Training Convolutional Neural Networks to Detect Waste in Train Carriages

Training Convolutional Neural Networks to Detect Waste in Train Carriages

Nathan Western
School of Engineering and Physical Sciences
Heriot-Watt University
Edinburgh, Scotland
nw29@hw.ac.uk

Xianwen Kong
School of Engineering and Physical Sciences
Heriot-Watt University
Edinburgh, Scotland
x.kong@hw.ac.uk

Mustafa Suphi Erden
School of Engineering and Physical Sciences
Heriot-Watt University
Edinburgh, Scotland
m.s.erden@hw.ac.uk

***Abstract*— This research constitutes a systematic investigation of the effect of image view on Convolutional Neural Networks (CNNs) when trained to detect waste in train carriages. Additionally, this research identifies neural network architecture and training conditions for use in an automated train cleaning robot. Specifically, we investigate the relationship between the size of the CNN training dataset, whether these images are taken from a view sympathetic to the CNN application, and the effectiveness of the trained networks. Three datasets were constructed specifically for this research; a large dataset of 58,300 studio images of waste in a variety of conditions, a smaller dataset of 4,515 images taken of actual waste items on trains, and a dataset of 7,290 images of actual waste on trains used to test the CNNs. The images taken on trains were captured from the perspective of a hypothetical cleaning robot that would use these networks. Additionally, we provide a comparison of MobileNetV2, ShuffleNet, and SqueezeNet CNNs based on their suitability for implementation in an automated train cleaning system, and the optimum conditions to do so. Training with a smaller dataset of images taken from a "robot-eye view" resulted in an average increase in classification accuracy of 10.5%, with the largest increase being 26%, when compared to training with a larger dataset of images of waste items in various poses. ShuffleNet was identified as the optimally performing CNN for waste detection, achieving an accuracy of 88.61% when trained with a small dataset of images sympathetic to the end use. MobileNetV2 was found to perform optimally with a larger dataset of training images, even if these are less specific to the application of the network.**

Keywords—Computer vision, object classification, feature extraction

I. INTRODUCTION

A. Using CNNs for Waste Detection in Train Carriages

Cleaning train carriages is an essential element of routine maintenance performed by every train operator. In the 2017-18 period UK train operators spent £4.4 billion on operating expenditure including fleet cleaning [1]. Automation could help reduce these costs and increase cleaning quality.

Convolutional Neural Networks are chosen as the method of detecting waste as they have been successfully utilized for a number of waste detection tasks [2][3][4][5]. There has been extensive work developing and adapting CNN models for use with mobile hardware [6][7], making them ideal for use in a mobile cleaning robot.

Research funded by the Rail Safety and Standards Board (RSSB), UK.

For utilizing machine learning for a specific task, it is not always optimal to use techniques such as one-shot learning [8] when image features are too similar or not clear due to low lighting. Additionally, transfer learning [9] is not appropriate for detection of items that do not have categories in available pre-trained CNN models. For this reason, existing CNN architecture is selected to be trained for use in an automated train cleaning system.

This presents the need for a specialized training method and the selection of appropriate CNN architecture. This needs to be sympathetic to the limitations posed on data collection and CNN accuracy by the train carriage environment.

B. Suitable CNNs for Mobile Applications

There are three popular CNNs for use with mobile computing. MobileNetV2 [10], ShuffleNet [11] and SqueezeNet [12]. The three networks take a three-dimensional image input (the x and y axes of the image, by the three RGB channels), process these with some form of convolutional layer, then provide an output of the percentage confidence in each of the unique image classes. Convolutional layers are used to extract features from an image, which can then be used to classify the image. All three are subject to testing to identify the most appropriate network for application in a train cleaning robot.

MobileNetV2 is a development of Google designed CNN MobileNet. It is the most complex neural network modified for this project, with the model used having 300 million multiply-adds. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is a benchmarking tool for image recognition algorithms with 1000 classes and 10,000,000 labelled images. MobileNetV2 achieved a top-1 accuracy of 74.7% on the ILSVRC ImageNet2012 dataset [10]. MobileNetV2 uses Google's " Inverted Residual and Inverted Bottleneck" layer [10], which is designed to extract image features more efficiently than traditional CNNs.

