Distracted Driver Detection

By:
Zamzam Alsarayrah
Ganesh Morye
Rahul Mukerjee

Data Scientist
General Assembly

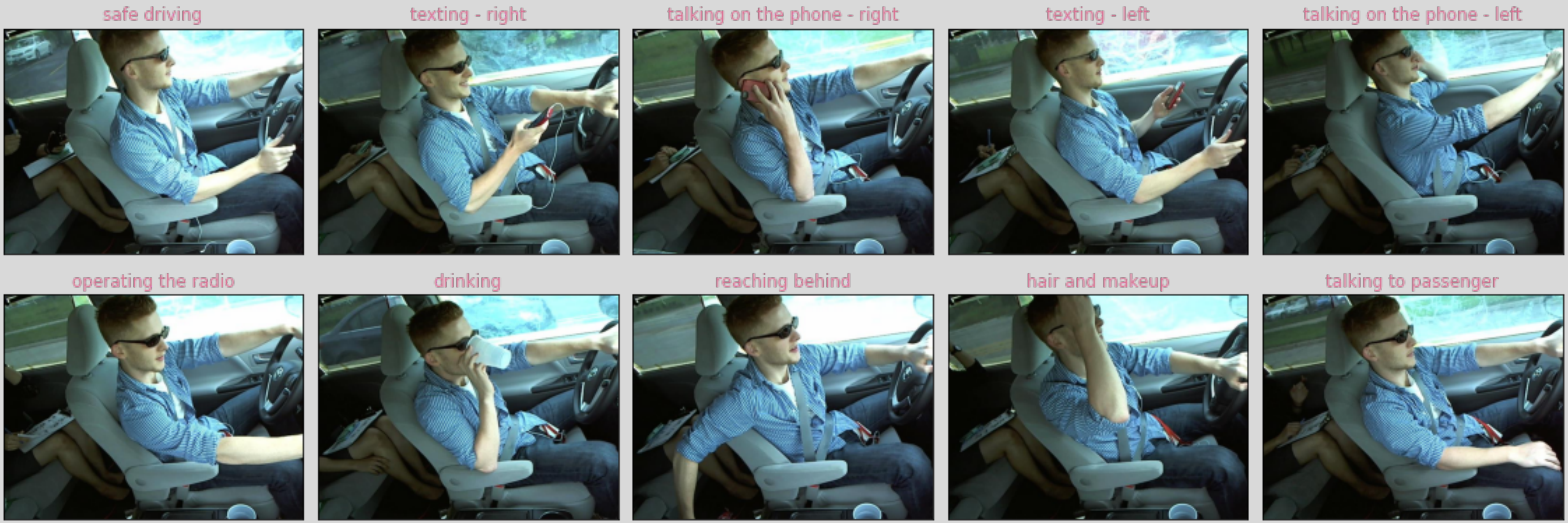
INTRODUCTION:

- People are often multi-tasking while driving
- Most common activities during driving are talking on phone, texting, eating, applying makeup
- These "distractions" make the driver less attentive and compromise the safety of the driver, passengers, bystanders, and others
- The United States Department of Transportation states that one in five car accidents are caused by distracted drivers, meaning that distracted driving causes injuries to 425,000 people and claims the lives of 3000 others every year[1].
- State Farm is a nationwide insurance company that help people manage the risks of everyday life, recover from the unexpected, and realize their dreams.
- State Farm released this dataset as a Kaggle challenge to test whether dashboard cameras can automatically detect drivers engaging in distracted behaviors to better insure their customers

PROBLEM STATEMENT :

