11-785 FINAL REPORT: DETECTING DISTRACTED DRIVERS

Natnael Daba
ndaba@andrew.cmu.edu

Bereket Frezgiy
bfrezgi@andrew.cmu.edu

ABSTRACT

According to World Health Organization (WHO), 1.25 million people die yearly due to road traffic accidents worldwide and the figure has been increasing steadily for the past few years. Close to fifth of these accidents are caused by distracted drivers. Due to this, there is a pressing need to come up with solutions that can improve these alarming statistics. Deep learning based vision systems have garnered attention due to their effectiveness in solving vision problems. In this paper, we propose the use of a special type of convlutional neural network called MobileNet as a deep learning architecture for detecting the state of a driver given a dataset of 24,424 images of drivers taken in a car. We experiment with the two versions of MobileNet and compare their performance with other complex models applied to the same problem with the same dataset. Our findings show that MobileNet-V2 performs better than existing complex models with less running time, less number of parameters and better accuracy.

1 INTRODUCTION

According to World Health Organization (WHO), 1.25 million people die yearly due to road traffic accidents worldwide and the figure has been increasing steadily for the past few yeas. Close to fifth of these accidents are caused by distracted drivers [5]. Mobile phone use while driving (MPUWD) is an increasingly common form of distracted driving. Given its widespread prevalence, it is important for researchers to identify factors that may predict who is more likely to engage in this risky behavior [10].

In an attempt to improve the alarming statistics, numerous works have been done to determine if the driver is engaged in a risky behavior or not. Among these works, deep learning based computer vision systems have garnered a lot of attention due to their effectiveness in determining the state of a driver. For more details, see [3],[4],[5].

In this paper, we propose the use of a special type of convolutional neural network called MobileNet as a deep learning architecture for detecting (via a 10 class classification task) the state of a driver. We chose this architecture because it is light weight(in terms of running time and complexity) and thus is suitable for being used in embedded vision applications. To the best of our knowledge, this architecture has never been applied to this problem.

1.1 DATA

The dataset we used for this project, provided by the State Farm group in collaboration with Kaggle [6], contains 22,424 labeled images of drivers each taken in a car with a driver doing different activities (texting, eating, talking on the phone, makeup, reaching behind, etc). The dataset is primarily used to asses the effectiveness of computer vision to spot distracted drivers. Fig. 1 and 2 below show a visualization of the data and histogram of the classes of the data respectively.

2 Related Work

An end-to-end deep learning solution for detecting distracted drivers was proposed in [4]. In the paper, a pre-trained VGG-19 network is used as a feature extractor. Their model achieved a classification accuracy of 95% after being tested with leave-one-driver-out cross validation method to ensure generalization. A similar approach is used in [8] but instead of using a pre-trained model, the authors used a custom built CNN as a classifier.

[5] proposes a genetically weighted ensemble of convolutional neural networks where the authors show that a weighted ensemble of classifiers using a genetic algorithm yields a better classification confidence. The authors also study the effect of different visual elements in distraction detection by means of face and hand localization's, and skin segmentation.

An interesting solution is proposed in [3]. In this paper, a deep learning based real-time distracted driver detector is built that makes use of four deep convolutional neural networks including VGG-16, AlexNet, GoogleNet, and residual network. The system is evaluated on an embedded graphic processing unit platform. In addition, the authors also developed a conversational warning system that alerts the driver in real-time when he/she does not focus on the driving task.

Some works also consider other techniques in addition to deep learning to get better results. For instance, in [7], a multi-modal vehicular and physiological sensor data is used where deep learning is applied to the fused multi-modal data rather than each modality being treated as a different feature. In [9], traditional handcrafted features paired with a Support Vector Machine classifier are contrasted with deep learning approaches. [11] uses a fuzzy logic together with a new machine learning algorithm that defines driver performance in lane keeping and speed maintenance on a specific road segment.

3 Methodology

To spot check algorithms on the problem to see if we have a useful basis for modeling our classification problem, we first chose ResNet-152 to obtain a baseline result. In this project, we decided to use top-1 accuracy, total number of learnable parameters, and training time per epoch for measuring performance and comparing models. Table 3 below shows how this baseline model performed.