ShuffleNet is also a development of MobileNet by researchers at Megvii Inc.. It is more efficient than MobileNetV2, the model modified in this project containing 140 million multiply-adds. It achieved a top-1 accuracy of 67.6% on the ImageNet 2012 dataset [11]. It uses the same inverted bottleneck design as MobileNetV2 but utilizes channel shuffling to gain more feature information with less computational complexity.

SqueezeNet is a unique CNN for mobile computing application developed by Stanford and Berkley universities. It

is the smallest network tested, containing no fully connected layers and instead using very efficient fire modules and convolutional layers to reduce the output of the NN to a 1x1x1000 tensor for classification. It achieved an ImageNet 2012 top-1 accuracy of 57.5% [12].

II. Methodology

To determine the most appropriate CNN training method to classify poorly lit images of waste items on trains, this research first collected a large dataset of images of waste items taken in a studio environment (the "Studio" dataset). These images contained images of waste in three light conditions, with three different backgrounds, and in varying poses. Two smaller datasets of images of waste on trains were also collected (the "Carriage" and "Test" dataset). These contained only images of waste found on trains, in situ, arriving at a busy London station. In this study, as for many applications, a large dataset of appropriate images is not available.

TABLE I. Comparison of the Developed Datasets

Dataset Name	Number of Unique Images	Number of Images After Preprocessing	"View" of included images
Studio Dataset	11,600	58,300	Three different lighting conditions, three different backgrounds.
Carriage Dataset	903	4,515	Actual images of waste on trains, mostly low light conditions.
Testing Dataset	1,458	7,290	Actual images of waste on trains, mostly low light conditions.

In the case of collecting images of waste items on trains, it is difficult to collect these safely due to the quick turn-around of trains at destination stops and the existing cleaning procedures that must be undertaken in this time. This created a need for a systematic approach to compare the effectiveness of a larger, less specialist dataset, against a smaller dataset of "ideal" images. The main differences between the two datasets is that the Carriage Dataset represents waste as it is found in train carriages, with lighting conditions and background consistent with the use case of the CNNs. This is considered the "view" of the images.

Initially, we modified the three chosen CNNs to be appropriate for the classification task by reducing the classifier layer to 5 outputs. These are: disposable coffee cups, plastic PET bottles, glass beer bottles, aluminum cans and newspapers. These classes are chosen as they are identified as being high volume and problematic waste items in interviews we have conducted with three representatives from three rail service providers in the UK, Greater Anglia, MerseyRail and ScotRail. Coffee cups and newspapers are highlighted as the most problematic waste items. Both occur in much higher volume than the other waste items, and newspapers additionally cause issues to the running of train services by becoming stuck in the door closing mechanism. This causes the doors to remain open, preventing onward travel. Due to high frequency, the Carriage Dataset contains more images of these two classes.

To change the CNNs to classify from 5 classes, the last fully connected layer in the network needs to be modified to give five outputs rather than the default (around 1000 classes for a network designed for the ImageNet2012 dataset). The last fully connected layer in a CNN takes the information processed about the input image and relates it to the output classes in the form of a $1*X$ vector, where X is the number of classes. For example, the distinctive shape of a takeaway coffee cup would produce a high output in the first dimension of the vector, relating to cups. The last fully connected layer in all networks is changed to a $1*5$ vector output. This also changes the shape of the softmax layer, immediately after the fully connected layer. The softmax function changes the numerical outputs of the fully connected layer from a value of ± infinity to a value between 0 and 1. This is more easily recognized as a percentage confidence value. Finally, the class output layer is changed to represent the five classes described previously. This makes the CNN prediction more suited to human interpretation, an example output may be "Newspaper, 0.7813%" indicating that the network predicts the input image is that of a newspaper, with 78% confidence.

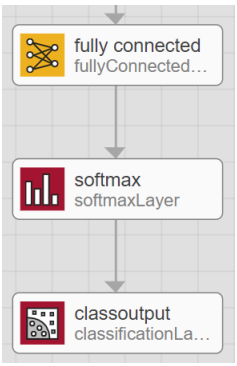

Figure 1. The fully connected, softmax and class output layers of a CNN.

