Distracted Driver Detection and Classification using Convolutional Neural Network

Vinay Kakkad AU1940012, Bhavya Peshavaria AU1940204,
Mananshi Vyas AU1940289, Nandini Bhatt AU1940283

***Abstract*—Number of road accidents is continuously increasing in last few years worldwide. As per the survey of National Highway Traffic Safety Administrator, nearly one in five motor vehicle crashes are caused by distracted driver. We attempt to develop an accurate and robust system for detecting distracted driver and warn him against it. Motivated by the performance of Convolutional Neural Networks in computer vision, we present a CNN based system that not only detects the distracted driver but also identifies the cause of distraction. ResNet50 architecture is modified for this particular task and various regularization techniques are implied in order to improve the performance. We infered that among various CNN architectures ResNet50 outperformed with an accuracy of 90%.**

***Index Terms*—computer vision, distracted driver, models, classification, convolutional neural networks, resnet50, vgg19**

I. Introduction

A camera is mounted on the car dashboard that captures real-time images of the driver. The dataset is taken from the ”State Farm Distracted Driver Detection Competition” on Kaggle. A classification output of the ten different postures are shown below. For the problem, we used Convolutional Neural Network(CNN) architecture ResNet50 for classification by detecting face and hand features. However, we did face computational complexity and memory allocation issues which we overcame by reducing the size of the images in order to increase accuracy for autonomous driving.

Fig. 1: Different classes

II. Literature Survey

Neural Networks provide an architecture wherein the interconnected layers which have some initial weight, updates these weights as networks gets trained. It indicates that the model has learnt the features of the dataset. This applies to the case on CNN where the hidden layers are used for training the model. The works by Bhakti Baheti *et al.* [1] studies the VGG16 architectures and uses it for distracted driver detection. Karen Simonyan *et al.* [2] also presented how CNN could be used for large scale image recognitions using 16-19 weight layers. The report thus attempts to use an architecture subject to Deep convolutional network approaches.

III. Implementations

A. Dataset

We obtained our dataset from Kaggle which is a collection of more than 22,424 unique images captured from a dashboard camera. We believe that these images are captured from a single video of multiple drivers and that could provide us better insights.

B. Problem Statement - Distracted Driver Detection

Our aim is to build and test models which can detect and classify the types of distractions these drivers experience and wish to warn them about the same.

C. Pre-processing Steps

We apply the following pre-processing steps to our data, to suit our analysis better -

1. To reduce the required computational power and fit the requirements of the models, we reduced the image size to 224x224 pixels.
2. We decide to keep the images in the RGB channel as the neural network would be able to capture features such as the position of hand, head, and the rotation of neck using the RGB channels.

D. Exploratory Data Analysis

We plot histograms to check the distribution of data amongst various classes and we infer that the data is not skewed and we also plot a histogram to check the distribution of data amongst 17 drivers to check if there is any possibility of any feature overlapping the other due to its frequent occurence. The results of both are shown below.

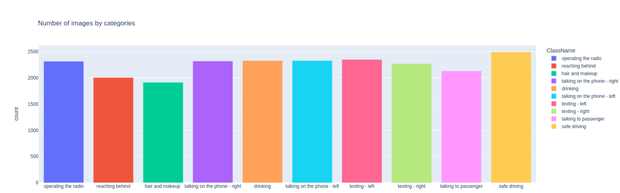


Fig. 2: EDA Results: Distribution of data among various classes

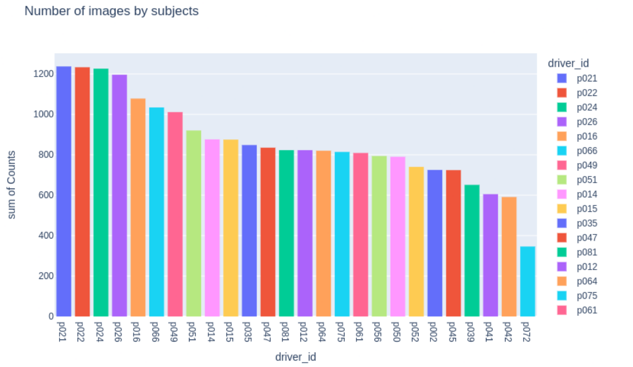


Fig. 3: EDA Results: Distribution of images among 17 drivers

E. **Classification Models**

1. VGG19 (Figure 7)
   a. Suggested that deeper CNN with smaller kernel gives better results than shallow CNN with large kernel size.
   b. Gave a layer of abstraction of block level architecture where each block has predefined layers packed in it. The idea proposed is to stack such blocks to make network deper to make it more complex.
2. Resnet50 (Figure 8)
   a. In VGG, each block depends on the output of the previous one. A minor deviation of learning of one block could lead to divergence of whole model from the actual mapping.
   b. To prevent this, ResNet model proposed to merge the input as well as prediction of the previous block as input to the next one. This ensures that the deviation is minimum.
   c. One benefit of this approach is it prevents gradient diminishing/exploding issue as there is always a addition term of the previous input while applying chain rule for backpropogation.