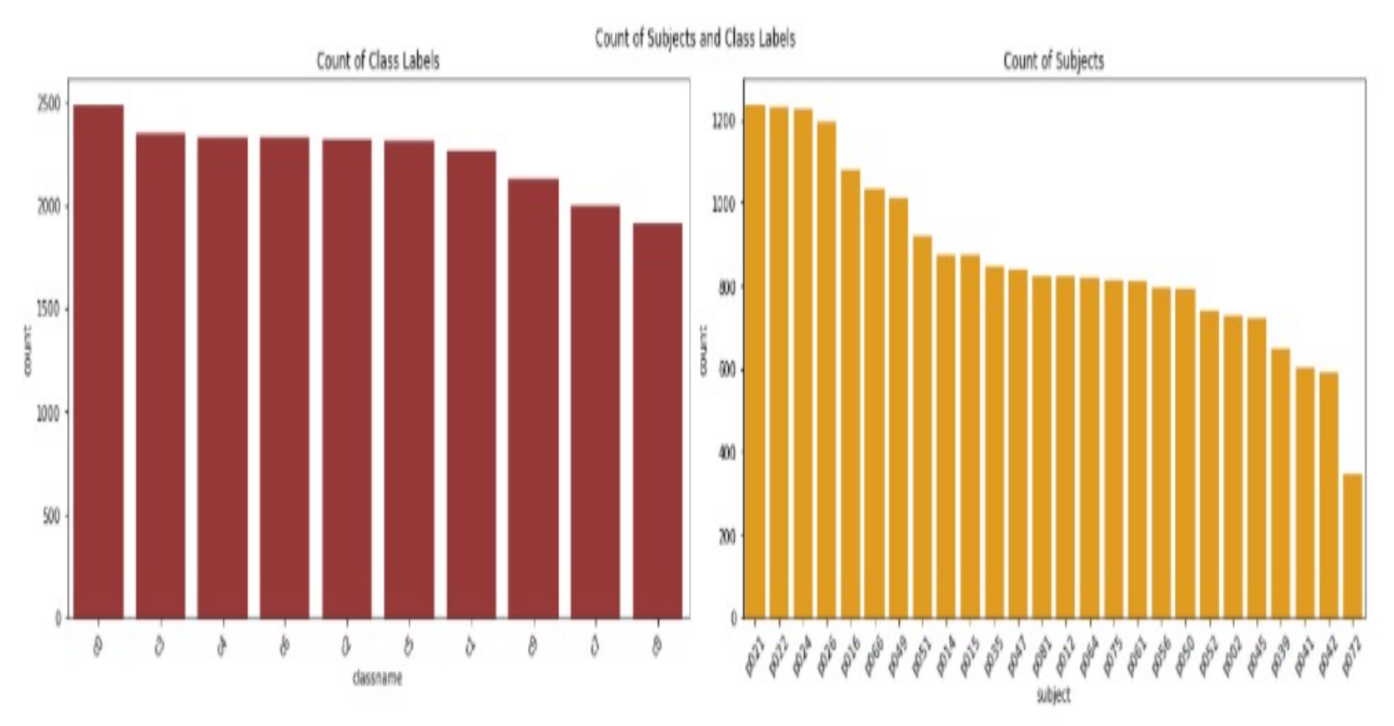
To predict the likelihood of what the driver is doing in each picture by classifying the images in one of the following 10 classes.