These CNNs are initially trained on the Studio dataset. The CNNs were tested for their accuracy, calculated as a percentage of correctly classified images. Then the CNNs were then trained using the Carriage Dataset and tested in the same way as the networks trained with the Studio dataset. All three networks are also trained with both datasets. This is performed twice, once training first with the Carriage Dataset, then fine-tuned with the Studio, and then trained first with the Studio images, then fine-tuned with the Carriage set. The first training case (Carriage images then Studio) demonstrates whether it is more important for the network to first identify features present in the end use case, where images are busy and poorly lit, and then fine-tune to "learn" further features that may not be present in the Carriage Dataset. The second training case (Studio images then Carriage) seeks to build a robust understanding of the features of each class before retraining with actual images of waste sympathetic to the end use of the classifier.

As well as identifying the ideal CNN architecture to deploy this training method, this study also seeks to identify an optimal CNN to use in the automated train cleaning robot. Therefore, the average classification speed for each network was also calculated to provide another metric for performance. Comparisons are also drawn to each CNNs accuracy in less specific classification problems, using ILSVRC's ImageNet2012 results. It is worth noting that the accuracy used is "Top-1" accuracy, which is the accuracy considering only the most confident classification from the network. Commonly used "Top-5" accuracy, the CNN accuracy of the 5 most confident classes for an image, is not appropriate here. This is because in the case of a CNN with 5 class outputs, "Top-5" accuracy is always 100%.

III. Creating the Datasets

A. Creating a Rig for Taking Training Images

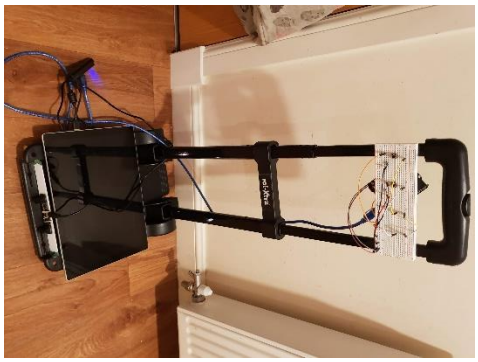

Figure 2. The rig for taking images.

In order to facilitate image collection on trains, we have designed and implemented a rig that can easily navigate train carriages and take images at floor level, including the hard to reach under-seat area. The rig is designed to be built around a luggage trolley as these are mobile and have mounting points at floor level for hardware. The shape of the trolley also means the camera can be mounted ahead of the user. This allows the user to take photographs underneath seats without bending down. Not only is this important for the comfort of the user, but also increases the speed in which images can be collected. Additionally, this dataset displays the features of the class item from a point of view that is sensitive to the application. Previous research designing a CNN for a mobile cleaning application discovered that when training a network on images taken from above, accuracy of recognition of objects from the "robot's eye view" is 3.38% less than when tested with images of the object taken from above [13].

The camera used is an ELP "Synchronisation" model camera. As the waste identification algorithm is designed to run on mobile hardware only low-resolution images will be used. This means it is appropriate to utilize a low-cost stereo imaging system, taking 1.3MP images. A stereo imaging system is chosen as it will take two similar pictures on each press of the shutter button, creating more training images.

Figure 3. The stereo camera.

A python program is created that identifies a number input from the Bluetooth number-pad, takes a photo, and then saves the image to a corresponding folder depending on which number key has been pressed. This creates the dataset. The images are split before being saved, as the stereo camera displays the output of both lenses as one image. Then each image is sequentially numbered. The program taking and collating the images runs from a Microsoft Surface Pro 4 mounted to the section of the trolley closest to the user. This is chosen as it can be mounted flat to the trolley to clearly display the images to the user. A Bluetooth number-pad acts as a shutter button, with each number corresponding to a different type of waste item.

B. Developing the Carriage and Test Datasets

For the Carriage Dataset, we collected 903 still images taken of waste items on trains over a period of three days. The images are taken on Greater Anglia trains arriving at London Liverpool Street station in the mornings. These include a selection of commuter, cross country and inter-city trains, including British Rail class 90, 317, 321, 360, and 379 Locomotives.

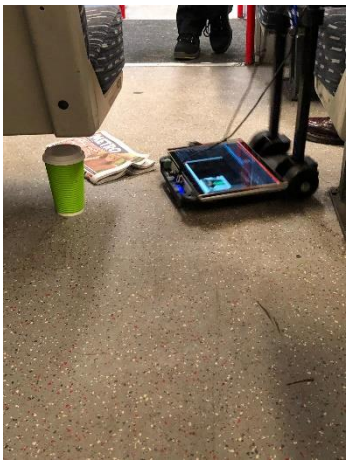

Figure 4. Taking images for the "Carriage" dataset.