F. **Fine Tuning**

1. Number of epochs
   a. For ResNet model we reduced the number of epochs to 60 to avoid overfitting.
   b. For VGG model we have kept the number of epocs as 50
2. Steps Per Epochs
   a. For both ResNet and VGG, we have kept the steps per epoch as 100.

IV. Results

Figure 4 and 5 shows how the accuracy changes after passing of each epoch for ResNet50 and VGG19 respectively. In the plots shown below, the orange line depicts validation loss and the blue line depicts training loss. We can observe how the accuracy changes significantly during the first few epochs and how ResNet50 provides an accuracy of around 0.93 at the end of 60 epochs and VGG19 provides an accuracy of around 0.83 at then end of 50 epochs. The confusion metrics shows how well the models predicts and the gap between the true label and the predicted label is less.

TABLE I: Results

Model	Accuracy	Loss
ResNet50	93%	0.23
VGG19	83%	0.51

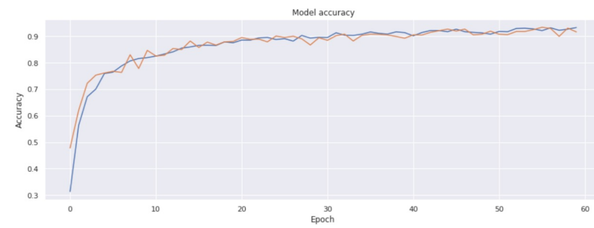


Fig. 4: Graph of loss v/s epoch for ResNet50

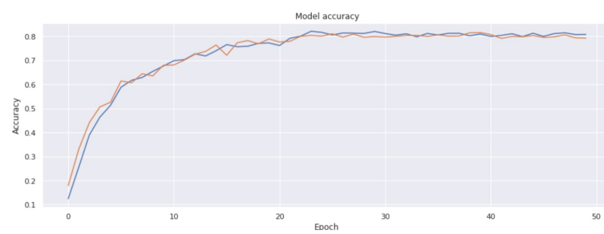


Fig. 5: Graph of loss v/s epoch for VGG19

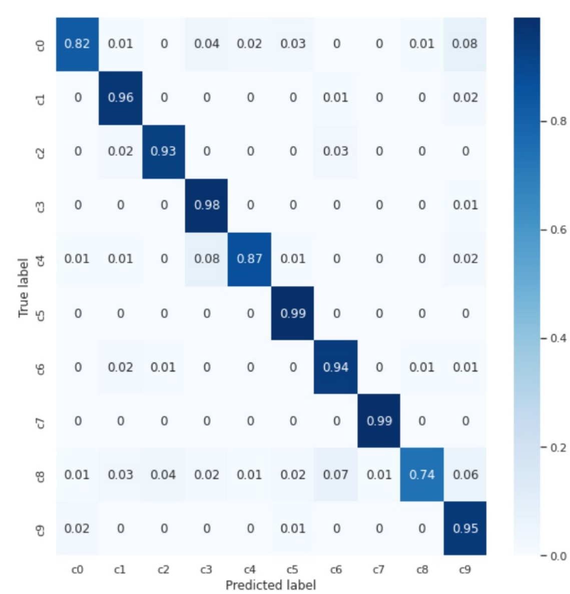


Fig. 6: Confussion Matrix of ResNet-50

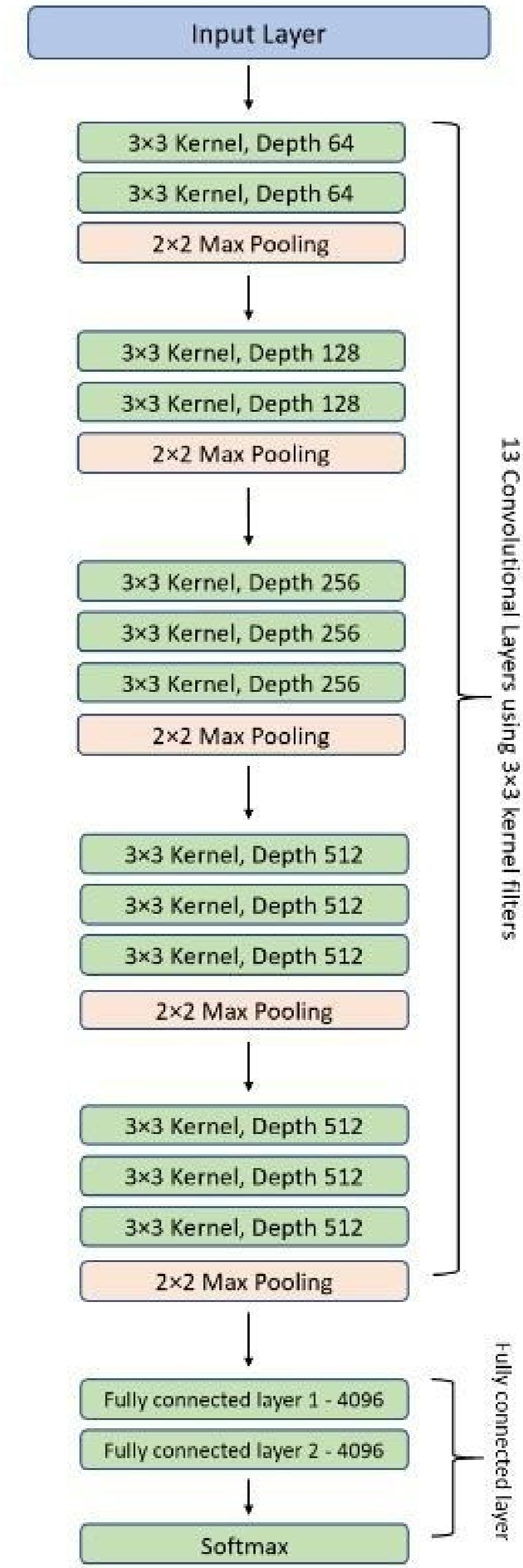
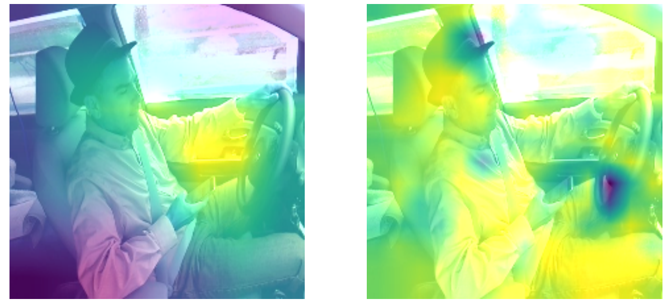
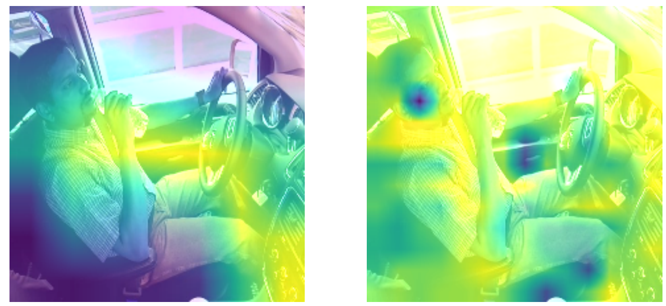


Fig. 7: VGG19 Architecture



(a) Grad-CAM (b) Occlusion Sensitivity

Fig. 9: Explainable AI on c1



(a) Grad-CAM (b) Occlusion Sensitivity

Fig. 10: Explainable AI on c6

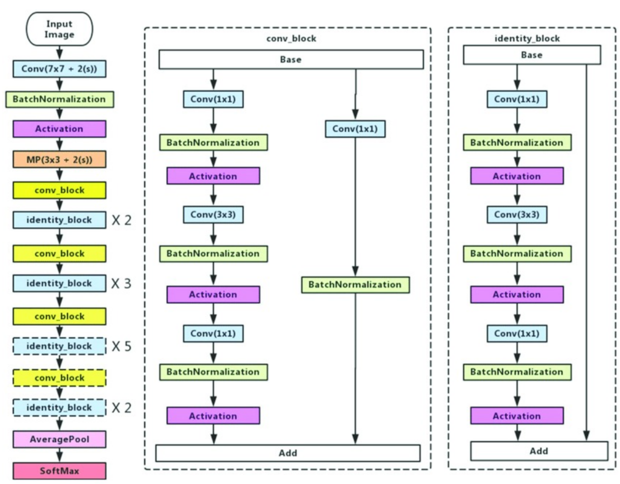


Fig. 8: ResNet50 Architecture

