

1. Importing All the Required Libraries

```
In [ ]: import tensorflow as tf  
import os  
import cv2  
import numpy as np  
from matplotlib import pyplot as plt
```

WARNING:tensorflow:From p:\MachineLearning\DriverBehaviorAnalysisAndViolationDetection\venv\Lib\site-packages\keras\src\losses.py:2976: The name tf.losses.sparse_softmax_cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_softmax_cross_entropy instead.

2. Loading Data To Pipeline

```
In [ ]: data = tf.keras.utils.image_dataset_from_directory('data',batch_size=16)
```

Found 618 files belonging to 3 classes.

```
In [ ]: data_iterator = data.as_numpy_iterator()
```

```
In [ ]: #get another batch from iterator  
batch = data_iterator.next()
```

```
In [ ]: #images represented as numpyarrays  
batch[0].shape
```

```
Out[ ]: (16, 256, 256, 3)
```

- Class 0 - Drowsy driver
- Class 1 - Normal Driving
- Class 2 - Using mobile Phone

```
In [ ]: fig, ax = plt.subplots(ncols=4, figsize=(20,20))  
for idx, img in enumerate(batch[0][:4]):  
    ax[idx].imshow(img.astype(int))  
    ax[idx].title.set_text(batch[1][idx])
```



3. Scaling Data

```
In [ ]: data = data.map(lambda x,y: (x/255, y))
```

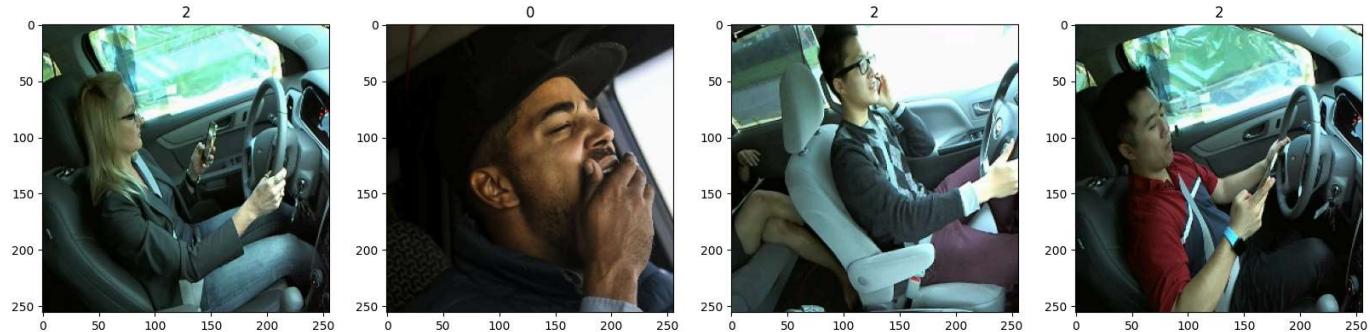
```
In [ ]: scaled_it = data.as_numpy_iterator()
```

```
In [ ]: batch = scaled_it.next()
```

```
In [ ]: batch[0].max()
```

```
Out[ ]: 1.0
```

```
In [ ]: fig, ax = plt.subplots(ncols=4, figsize=(20,20))
for idx, img in enumerate(batch[0][:4]):
    ax[idx].imshow(img)
    ax[idx].title.set_text(batch[1][idx])
```



4. Splitting Data

```
In [ ]: len(data)
```

```
Out[ ]: 39
```

```
In [ ]: train_size = int(len(data)*.72)
val_size = int(len(data)*.2)
test_size = int(len(data)*.1)+1
```

```
In [ ]: train_size
```

```
Out[ ]: 28
```

```
In [ ]: val_size
```

```
Out[ ]: 7
```

```
In [ ]: test_size
```

```
Out[ ]: 4
```

```
In [ ]: train = data.take(train_size)
val = data.skip(train_size).take(val_size)
test = data.skip(train_size+val_size).take(test_size)
```

```
In [ ]: print('Test: ',len(test))
print('Validation: ',len(val))
print('Train: ',len(train))
```

```
Test: 4
```

```
Validation: 7
```

```
Train: 28
```