The Test dataset is collected in the same manner, taking images of waste on trains arriving at London Euston station. It was important these images were entirely separate to the Carriage Dataset, so no unique waste item appears in the testing and training of the CNNs. 1458 photos were taken.

C. Developing the Studio Dataset

The Studio dataset is comprised of images taken with the rig in a studio environment. These are sorted into folders pertaining to the type of waste present in each image. The folders are labelled:

- Class 1: bottle, plastic bottle, drinks bottle
- Class 2: can, aluminum can, drinks can
- Class 3: coffee cup, paper cup, disposable cup
- Class 4: glass bottle, beer bottle
- Class 5: newspaper, magazine

To take the images, we set up a small studio, with three different neutral backgrounds and three different lighting intensities. These are low light, indirectly lit, and directly lit. Having these different conditions represented is to ensure good detection levels in all lighting. The images are taken with a rig holding a stereo camera at a "robot's eye view", with a python program splitting the stereo image into two images and storing them in the appropriate folder. A total of 11,660 images are collected to make the dataset.

The studio images were designed to provide as much data about the features of each waste item as possible, so each unique item was photographed at many different angles. In addition, waste items that are deformable (such as newspapers or coffee cups) were also represented in their deformed states.

This is because waste items are often found like this in train carriages. This designed hoped to reduce any differences between the two training datasets except the background and lighting of training images, and the size of the datasets.

Figure 5. A sample of images taken for the Studio dataset.

D. Image Preprocessing

Image preprocessing is achieved with a Python program written to crop the images to the desired size, and then rotate and flip them, and then number and save the images. This is modified to create 2 datasets, for the different image input sizes needed for the CNNs. Python Image Library (PIL) is used as it has premade functions to crop and flip images. Images are rotated 90 degrees left and right and flipped horizontally and vertically. This creates five images for every photo captured.

IV. Training the Networks

The CNNs are tested and evaluated on a PC utilizing an AMD FX-8350 CPU, 32GB of RAM, and an NVidia GeForce RTX 2070 graphics card with 8GB of dedicated memory. MATLAB was used to modify, train, test, and evaluate the three CNN models, as recommended by Nvidia [14].

The MobileNetV2 and ShuffleNet require an input image size of 224x224 pixels, SqueezeNet has an input image size of 227x227 pixels. For both the Studio and Carriage Datasets, 70% of the images are used for training, with the remaining 30% used for validation, as recommended by Elkan in [15]. Validation is used during training to monitor the network accuracy on a set of unseen images.

The models are trained with the Adam optimizer [16], with 8 epochs and a mini batch size of 64. This is chosen as it is an efficient optimizer that performs well in comparison with other available options such as Stochastic Gradient Descent optimization. The learning rate used is 0.001 as is recommended in the Adam paper [16]. The loss function is calculated using cross entropy loss.

We trained the MobileNetV2, ShuffleNet and SqueezeNet CNNs for a comparison of top-1 accuracy and classification speed. Top-1 accuracy is chosen over top-5 as in a task with 5 classes top-5 accuracy will always be 100%. Classification speed is used to determine the relative complexity of the three networks. Each unique neural network is first trained three times with the Studio dataset. This creates three trained models, which are then tested with the Test dataset. In order to maintain repeatable results, the Mersenne Twister Pseudorandom Number Generator (PRNG) is used, with seeds of 3, 6 and 9 for all networks. These three tests of each network are used to create an average, which is used in the results below.

Then we trained the three CNNs with the Carriage Dataset, again using the Mersenne Twister PRNG for replicable results, with seeds 3, 6, and 9. Testing was conducted with the same Test dataset as described above.

To understand the effect of training with both datasets, the models trained with the Studio and Carriage Datasets are fine-tuned with the other, following the same training and testing process.

The results compare top-1 classification accuracy, precision, and recall of the neural networks. Comparisons are drawn between the effectiveness of training with a large dataset of less specific images and a smaller dataset of images with a "view" that is more specific to the use case. Additionally, the three networks are compared against each other, with a view to identifying the network most suitable for use in an automated train cleaning system.