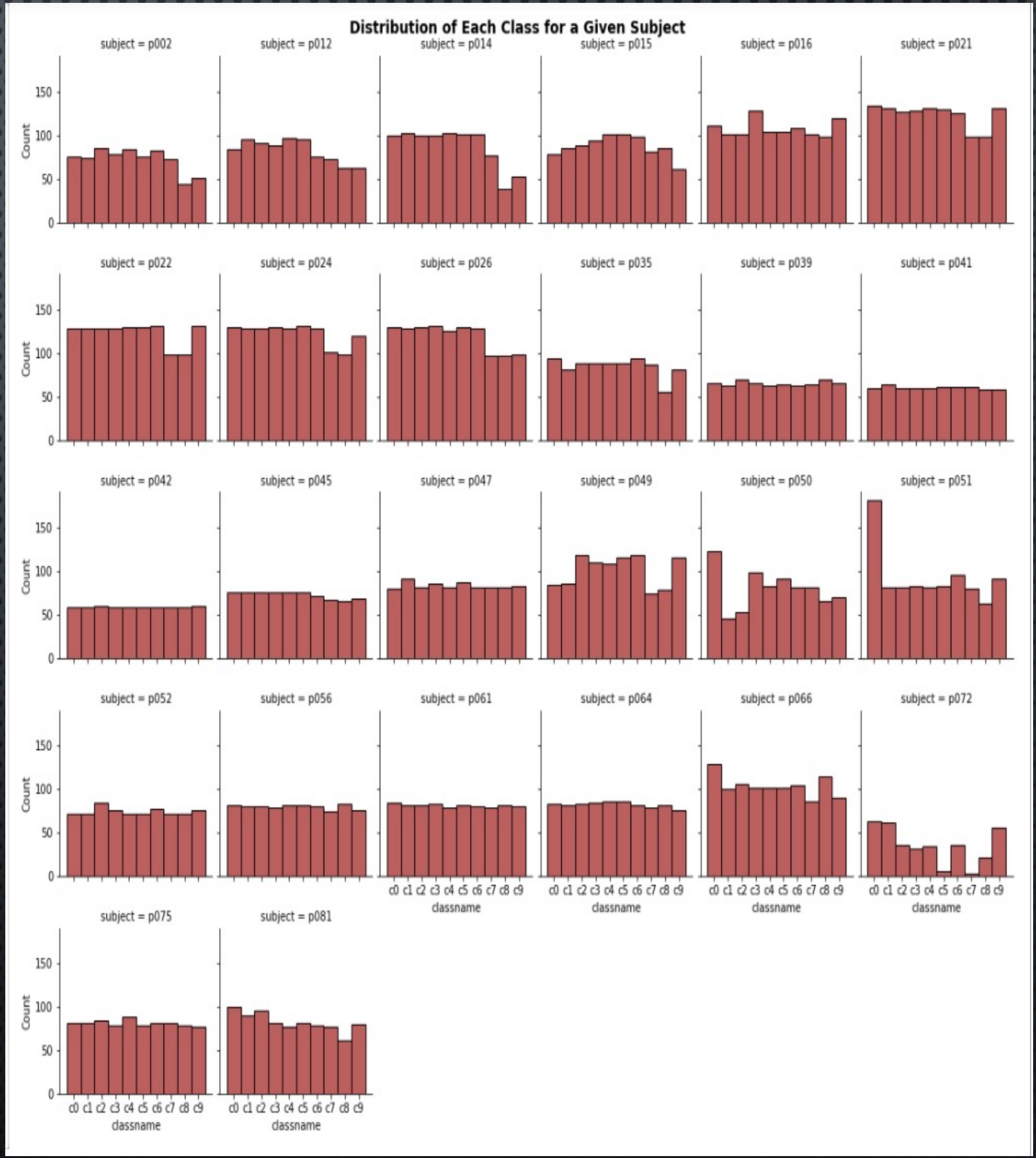
Dataset for this project was downloaded from Kaggle and predictions were uploaded to score the models. The models are scored on ***LogLoss*** *metric*



DATASET / EXPLORATORY VISUALIZATION

- TRAINING DATASET: 22k Images of 26 subjects
- TESTING DATASET: 79k Images
- Subjects appear in either the training or testing dataset but not in both
- Labels are evenly distributed
- Image count of subjects widely varies





ALL THE 26 SUBJECTS IN THE TRAINING DATASET

METHODS AND MODELS:

- Neural Network
- Convolutional Neural Networks(CNN)
- Image Data Generator
- pre-trained VGG16 & VGG19
- pre-trained EfficientNetB0

BASE MODEL STRUCTURE :

Our base model is a simple CNN model :

1- we use 2 CNN layers:
  a- layer 1 with 50 filters, (3,3) kernel size.
  b- layer 2 with 16 filters, (3,3) kernel size and relu activation function.

