Advances in Manufacturing Technology XXXIV
M. Shafik and K. Case (Eds.)

doi:10.3233/ATDE210005

Development of an Optimised Dataset for Training a Deep Neural Network

Callum NEWMAN[a,1], Jon PETZING[a], Yee Mey GOH[a] and Laura JUSTHAM[a]
[a] *Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University, Epinal Way, Loughborough, LE11 3TU, UK*

Abstract. Artificial intelligence in computer vision has focused on improving test performance using techniques and architectures related to deep neural networks. However, improvements can also be achieved by carefully selecting the training dataset images. Environmental factors, such as light intensity, affect the image's appearance and by choosing optimal factor levels the neural network's performance can improve. However, little research into processes which help identify optimal levels is available. This research presents a case study which uses a process for developing an optimised dataset for training an object detection neural network. Images are gathered under controlled conditions using multiple factors to construct various training datasets. Each dataset is used to train the same neural network and the test performance compared to identify the optimal factors. The opportunity to use synthetic images is introduced, which has many advantages including creating images when real-world images are unavailable, and more easily controlled factors.

Keywords. Artificial Intelligence, Object Detection, Optimisation, Construction

1. Introduction

Object detection has been successfully implemented in many applications, such as autonomous vehicle object avoidance [1]. Although many cases have good test performance, there is potential for further improvements by using an optimised training dataset, which is important to consider as it is one of the fundamental tools used during training. When a training dataset is created, various images of the object of interest are gathered usually without considering how environmental factors affect the final test performance. Environmental factors greatly affect the appearance of an image and therefore affect the appearance of image features, which in turn can affect the test performance. These affects are something current research has not fully investigated.

This research investigates how environmental factors affect the test performance of a deep neural network (DNN) being trained to detect various construction machines. The process identifies their affect and how an optimal level can be found. The research shows a training dataset can be created which improves the performance of a DNN. It also briefly introduces how the same datasets can be replicated using synthetic data.

It was hypothesised that different factors will affect the performance in either a positive or negative manner. By selecting the best factor levels the performance can be increased compared to a dataset with random distributions of factor levels.

[1] Corresponding Author. c.newman@lboro.ac.uk

2. Background

Object detection can be achieved using multiple tools, however DNNs are a popular choice as they can achieve fast speeds and increased accuracy. Multiple DNNs can be used and fall into two categories: single-stage or two-stage detectors. Single-stage detectors, such as You Only Look Once (YOLO) [2], perform bounding box regression and classification in one network. Two-stage detectors, for example R-CNN [3], isolate bounding boxes and then perform classification with separate networks. Although single-stage detectors are typically faster, it usually comes with a sacrifice to the test accuracy. Additionally, there are multiple network architectures which are commonly used as the base architecture of the DNN being trained, such as ResNet [4].

Training DNNs uses a dataset of images split into at least two parts: training and testing. The test dataset contains images which accurately represent the target domain, and the training dataset contains various images of the object of interest in different scenarios. Datasets are publicly available for computer vision tasks, such as object detection and semantic segmentation, and cover various applications, for example general objects in ImageNet [5] and autonomous vehicles [6]. Public datasets cover few applications and so it is common for new datasets to be made specific to the developed application, for example detecting construction machines [7]. While any dataset can be used to successfully train a DNN there is no guarantee it will achieve optimal performance. Typically, training datasets are created using a process which does not consider the distribution of factors that affect the appearance of images and if there are any interaction between the factors. Some factors are loosely considered, for example vehicle datasets drive around different streets encompassing various scenarios [6], but do not consider the effect of these on the test performance. Therefore, a dataset may be using images taken under suboptimal conditions and negatively affecting the training performance of the DNN.

Furthermore, a dataset can be created using synthetic images created using computer programs by either altering existing real-world images [8], combing real-world images with 3D models of objects [9] or rendering images from 3D simulations [10]. Environmental factors can also be considered when developing synthetic datasets, along with simulation factors related to the process used to generate the images. Synthetic datasets have more investigations into the factors but are limited and do not consider factor interactions.