Precision is a measure of a neural network's ability to correctly classify a certain class. It is calculated by taking the number of true positives (correct classifications) and dividing by the true positives plus the false positives (items incorrectly identified as belonging to the class in question) [18]. Recall is the CNN's ability to classify as many items in a class as possible. It is calculated by taking the number of true positives and dividing by the true positives plus false negatives (items from that class not correctly predicted) [18].

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (1)$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (2)$$

Additionally, the speed of each network was calculated to represent their relative computational complexity. This was achieved using the MATLAB "tic/toc" function. We created a program that loads the test images and then measures the execution time of only the classification of the CNN. This program measures an average of the three trained models of each CNN.

Training for each of the three CNN models with the Studio and Carriage Datasets is repeated three times, using a Random Number Generator (RNG) to control randomly generated variables in the machine learning process and produce replicable results. This process resulted in 18 trained networks.

V. Results

A. MobileNetV2

The fully connected layer before the classifier in the MobileNetV2 CNN is modified to reduce the number of classes to five. The modified MobileNetV2 is trained with the Studio dataset and provided an accuracy of 61.1%. This network was trained with the Carriage Dataset giving an accuracy of 54.3%. The loss in accuracy when trained with the Carriage Dataset was theorized to be due to the small size of the dataset when compared to the 1,281,167 images in the ImageNet 2012 dataset [17]. When trained first on the Studio dataset and then fine-tuned with the Carriage Dataset MobileNetV2 achieved an accuracy of 62.7%. When this was reversed, and the network trained with the Carriage then the Studio dataset accuracy was 72.28%. Evaluation also

provided the classification speed of 33.5 milliseconds, which will be used to compare with the other networks.

TABLE II. MOBILENETV2 PRECISION AND RECALL

MobileNetV2 Precision and Recall	Precision		Recall	
	Cups	*Newspapers*	*Cups*	*Newspapers*
Trained on Studio Dataset	66.62%	63.26%	29.61%	88.33%
Trained on Carriages Dataset	48.92%	78.9%	76.16%	41.78%
Trained on Studio then Carriages Dataset	75.78%	84.56%	66.51%	80.63%
Trained on Carriages then Studio Dataset	67.14%	62.66%	30.74%	90.4%

For the precision and recall metrics only cups and newspapers are considered here. As by far the largest contributor to waste on trains these are the most important classes to correctly identify. The highest precision and recall MobileNetV2 achieved for the newspaper class was when trained with the Studio then the Carriage dataset. For the cups class the highest precision the network achieved again was with the Studio then Carriages training scenario. For recall in the cups class the network performed best when trained with the Carriages dataset alone.

B. ShuffleNet

The ShuffleNet model had the same changes made to the architecture as MobileNetV2, reducing the output classes to 5. Training on the Studio dataset produced an accuracy of 62.6%. When trained with the Carriage Dataset the network achieved an accuracy of 88.6%. Training with the Studio dataset followed by fine-tuning with the Carriage Dataset gave an accuracy of 59.04%. Training with the Carriage Dataset and then the Studio dataset produced an accuracy of 83.42%. This ShuffleNet CNN had a mean prediction time of 38.2 milliseconds.

TABLE III. SHUFFLENET PRECISION AND RECALL

ShuffleNet Precision and Recall	Precision		Recall	
	Cups	*Newspapers*	*Cups*	*Newspapers*
Trained on Studio Dataset	88.24%	61.33%	19.73%	98.13%
Trained on Carriages Dataset	93.01%	91.05%	83.72%	96.46%
Trained on Studio then Carriages Dataset	91.22%	58.68%	8.96%	99.52%
Trained on Carriages then Studio Dataset	91.54%	88.26%	76.54%	92.32%

ShuffleNet performed best in the precision metric for both cups and newspapers when trained with the Carriages dataset only. This is also true for recall in the cups class. The highest recall for the newspapers class achieved by the network occurred when it was trained with the studio dataset and then the carriages dataset.