2- after that we use 2 MaxPooling with (2,2) pool size.

3- two Dense layers, the first one with 128 neurons and the second one with 64 neurons, both use Relu activation function.

4- Output layer with 10 neurons and softmax activation function.

5- Two Base models are generated
Regular train_test_split
GroupKFold train_test_split.

IMAGE DATAGENERATOR

- Image augmentation is a technique to expand your dataset and add variations to the images which are not captured in the original dataset
- Apply random transformations like rotation, shear, horizontal and vertical offsets etc., to the original train images
- Results in multiple transformed copies of the same image
- Variations in the dataset allows the model to generalize better and it becomes more robust

PRE-TRAINED MODELS

- What is a Pre-trained Model?
  - A pre-trained model is a model created by some one else to solve a similar problem. We use this model as a starting point.
- Why use Pre-trained Models?
  - To "transfer" the learning from the pre-trained model to our model to avoid having to train our model from scratch
- How to use a Pre-trained Models?
  - Define the objective of our problem to identify the class of pre-trained models to use
  - Use a pre-trained model which was trained for a similar problem. For e.g., Image classification pre-trained model for an Image classification problem.

Transfer Learning

- Using a pre-trained model to our specific problem is called **Transfer Learning**
- Use of pre-trained models takes 3 different forms:
  - Feature Extraction:- Directly use the weights and architecture and apply that learning to our problem. The only modification that would be required in this case is changing the output layer depending on our class labels
  - Transfer Learning:- Use the weights and architecture of the convolutional layers and train the user-defined FNN layers
  - Fine-tuning:- Tweak the already trained convolutional layers and train the model.

PRE-TRAINED MODELS USED IN THE STUDY

- All the pre-trained models used in the study were trained on the imagenet dataset of more than 1 million images in 1000 different classes
- VGG16
  - developed at university of oxford
  - 16 layers and 138,423,208 million trainable parameters
- VGG19
  - Like vgg16 with 19 layers and 143,667,240 trainable parameters
- EFFICIENTNETBO
  - Developed by Google
  - 8 alternative implementations (b0 to b7)
  - 5.3 million trainable parameters