3. Data Collection Methodology: Real-world controlled environment

The following dataset was developed to test how different factors affect the performance of a DNN. A scale model environment was used to implement and control the factors; background environment, light intensity, and occlusion. By using a controlled environment, each factor level was known and different combinations of factor levels were gathered. The objects of interest which have been used in the example dataset are three construction machines: excavator, wheel loader, and dump truck.

Multiple scenes were created where objects were placed and each factor was varied, as demonstrated in Figure 1, whilst minimising the movement of objects to ensure the features of the images changed primarily from altering each factor. The training and test set images were gathered by placing objects in the environment, taking images under each light intensity, adding in occlusion, retaking the images, changing the background, and repeating the process. The training set used 191 combinations of machines.

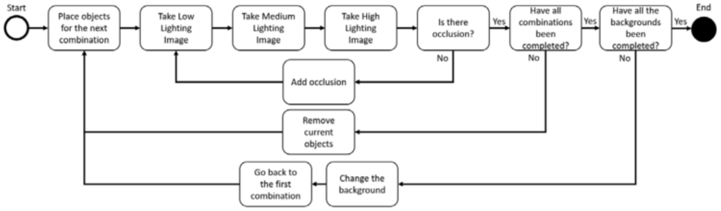
Start
Place objects for the next combination
Take Low Lighting Image
Take Medium Lighting Image
Take High Lighting Image
Is there occlusion?
Yes
Have all combinations been completed?
Yes
Have all the backgrounds been completed?
Yes
End
No
No
No
Add occlusion
Remove current objects
Go back to the first combination
Change the background

Figure 1. Diagram illustrating the process in which the images for the dataset were taken in order to control the three factors whilst minimising the movement of objects.

There were two background environments created to take images for each background. The first environment represented a quarry using a 3D rendered backdrop of a rock wall, sand as a ground material, and rocks as extra objects. The second environment was a forest with a backdrop of trees, soil as the ground material and wood as extra objects. Each background environment is illustrated in Figure 2.

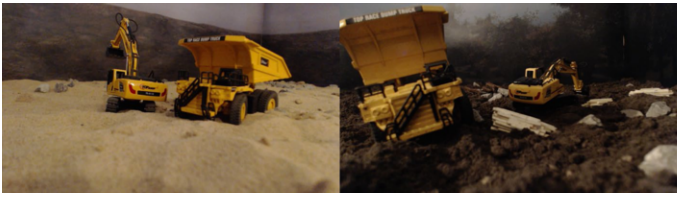

Figure 2. Images of the dump truck and excavator on different backgrounds. Left: Quarry, Right: Forest.

Light intensity had three levels created using a single light source which had its position changed to alter the intensity of light incident on the objects, as demonstrated in Figure 3. Figure 3 also shows how occlusion can be achieved using the environment.

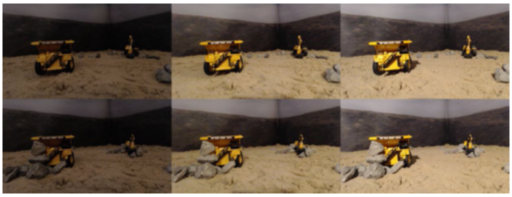

Figure 3. Example scenes under each light condition and occlusion status. Left Column: Low lighting, Middle Column: Medium lighting, Right Column: High lighting. Top Row: Without occlusion, Bottom Row: With Occlusion.

The change in light intensity had a varying affect with each background due to the properties of each material. To quantify the change in intensity the RGB values for each pixel was totaled and averaged for each background test set. This illustrates the relative change in intensity, because when the light intensity increases the RGB values increase towards values of [255, 255, 255]. The average intensity value for each background is presented in Table 1.

Table 1. Average intensity value for each light category across each test set background.

Background	Low Intensity	Medium Intensity	High Intensity
Quarry	3.0x10^{8}	6.1x10^{8}	6.8x10^{8}
Forest	7.0x10^{7}	1.8x10^{8}	5.0x10^{8}

4. Dataset Evaluation and Analysis

In total, the training and test set had 2796 and 681 unique images respectively, each labelled with the machine labels: "Loader", "Dumper", and "Excavator". The training images were split into 60 sub-datasets containing 191 images encompassing various levels of each factor. The background was split into three levels having either the camera remain stationary in the quarry environment creating only one background, the camera moving ten times within the quarry environment making 1 scenario, or the camera moving but with an even split across the quarry and forest environment. For lighting either one of the three light intensities were used or a 33% mix of the three. For occlusion the sub-datasets had either 0%, 25%, 50%, 75%, or 100% of the images with occlusion.