We then used the first version of MobileNet [1] hereinafter referred to as MobileNet-V1 as our network architecture to perform the task of classifying the state of a driver given a dataset of 2D dashboard camera images. However, training the MobileNet-V1 was very straight forward. We reached 95% accuracy within the first seven epochs without much effort. We did not do much tuning on MobileNet-V1 except for learning rate.

Therefore, we next considered MobileNet-V2 [2]. The major hyper-parameter tuning we made was on MobileNet-V2. We chose to consider MobileNet-V2 because according to the experiments performed on a different problem in the original paper of MobileNet-V2, the authors reported a significant reduction in the number of learnable parameters and multiply-accumulate operations compared to MobileNet-V1 and other complex models [2].

We finally compared the performance of MobileNet-V2 with MobileNet-V1 and other complex models which are used to solve the same problem with the same dataset.

3.1 Baseline

The task of determining the state of a driver is a multiclass classification problem, where the input is a dashboard image of a driver and the output is a label classifying the driver being in one of the ten possible states: texting, eating, talking on the phone, makeup, reaching behind, etc [6]. Our baseline model is a deep convolutional neural network based on ResNet-152 with the final set of fully connected layers replaced with projection layers with ten as number of output neurons. The Pre-trained ResNet-152 model was fine-tuned as a basis for our classifier. The final fully connected layer was removed and replaced by a linear layer with binary outputs.

Weights from a Pre-trained model were loaded and the model was fine-tuned end-to-end using a default Adam optimizer. Before training, some pre-processings were performed on the original image data. These pre-processings include data augmentation and transformation.

Figure 1: Sample images and the driver's corresponding state

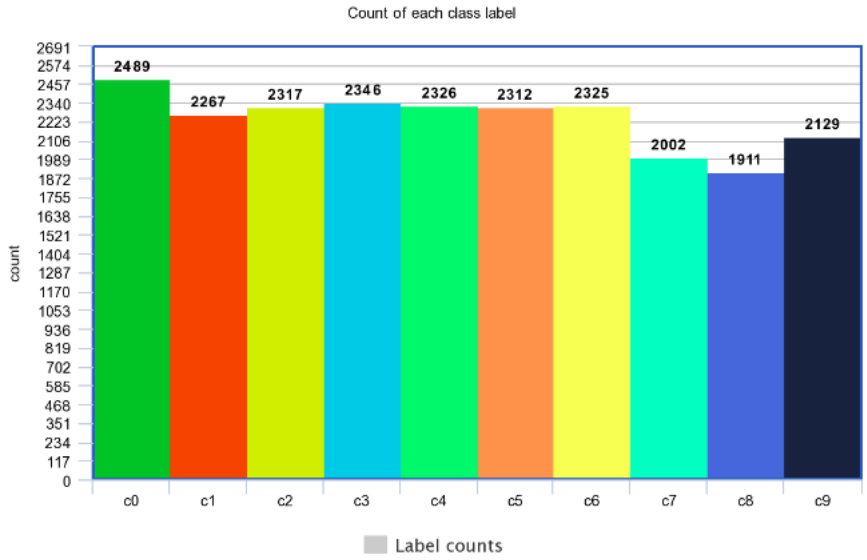

Figure 2: Distribution of classes

3.2 MOBILENET

The baseline image classification was improved by using a different CNN architecture called MobileNet based on [1] and [2]. The first paper introduces a class of efficient models called MobileNets for mobile and embedded vision applications. MobileNets are based on a streamlined architecture that uses depthwise separable convolutions to build light weight deep neural networks. The second paper introduces an improved version of MobileNet-V1 by incorporating an inverted residual structure where the shortcut connections are between the thin bottleneck layers. The intermediate expansion layer uses lightweight depthwise convolutions to filter features as a source of non-linearity. Fig.3 shows the two main blocks used in MobileNet-V1 and MobileNet-V2. Fig 4. shows the architecture of both MobileNet-V1 and MobileNet-V2.

