DISTRACTED DRIVER DETECTION

Sena Çakırlı - 2200356074
Zübeyde Civelek - 2200356814
Mehmet Akif Özgür - 2200356832

The Problem

Driver distraction has been identified as one of the leading causes of accidents on the roads.

Distractions can arise from various factors, such as mobile phone usage, drinking, facial makeup, and social interaction.

To mitigate these risks and enhance road safety, it is crucial to develop effective methods to detect and classify driver distractions in real-time.

Dataset

We have utilized the State Farm Distracted Driver Detection dataset, which is available on Kaggle.

➔ **Labeled Images**

15.200 labeled train images and 3.040 test images.

➔ **Nine Different Distraction Classes**

Texting with right hand, talking on the phone with right hand, texting with left hand, talking on the phone with left hand, operating the radio, drinking, reaching behind, hair and makeup, talking to passenger.

Our Aim

Our aim is to demonstrate the development of a machine learning model using computer vision techniques for classifying nine different types of driver distractions.

We aim to highlight the significance of our automated system in accurately identifying and categorizing distractions in real-time. We seek to contribute to improving road safety by reducing accidents caused by driver distractions.

The Model

We made our model using Keras framework with Convolutional Neural Network (CNN) architecture that comprise of 6 convolution, 6 max-pooling, and 4 dense layers.

Model: "sequential_2"

Layer (type)	Output Shape	Param #
conv2d_10 (Conv2D)	(None, 254, 254, 32)	896
max_pooling2d_10 (MaxPooling2D)	(None, 127, 127, 32)	0
conv2d_11 (Conv2D)	(None, 125, 125, 64)	18496
max_pooling2d_11 (MaxPooling2D)	(None, 62, 62, 64)	0
conv2d_12 (Conv2D)	(None, 60, 60, 128)	73856
max_pooling2d_12 (MaxPooling2D)	(None, 30, 30, 128)	0
conv2d_13 (Conv2D)	(None, 28, 28, 128)	147584
max_pooling2d_13 (MaxPooling2D)	(None, 14, 14, 128)	0
conv2d_14 (Conv2D)	(None, 12, 12, 256)	295168
max_pooling2d_14 (MaxPooling2D)	(None, 6, 6, 256)	0
conv2d_15 (Conv2D)	(None, 4, 4, 512)	1180160
max_pooling2d_15 (MaxPooling2D)	(None, 2, 2, 512)	0
flatten_2 (Flatten)	(None, 2048)	0
dense_26 (Dense)	(None, 1024)	2098176
dense_27 (Dense)	(None, 512)	524800
dense_28 (Dense)	(None, 128)	65664
dense_29 (Dense)	(None, 10)	1290

Total params: 4406090 (16.81 MB)
Trainable params: 4406090 (16.81 MB)
Non-trainable params: 0 (0.00 Byte)

Components

- **Optimizer**
  RMSprop (Root Mean Squared Propagation)
- **Loss Function**
  Categorical Cross-entropy
- **Activation Function**
  ReLU activation function is used for all layers except the final fully connected layer, which Softmax activation function is used.

Training

We used early stopping that depends on validation accuracy value to stop over-fitting. So even though we aim to train our model with 50 epochs, it stopped after 7 epochs.

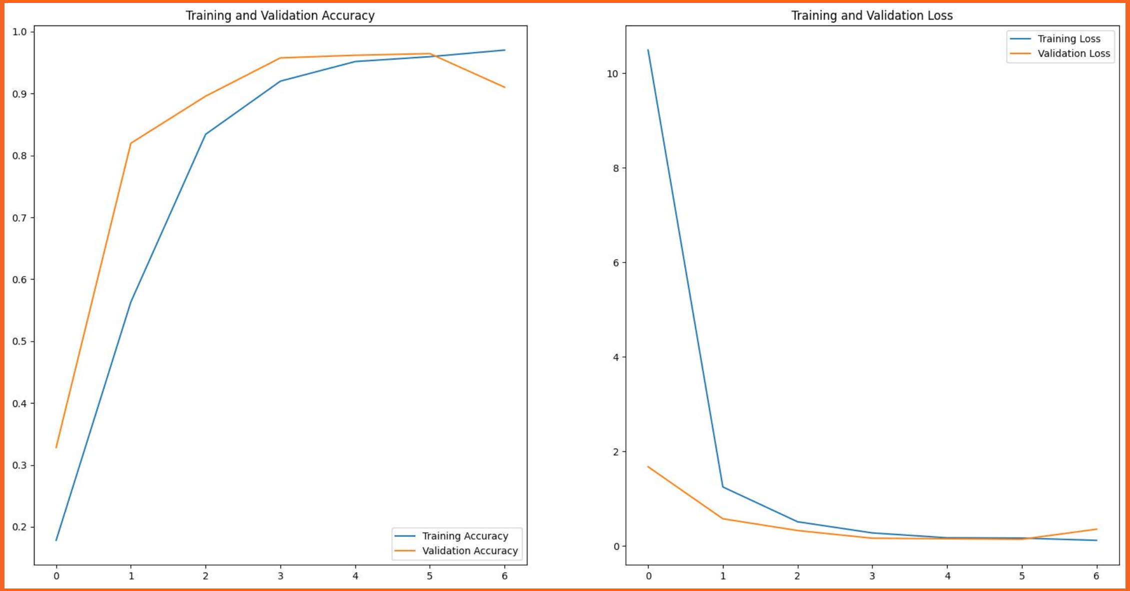
```
history = model.fit(
        train_ds,
        epochs=50,
        validation_data=val_ds,
        callbacks=[early_stopping])
```

✓ 75m 19.8s

```
Epoch 1/50
95/95 [==============================] - 489s 5s/step - loss: 10.4874 - accuracy: 0.1781 - val_loss: 1.6732 - val_accuracy: 0.3280
Epoch 2/50
95/95 [==============================] - 587s 6s/step - loss: 1.2464 - accuracy: 0.5633 - val_loss: 0.5732 - val_accuracy: 0.8197
Epoch 3/50
95/95 [==============================] - 629s 7s/step - loss: 0.5097 - accuracy: 0.8341 - val_loss: 0.3242 - val_accuracy: 0.8957
Epoch 4/50
95/95 [==============================] - 650s 7s/step - loss: 0.2738 - accuracy: 0.9200 - val_loss: 0.1616 - val_accuracy: 0.9576
Epoch 5/50
95/95 [==============================] - 745s 8s/step - loss: 0.1708 - accuracy: 0.9516 - val_loss: 0.1508 - val_accuracy: 0.9618
Epoch 6/50
95/95 [==============================] - 730s 8s/step - loss: 0.1648 - accuracy: 0.9595 - val_loss: 0.1394 - val_accuracy: 0.9645
Epoch 7/50
95/95 [==============================] - 691s 7s/step - loss: 0.1163 - accuracy: 0.9701 - val_loss: 0.3525 - val_accuracy: 0.9102
```

Experimental Results

In training the model with the training dataset the training and validation accuracy gradually increased, while the training and validation loss values gradually decreased.



The best-predicted classes were determined to be **texting-left, drinking** and **reaching behind**. On the other hand, the worst-predicted classes were **hair and makeup** and **talking to passenger**. Furthermore, the most commonly mispredicted classes was **texting-left** when the actual activities were **normal driving**, **hair and makeup** and **talking to passenger**. Further improvements may be required in those area.

THANKS FOR LISTENING!

Now we will show our demo.