Each sub-dataset was used to successfully train a YOLOv2 [2] network with ResNet-18 [4] as the base architecture, and then tested on the same test set, which had an even split of images across each background, light intensity and with and without occlusion. The test performance was determined using the standard metric of Mean Average Precision (mAP_{50}). This produced a value between 0 and 100, where 100 is the highest performance. Transfer learning [11] can improve training performance and time, therefore a network pre-trained on the ImageNet dataset [5] from the MATLAB deep learning toolbox [12] was used. Each network was trained for 50 epochs at a learning rate of $1x10^{-3}$, followed by 10 epochs at $1x10^{-4}$ for finetuning. Table 2, 3 and 4 present the results for the 1 Background, 1 Scenario and 2 Scenarios categories respectively.

Table 2. mAP_{50} for each dataset using "1 Background". Light Intensity VS percentage of images with occlusion.

Lighting	0%	25%	50%	75%	100%
Low	42.3	43.1	45.9	43.2	49.4
Medium	45.9	52.8	48.9	47.6	49.2
High	36.3	43.4	46.7	50.0	53.8
33% Mix	39.8	46.6	47.0	57.9	49.0

Table 3. mAP_{50} for each dataset using "1 Scenario". Light Intensity VS percentage of images with occlusion.

Lighting	0%	25%	50%	75%	100%
Low	50.7	48.7	46.3	42.6	50.0
Medium	49.2	56.6	52.0	54.0	52.6
High	49.9	53.8	56.1	50.6	52.2
33% Mix	41.3	45.4	50.2	50.7	53.4

Table 4. mAP_{50} for each dataset using "2 Scenarios". Light Intensity VS percentage of images with occlusion.

Lighting	0%	25%	50%	75%	100%
Low	49.4	51.2	49.8	49.9	49.2
Medium	62.0	58.7	58.9	60.2	59.6
High	54.9	57.4	57.5	61.0	56.8
33% Mix	61.6	53.0	58.8	52.9	54.7

An Analysis of Variance test (ANOVA) was performed to help identify which factors had the most significant affect on the test performance and whether any factors interacted which might then need optimizing together. The p-values produced by the ANOVA test are presented in Table 5. The p-values represent the probability that the null-hypothesis can be accepted. In this case the null hypothesis is "the factor/interaction does NOT have significant affect on the test performance of the trained DNN". It will be considered that any value below 0.05 (95% confidence level) will reject the null hypothesis and therefore the factor/interaction has significance.

Table 5. P-value results from the ANOVA test performed on the test results of each sub-dataset.

	Background (B)	Lighting (L)	Occlusion (O)	B*L Interaction	B*O Interaction	L*O Interaction
P-value	0.000	0.000	0.07	0.117	0.088	0.562

The ANOVA results suggest the background and light intensity factors are significant, however occlusion is less so. Furthermore, no interactions between factors are indicated. No interactions allows the optimal level to be picked by identifying the mean value of each factor, which can be found in Table 6. These suggest moving and increasing the number of scenarios improves performance. The light intensity should not be mixed with medium being the best and adding occlusion may improve performance.

Table 6. Mean values for each factor level.

Background Level	Mean mAP_{50}
1 Background	46.6
1 Scenario	50.3
2 Scenarios	55.9

Lighting Level	Mean mAP_{50}
Low	47.4
Medium	53.4
High	52.0
33%	50.8

Occlusion Level	Mean mAP_{50}
0%	48.6
25%	50.3
50%	51.5
75%	51.7
100%	52.5

Furthermore, to compare the sub-datasets performance to datasets created in a more conventional manner, three datasets were created which had random distribution of lighting and occlusion for each of the background levels. Each datasets test performance is in Table 7, which show the random datasets do outperform some sub-datasets but the more optimal sub-datasets perform better.

Table 7. mAP_{50} for each random dataset.