3.3 EXPERIMENTATION

We experimented with MobileNet-V2 using two tunable parameters. These tunable hyperparameters (from the model) are depth multiplier (alpha) and expansion factor (**t**). The depth multiplier (alpha) changes how many channels are in each layer. For example, using a depth multiplier of 0.5 will

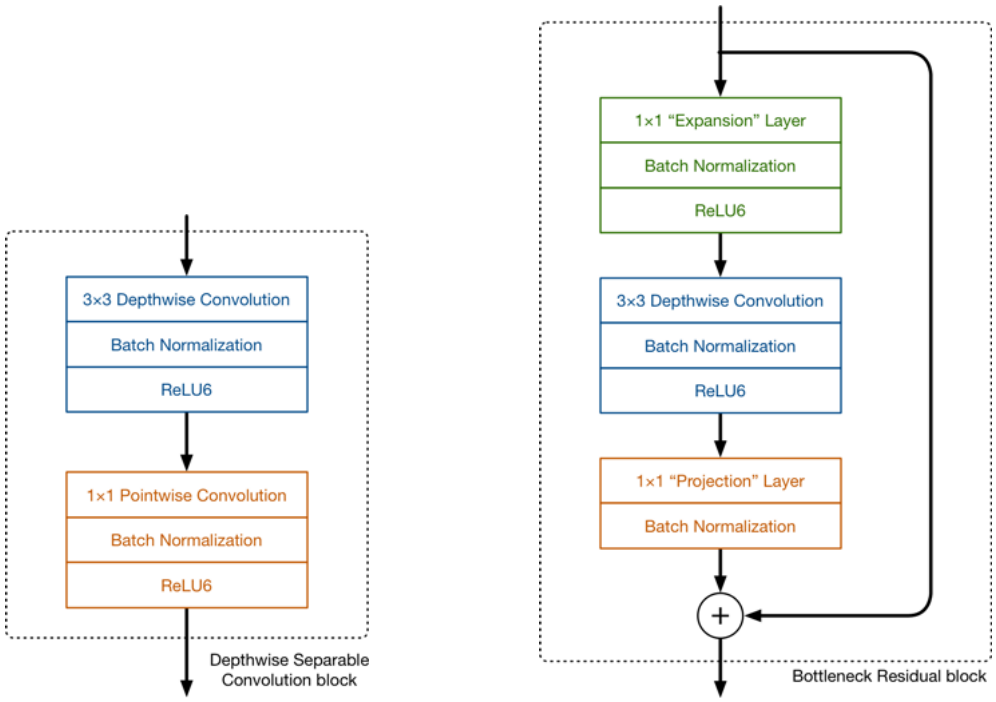

Figure 3: The main building block of MobileNet-V1(left) and MobileNet-V2(right)

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Figure 4: Architecture of MobileNet-V1(left) and MobileNet-V2(right)

halve the number of channels used in each layer. The expansion factor (t) specifies exactly by how much the data gets expanded before it goes into the expansion layer as shown in Fig. 4.

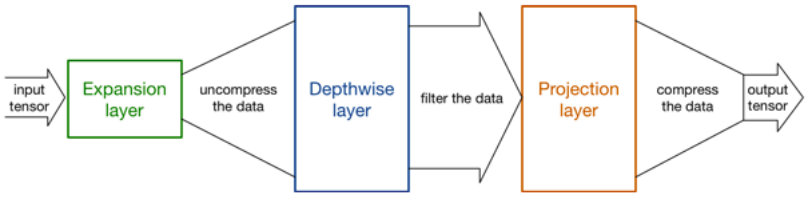

Figure 5: Data expansion and compression in each bottleneck residual layer

4 RESULTS

The classification accuracy, training time per epoch, and number of learnable parameters (in millions) are reported in the following tables. Table 1 shows the result of tuning the depth multiplier (alpha) by keeping the expansion factor constant (t = 6). Table 2 shows the result of tuning the expansion factor (**t**) by keeping the depth multiplier constant (alpha = 1). Figure 6 shows the loss curves of both MobileNet-V1 and MobileNet-V2 on the validation dataset. In table 3, we compare the performance of one model that was used on the same dataset and the same task with that of our model. Figure 7 shows the Receiver Operating Characterstics (ROC) curve and confusion matrix of the final model i.e. MobileNet-V2.

	alpha = 0.5	= 0.7	= 0.9	= 1.2	= 1.4
Top-1 Accuracy	96.65%	97.03%	97.25%	97.83%	97.54%
Running time (per epoch)	5 min	12 min	11 min	13 min	14 min
Number of trainable parameters(millions)	0.697	1.2	2.23	3.18	4.3

