed approach and additional picture information. The information usually comes in the form of the object bounding box, which often requires additional operations to ensure that new picture contains the requested information.

New data augmentation techniques, such as image-to-image translation will allow photograph to retain useful object information, with no impact on the bounding box. This in turn will reduce model turnaround time and data acquisition costs, while increasing model accuracy.

Simple data augmentation techniques, like image vertical mirroring enable higher true positive rate with drawback in terms of higher false positive and false negative rate. Overall accuracy reduction of approximately 2% as well as increase in model computational time prevent object mirroring as a single only technique used for data augmentation.

Future work will consist of utilization of other techniques and their combination to enable the increase in overall model accuracy while preventing the increase in computational time.

References

[1] Başer E., Altun Y. Detection And Classification Of Vehicles In Traffic By Using Haar Cascade Classifier. Proceedings of the 58th ISERD International Conference, Prague, Czech Republic, December 2016.

[2] Breckon T. P., Barnes S. E., Eichner M. L., Wahren K. Autonomous Real-time Vehicle Detection from a Medium-Level UAV. Proc. 24th International Unmanned Air Vehicle Systems, 29.1-29.9, 2009.

[3] Capece N., Erra U., Scolamiero R. Converting Night-Time Images to Day-Time Images through a Deep Learning Approach. 2017 21st International Conference Information Visualisation (IV), London, 2017, 324-331, doi: 10.1109/iV.2017.16.

[4] Cheng H. Y., Weng Ch. Ch., Chen Y. Y. Vehicle detection in aerial surveillance using dynamic Bayesian networks. IEEE Transactions on Image Processing, vol. 21, no. 4, April 2012, pp. 2152-2159.

[5] Haralick R. M., Shanmugam K. Texturial Features for Image Classification. IEEE Transactions on Systems, Man, and Cybernetics, 1973; doi:10.1109/TSMC.1973.4309314.

[6] Haar A. Zur Theorie der orthogonalen Funktionensysteme. Mathematische Annalen, 1910, 331–371.

[7] Papageorgiou C., Oren M., Poggio T. A general framework for object detection. Proceedings of the IEEE International Conference on Computer Vision, February 1998, 555-562, doi: 10.1109/ICCV.1998.710772.

[8] Pawełczyk M., Bibik P. Usage of modern engineering software in the design of unmanned rotorcraft. Prace Instytutu Lotnictwa eISSN 2300-5408 231, Warsaw 2013, 52-59.

[9] Soo S. Object detection using haar-cascade classifier. Institute of Computer Science, University of Tartu, 2014, 1-12.

[10] Tsai L. W., Hsieh J. W., Fan K. Ch. Vehicle detection using normalized color and edge map. IEEE Transactions on Image Processing, March 2007, 16; 3: 850-864.

[11] Viola P., Jones M. Rapid Object Detection using a Boosted Cascade of Simple Features, Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, doi: 10.1109/CVPR.2001.990517.

[12] Zhu J.-Y., Park T., Isola P., Efros A. A. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, 2017 IEEE International Conference on Computer Vision (ICCV), 2017, doi: 10.1109/ICCV.2017.244.

[13] https://github.com/junyanz/CycleGAN.

[14] https://github.com/tzutalin/labelImg.

[15] https://software.intel.com/en-us/neural-compute-stick.