C. SqueezeNet

We modified the SqueezeNet model to produce 5 class outputs. Initially this is done using a 1x1 filter to reduce the depth of the class layer, as in SqueezeNet's fire modules, but this produced poor results. Instead, an additional fully connected layer is added before the classifier. The modified SqueezeNet model achieved a training accuracy of 55.4% when trained with the dataset of studio images. When trained with the Carriage Dataset, the network achieved an accuracy of 67.7%. Training with the Studio and fine-tuning with the Carriage Dataset produced an accuracy of 61.02%, while training with the Carriage images then using the Studio dataset for fine-tuning produced a 66.31% accuracy. This modified version of SqueezeNet had a mean classification speed of 10.2 milliseconds.

TABLE IV. SQUEEZENET PRECISION AND RECALL

SqueezeNet Precision and Recall	Precision		Recall	
	Cups	*Newspapers*	*Cups*	*Newspapers*
Trained on Studio Dataset	53.5%	73.87%	57.23%	57.53%
Trained on Carriages Dataset	63.46%	70.37%	56.56%	80.72%
Trained on Studio then Carriages Dataset	67.62%	60.63%	23.21%	92.85%
Trained on Carriages then Studio Dataset	67.29%	74.42%	56.85%	76.47%

The best performing conditions for SqueezeNet when considering the precision in the cups class and recall for newspapers was training with the Studio then the Carriages dataset. For the metric of precision in the newspapers class training with the Carriages then the Studio dataset achieved the highest accuracy for SqueezeNet. The highest recall in the cups class SqueezeNet achieved was the result of training with the studio dataset only.

D. Comparisons Between the Networks

TABLE V. FINAL TEST RESULTS – CNN ACCURACY, A COMPARISON OF THE FOUR TRAINING CONDITIONS

Final Testing Results	CNN Performance (Top-1 Accuracy)			
	Trained on Studio Dataset	*Trained on Carriage Dataset*	*Trained on Studio then Carriage Dataset*	*Trained on Carriage then Studio Dataset*
MobileNetV2	61.12%	54.34%	62.7%	72.28%
ShuffleNet	62.57%	**88.61%**	59.04%	83.42%
SqueezeNet	55.38%	67.71%	61.02%	66.31%

For comparison, classification speed of the CNNs and the top-1 accuracy of the three networks when trained and tested on the ILSVRC2012 1000 class problem is displayed below:

TABLE VI. COMPARISON OF CNN ILSVRC2012 TOP-1 ACCURACY AND CLASSIFICATION SPEED

	ILSVRC2012 Top-1 Accuracy	*Classification Speed for 1 Image (Milliseconds)*
MobileNetV2	74.7% [10]	33.5
ShuffleNet	67.6% [11]	38.2
SqueezeNet	57.5% [12]	10.2

VI. CONCLUSION

Training with the Studio dataset and then fine-tuning with the Carriage dataset was tested to discover if this would provide the networks with a complete feature map of all the chosen waste items, and then retrain with more specific images. This did not occur in all models, producing the lowest accuracy of all trained ShuffleNet models, the second lowest

of the trained SqueezeNet models. It did provide an increase in MobileNetV2 accuracy when compared to training with only a single network.

When training with the Carriage dataset, then fine-tuning using the Studio dataset, all three networks produced a higher accuracy than when trained with the Studio then Carriage datasets. Notably, this training case produced the highest accuracy for MobileNetV2. Training in this way is designed to provide the neural network with a good map of waste item features that appear in the low light under seat areas, and then using the Studio dataset to improve the feature maps.

Two of the CNNs benefited from the smaller dataset of images which only included photos taken in the same lighting and background conditions as the end use. This would indicate that when using ShuffleNet or SqueezeNet, ensuring that the lighting and background of the training images is appropriate is more important than the number of images.

The CNN that achieved the highest accuracy was ShuffleNet when trained with the Carriage dataset (bold in Table V). The network achieved an accuracy of 88.61% under these conditions. It also achieved the highest precision and recall in the cups and newspapers class in almost all training scenarios, notably also when trained with only the Carriage images. This shows that ShuffleNet can accurately classify images even in low light, busy conditions.

The ShuffleNet paper [11] reports an error rate of 32.4% when trained and tested using the ILSVRC ImageNet 2012 dataset. An increase in accuracy of 21% is seen here.