Table 1: Performance of MobileNet-V2 for different values of depth multiplier (alpha)

	t = 3	t = 4	t = 5	t = 6	t = 7	t = 8
Top-1 Accuracy	96.65%	96.79%	97.01%	97.25%	98.0%	98.0%
Running time	8 min	9.1 min	9.3 min	11 minutes	11.6 min	13 min
Number of trainable parameters(millions)	1.33	1.72	2.03	2.24	2.73	3.22

Table 2: Performance of MobileNet-V2 for different values of expansion factor(t)

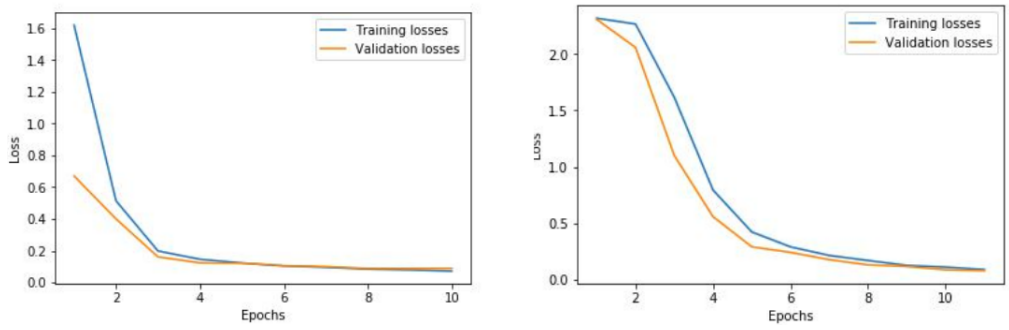

Figure 6: Validation loss curves of MobileNet-V1(left) and MobileNet-V2(right).

5 CONCLUSION AND FUTURE WORK

This paper explored the application of a light weight deep learning architecture to an important computer vision problem. The architecture is suitable to be implemented in an embedded device. The embedded device can be used to identify a distracted driver and help reduce accidents caused

	Top-1 Accuracy	Running time(per epoch)	Number of trainable parameters	Epochs
MobileNet-V1	99.125%	6.23 minutes	3.24	20
MobileNet-V2	99.396%	11.64 minutes	2.20	20
ResNet152	82.101%	13 minutes	60.2	20
VGG16	99.46%	15 minutes(with 4 GPUs)	138.35	50

Table 3: comparing performance of models

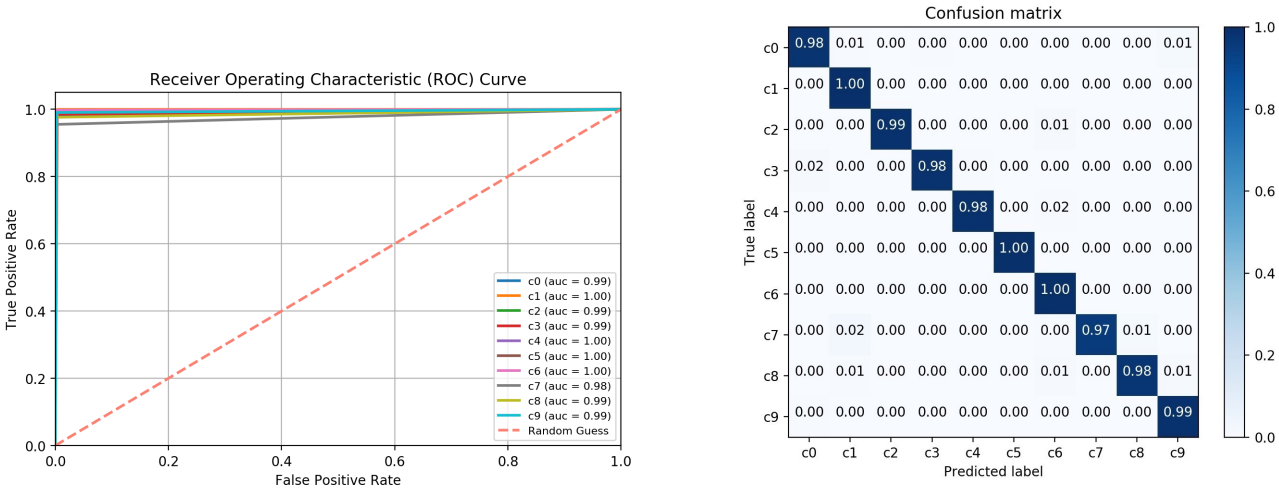

Figure 7: ROC curve(left) and Confusion matrix of the final model MoblieNetV2

by distracted driver. Developing an embedded vision application that makes use of a state-of-the-art deep learning architecture is difficult. In this paper we proposed a simpler CNN architecture called MobileNet that is suited for embedded vision applications and that can perform the task of detecting distracted drivers with a comparable performance with other complex models. We first used the first version of MobileNet as a model. However, since we found the result we expected without much effort, we decided to tryout MobileNetV2, an improved version of the first version.