MODELING RESULTS

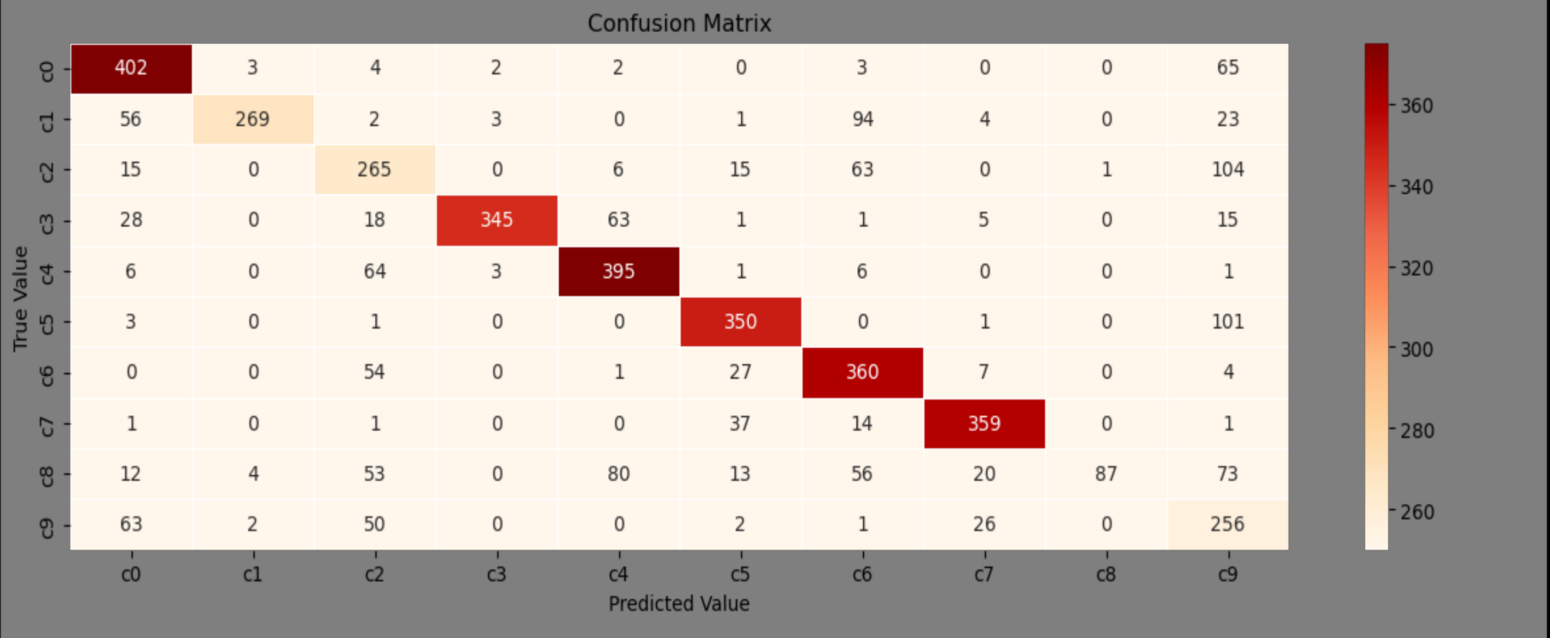
	Model	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy	Kaggle Testing Loss	Model Type
0	base_model_kfold	0.000	1.000	0.005	0.999	26.643	Base-Kfold
1	base_model_gkfold	0.746	0.855	3.525	0.315	23.763	Base-Gkfold
2	best_model_vgg16_dropout_1024	0.193	0.937	0.929	0.690	0.731	VGG16
3	best_model_vgg16	0.181	0.944	0.891	0.718	0.752	VGG16
4	best_model_vgg19_l2_4096_1024_adam_lr	0.512	0.943	1.181	0.719	0.806	VGG19
5	best_model_vgg16_l2_1024	0.433	0.926	1.324	0.645	0.837	VGG16
6	best_model_vgg16_l2_4096_1024_adam_lr	0.574	0.944	1.280	0.720	0.850	VGG16
7	best_model_EfficientNetB0_v2_base	0.090	0.984	0.949	0.686	0.857	EfficientNetB0
8	best_model_EfficientNetB0_v2_augment	0.205	0.952	0.983	0.693	0.878	EfficientNetB0

- 1st model is the base model with random train_test_split gave accuracy of 1 for both training and validation but a testing loss of 26.64
- 2nd model is the base model with Group Kfold train test split reduced the validation accuracy to 30% and improved the testing loss to 23.7
- Our best model performance with pre-trained model was for VGG16 model with a single hidden layer of 1024 nodes. The testing loss was 0.73