For cases where collecting training images for a task is not possible, MobileNetV2 achieved accuracy of 61.12% when trained on a dataset of more generalized images. It is the only CNN that showed greater accuracy when trained with a larger dataset rather than the more specialized Carriage dataset. MobileNetV2 achieved a 72.28% accuracy training with the Carriage dataset then the Studio dataset. Although this training method did not exceed the MobileNetV2 ILSVRC2012 competition accuracy (74.7%), it demonstrates a method that could be utilized and further refined for waste classification when there is not a large dataset of images available. For this reason, a further contribution from this work will be making the Studio dataset, "RubbishNet" available for this purpose.

SqueezeNet, whilst not achieving a higher accuracy than ShuffleNet or MobileNetV2 in any of the four training conditions is over three times as fast as the other networks. The highest accuracy SqueezeNet achieved is 67.71% when trained on the smaller Carriage dataset. With some improvements this could be utilized in a waste management task where computational resources are limited.

As it achieved the highest accuracy, precision, and recall, further work will implement the ShuffleNet network trained with the Carriages dataset. It will be developed into a detection algorithm to locate waste in train carriages. This will be used with a manipulator to perform an automated cleaning service.

ACKNOWLEDGMENT

The authors would like to thank the Greater Anglia train presentation staff at London Liverpool Street station and the West Midlands train presentation staff at Euston for their support collecting images of waste items on trains.

REFERENCES

[1] *Office of Rail and Road, "UK rail industry financial information 2017-18." 2019.*

[2] *S. Noorani and M. Fernandes, "Evaluation of convolutional neural networks for waste identification," in Proceedings of the International Conference on Computing Methodologies and Communication, ICCMC 2017, 2018, pp. 204–207.*

[3] *Y. Chu, C. Huang, X. Xie, B. Tan, S. Kamal, and X. Xiong, "Multilayer Hybrid Deep-Learning Method for Waste Classification and Recycling," Comput. Intell. Neurosci., 2018.*

[4] *C. Bircanoglu, M. Atay, F. Beser, O. Genc, and M. A. Kizrak, "RecycleNet: Intelligent Waste Sorting Using Deep Neural Networks," 2018 IEEE Int. Conf. Innov. Intell. Syst. Appl. INISTA 2018, 2018.*

[5] *A. N. Kokoulin, A. I. Tur, and A. A. Yuzhakov, "Convolutional neural networks application in plastic waste recognition and sorting," Proc. 2018 IEEE Conf. Russ. Young Res. Electr. Electron. Eng. ElConRus 2018, vol. 2018-Janua, pp. 1094–1098, 2018.*

[6] *Z. Lu, S. Rallapalli, K. Chan, and T. La Porta, "Modeling the resource requirements of convolutional neural networks on mobile devices," Proc. 2017 ACM Multimed. Conf., pp. 1663–1671, 2017.*

[7] *S. Rallapalli et al., "Are Very Deep Neural Networks Feasible on Mobile Devices ?" .*

[8] *L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 4, pp. 594–611, 2006.*

[9] *J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," Adv. Neural Inf. Process. Syst., vol. 27, pp. 3320–3328, 2014.*

[10] *M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 4510–4520, 2018.*

[11] *X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 6848–6856, 2018.*

[12] *F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level Accuracy with 50x Fewer Parameters and <0.5mb Model Size." pp. 1–13, 2016.*

[13] *M. Elara, M. Ilyas, A. Lakshmanan, A. Le, and B. Ramalingam, "Cascaded Machine-Learning Technique for Debris Classification in Floor-Cleaning Robot Application," Appl. Sci., vol. 8, no. 2649, pp. 1–19, 2018.*

[14] *NVidia Developer, "Deep Learning Frameworks," NVidia Developer, 2018. [Online]. Available: https://developer.nvidia.com/deep-learning-frameworks.*

[15] *C. Elkan, "Evaluating Classifiers." University of California San Diego, 2012.*

[16] *D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," pp. 1–15, 2014.*

[17] *O. Russakovsky et al., "ImageNet Large Scale Visual Recognition Challenge." 2015.*

[18] *G. P. Visa and P. Salembier, "Precision-Recall-Classification Evaluation Framework: Application to Depth Estimation on Single Images," in Computer Vision – ECCV 2014, 2014, pp. 648–662.*

[19] *A. Matsumoto and K. Yanai, "Continual Learning of Image Translation Networks Using Task-Dependent Weight Selection Masks," in Pattern Recognition, 2019, pp. 129–142.*