We experimented with MobileNetv2 by tuning hyperparameters(from the model): Depth multiplier(alpha) and expansion factor(t). The optimal values of alpha and t that gave better results were alpha = 0.9 and t = 6. Tables 3 and 4 clearly show the tradeoff between model complexity and accuracy. We finally compared our model (i.e. MobileNetV2 with alpha = 0.9 and t = 6) with other complex (in terms of the number of learnable parameters) that were used on the same dataset for performing the same task. We can see that MobileNetV2 gave a comparable accuracy (99.12%) with a fewer number of parameters(2.2 million) when compared to the best model(VGG16 with transfer learning) with accuracy(99.396%) with a staggering 138.35 million parameters.

In the future, we will try to implement this architecture on a small embedded vision system with an integrated camera that can perform the detection and in addition contains a feedback or alarm system that can notify the driver or other concerned body. Moreover, this work can further be improved by using an ensemble of MobileNet-V2 with other light weight architectures that are suitable for embedded vision applications.

REFERENCES

[1] A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. *Mobilenets: Efficient convolutional neural networks for mobile vision applications.* arXiv preprint arXiv:1704.04861, 2017.

[2] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: *Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation.* arXiv preprint arXiv:1801.04381 (2018).

[3] D. Tran, H. Manh Do, W. Sheng, H. Bai and G. Chowdhary, *Real-time detection of distracted driving based on deep learning*, IET Intelligent Transport Systems, vol. 12, no. 10, pp. 1210-1219, 2018. Available: 10.1049/iet-its.2018.5172.

[4] A. Koesdwiady, S. Bedawi, C. Ou and F. Karray, *End-to-End Deep Learning for Driver Distraction Recognition*, 2019. URL https://link.springer.com/chapter/10.1007/978-3-319-59876-5_2

[5] H. Eraqi, Y. Abouelnaga, M. Saad and M. Moustafa, *Driver Distraction Identification with an Ensemble of Convolutional Neural Networks*, 2019. URL https://arxiv.org/abs/1901.09097

[6] Kaggle Inc. State farm distracted driver detection, 2016. Available: https://www.kaggle.com/c/state-farm-distracted-driver-detection

[7] S. Lim and J. Yang, *Driver state estimation by convolutional neural network using multimodal sensor data*, Electronics Letters, vol. 52, no. 17, pp. 1495-1497, 2016. Available: 10.1049/el.2016.1393.

[8] P. Horel, P. Tiwari, A. Tiwari, P. Chauhan *Autonomous Distracted Driver Detection using Machine Learning Classifer*, International Journal of Advance Engineering and Research Development, vol. 4, no. 04, 2017. Available: 10.21090/ijaerd.97046.

[9] M. Hssayeni, S. Saxena, R. Ptucha and A. Savakis, *Distracted Driver Detection: Deep Learning vs Handcrafted Features*, Electronic Imaging, vol. 2017, no. 10, pp. 20-26, 2017. Available: 10.2352/issn.2470-1173.2017.10.imawm-162.

[10] M. Sween, A. Ceschi, F. Tommasi, R. Sartori and J. Weller, *Who is a Distracted Driver? Associations between Mobile Phone Use while Driving, Domain-Specific Risk Taking, and Personality*, Risk Analysis, vol. 37, no. 11, pp. 2119-2131, 2017. Available: 10.1111/risa.12773.

[11] Aksjonov, Andrei & Nedoma, Pavel & Vodovozov, Valery & Petlenkov, Eduard & Herrmann, Martin. (2018). *Detection and Evaluation of Driver Distraction Using Machine Learning and Fuzzy Logic.* IEEE Transactions on Intelligent Transportation Systems. PP. 1-12. 10.1109/TITS.2018.2857222.