V. Conclusions

We conclude that the ResNet50 performs better on our dataset with and accuracy of around 93 percent compared to VGG-19 which gave an accuracy of 83 percent.Even though VGG-19 has more number of parameter, most of the parameters are a result of the fully connected layers, whereas ResNet is more deeper and thus provides a better feature extraction. ResNet-50 also has a identity block which also helps in the problem of vanishing or exploding gradients. Due to these reason we feel that ResNet50 gave us a better accuracy. We also implemented Explainable AI (Figure 9 and Figure 10), a technique used to understand model prediction on image data, we studied model predictions of the image features in each of the ten classes.

REFERENCES

[1] B. Baheti, S. Gajre, and S. Talbar, "Detection of distracted driver using convolutional neural network," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 1145–11 456. DOI: 10.1109/CVPRW.2018.00150.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[3] P. Canuma, *Image classification: Tips and tricks*, Nov. 2021. [Online]. Available: https://neptune.ai/blog/image-classification-tips-and-tricks-from-13-kaggle-competitions.

[4] *State farm distracted driver detection*, https://www.kaggle.com/competitions/state-farm-distracted-driver-detection, 2016.

[5] *Resnet50 architecture*, https://www.researchgate.net/publication/331364877/figure/fig3/AS:741856270901252@1553883726825/Left-ResNet50-architecture-Blocks-with-dotted-line-represents-modules-that-might-be_W640.jpg, Feb. 2019.