CLASSIFICATION REPORT AND CONFUSION MATRIX

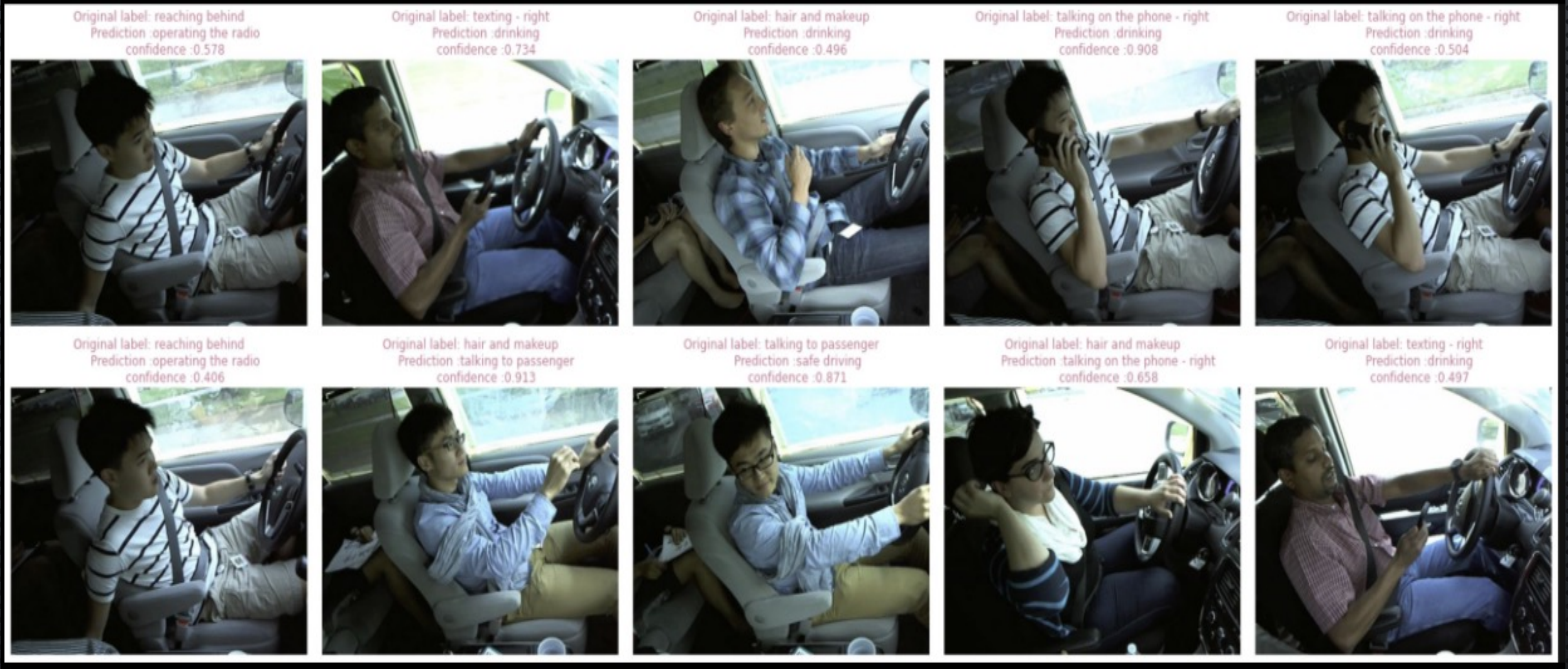
	precision	recall	f1-score	support
c0	0.69	0.84	0.75	481
c1	0.97	0.60	0.74	452
c2	0.52	0.57	0.54	469
c3	0.98	0.72	0.83	476
c4	0.72	0.83	0.77	476
c5	0.78	0.77	0.78	456
c6	0.60	0.79	0.69	453
c7	0.85	0.87	0.86	413
c8	0.99	0.22	0.36	398
c9	0.40	0.64	0.49	400
accuracy			0.69	4474
macro avg	0.75	0.68	0.68	4474
weighted avg	0.75	0.69	0.69	4474

Class labels c8 and c9 have the lowest f1-score and c3 and c7 have the highest



c0 : safe driving	c2 : talking on the phone - right	c4 : talking on the phone - left	c6 : drinking	c8 : hair and makeup
c1 : texting - right	c3 : texting - left	c5 : operating the radio	c7 : reaching behind	c9 : talking to passenger

MISLABELED PREDICTION ANALYSIS



- Use of pre-trained models significantly improved the predictions of the CNN model
- A CNN model using a VGG16 pre-trained model gave testing loss on Kaggle of 0.73
- Model performance along with ImageAugmentation did not offer any noticeable improvement in the scores
- Fine tuning of the convolutional layers was not attempted which could have led to better scores