	1 Background Random	1 Scenario Random	2 Scenarios Random
mAP_{50}	45.3	51.0	52.4

5. Conclusion and Future Work

The aim of the paper was to show the investigation and process which has taken place to identify the affect different environmental factors have on the training of a deep neural network, and which factor levels are the most optimal. Various training datasets of construction machines have been created with different breakdowns of each factor level and all tested on the same test dataset. It can be concluded that some factors will have an effect on the final test performance, with each factor having a different significance on the performance, and that there will be an optimal factor level. However, the best performing dataset does not agree with the occlusion mean, perhaps suggesting that factors with less significance may not agree with the means. It was also seen that using the optimisation process produced optimal datasets that outperform more conventional dataset which have random distributions of factor levels.

To further develop the investigation, the results can be used to create larger datasets based upon which factor levels perform best and worst. If the best datasets outperform the worst it will suggest there is more value to the data than just finding the best performing sub-dataset. Additionally, the same tests can be applied to synthetic datasets. When developing the synthetic dataset, the same environmental factors can be considered with more factors related to the way in which the simulation is developed, such as the rendering engine used. The increase in the number of factors may also be common in applications with higher accuracy requirements which will be advantageous as there is more to investigate; however, the large number of factors can make it time consuming to test all possibilities. Therefore, the future work will investigate adaptations to the process to reduce the number of tests needed as the number of factors increases. The results of the tests on synthetic data may differ due to the domain gap between the real-world and simulation. For the current research the synthetic images will be rendered using a 3D simulation developed in Blender [13] to replicate the real-world controlled environment. Currently, the simulation environment and 3D models have been created, an example of which is illustrated in Figure 4.

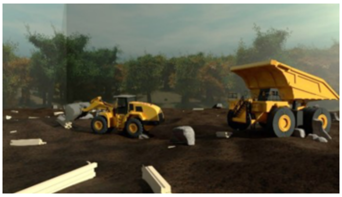

Figure 4. Examples of an image that could be generated using the Blender 3D simulation.

Acknowledgements

This work has been kindly funded by the UK Engineering and Physical Sciences Research Council (EPSRC) (Grant number EP/S515140/1).

References

[1] S. Howal, A. Jadhav, C. Arthshi, S. Nalavade and S. Shinde, “Object Detection for Autonomous Vehicle Using TensorFlow,” *International Conference on Intelligent Computing, Information and Control Systems*, vol. 1039, October 2019. Springer, Cham.

[2] J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, 2017.

[3] R. Girshick, J. Donahue, T. Darrell and J. Malik, “Region-Based Convolutional Networks for Accurate Object Detection and Segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142-158, 2016.

[4] 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*, Las Vegas, pp. 770-778, 2016.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, “ImageNet: A Large-scale Hierarchical Image Database,” in *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, pp. 248-255, 2009.

[6] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzwieler, R. Benenson, U. Franke, S. Roth and B. Schiele, “The Cityscapes Dataset for Semantic Urban Scene Understanding,” arXiv:1604.01685

[7] N. D. Nath and A.H. Behzadan, “Deep Convolutional Networks for Construction Object Detection Under Different Visual Conditions,” *Frontiers in Built Environment*, Vol. 6, pp. 97, August 2020.

[8] D. Dwibedi, I. Misra and M. Hebert, “Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection,” *IEEE Conference on Computer Vision (ICCV)*, Venice, Italy, December 2017.

[9] M. M. Soltani, Z. Zhu and A. Hammad, “Automated Annotation for Visual Recognition of Construction Resources Using Synthetic Images,” *Automation in Construction*, vol. 62, pp. 14-23, 2016.

[10] S. Hinterstoisser, O. Pauly, H. Heibel, M. Martina and M. Bokeloh, “An Annotation Saved is an Annotation Earned: Using Fully Synthetic Training for Object Detection”, *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW),* October 2019, Seoul, South Korea.

[11] L. Shoa, F. Zhu and X. Li, “Transfer Learning for Visual Categorization: A Survey,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 1019-1034, 2015.

[12] MathWorks, “MATLAB deep learning toolbox,” https://uk.mathworks.com/products/deep-learning.html
Blender, “Blender 2.81 Manual”, https://docs.blender.org/manual/en/2.81/#