5. Build Deep Learning Model

```
In [ ]: from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Dense, Flatten, BatchNormalization, Dropout
from tensorflow.keras.initializers import HeNormal
```

```
In [ ]: model = Sequential()
```

WARNING:tensorflow:From p:\MachineLearning\DriverBehaviorAnalysisAndViolationDetection\venv\Lib\site-packages\keras\src\backend.py:873: The name tf.get_default_graph is deprecated. Please use tf.compat.v1.get_default_graph instead.

```
In [ ]: model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(256, 256, 3), kernel_initializer=HeNormal()))
model.add(BatchNormalization())
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer=HeNormal()))
model.add(BatchNormalization())
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(64, (3, 3), activation='relu', kernel_initializer=HeNormal()))
model.add(BatchNormalization())
model.add(Conv2D(64, (3, 3), activation='relu', kernel_initializer=HeNormal()))
model.add(BatchNormalization())
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(128, (3, 3), activation='relu', kernel_initializer=HeNormal()))
model.add(BatchNormalization())
model.add(Conv2D(128, (3, 3), activation='relu', kernel_initializer=HeNormal()))
model.add(BatchNormalization())
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(128, activation='relu', kernel_initializer=HeNormal()))
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(Dense(3, activation='softmax'))
```

WARNING:tensorflow:From p:\MachineLearning\DriverBehaviorAnalysisAndViolationDetection\venv\Lib\site-packages\keras\src\layers\normalization\batch_normalization.py:979: The name tf.nn.fused_batch_norm is deprecated. Please use tf.compat.v1.nn.fused_batch_norm instead.

```
In [ ]: model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

WARNING:tensorflow:From p:\MachineLearning\DriverBehaviorAnalysisAndViolationDetection\venv\Lib\site-packages\keras\src\optimizers__init__.py:309: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

```
In [ ]: model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 254, 254, 32)	896
batch_normalization (Batch Normalization)	(None, 254, 254, 32)	128
conv2d_1 (Conv2D)	(None, 252, 252, 32)	9248
batch_normalization_1 (Batch Normalization)	(None, 252, 252, 32)	128
max_pooling2d (MaxPooling2D)	(None, 126, 126, 32)	0
dropout (Dropout)	(None, 126, 126, 32)	0
conv2d_2 (Conv2D)	(None, 124, 124, 64)	18496
batch_normalization_2 (Batch Normalization)	(None, 124, 124, 64)	256
conv2d_3 (Conv2D)	(None, 122, 122, 64)	36928
batch_normalization_3 (Batch Normalization)	(None, 122, 122, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 61, 61, 64)	0
dropout_1 (Dropout)	(None, 61, 61, 64)	0
conv2d_4 (Conv2D)	(None, 59, 59, 128)	73856
batch_normalization_4 (Batch Normalization)	(None, 59, 59, 128)	512
conv2d_5 (Conv2D)	(None, 57, 57, 128)	147584
batch_normalization_5 (Batch Normalization)	(None, 57, 57, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 28, 28, 128)	0
dropout_2 (Dropout)	(None, 28, 28, 128)	0
flatten (Flatten)	(None, 100352)	0
dense (Dense)	(None, 128)	12845184
batch_normalization_6 (Batch Normalization)	(None, 128)	512
dropout_3 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 3)	387

Total params: 13134883 (50.11 MB)
Trainable params: 13133731 (50.10 MB)

Non-trainable params: 1152 (4.50 KB)

6. Train

```
In [ ]: logdir='logs'
```

```
In [ ]: tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=logdir)
```

```
In [ ]: hist = model.fit(train, epochs=20, validation_data=val, callbacks=[tensorboard_callback])
```

Epoch 1/20

WARNING:tensorflow:From p:\MachineLearning\DriverBehaviorAnalysisAndViolationDetection\venv\Lib\site-packages\keras\src\utils\tf_utils.py:492: The name tf.ragged.RaggedTensorValue is deprecated. Please use tf.compat.v1.ragged.RaggedTensorValue instead.

WARNING:tensorflow:From p:\MachineLearning\DriverBehaviorAnalysisAndViolationDetection\venv\Lib\site-packages\keras\src\engine\base_layer_utils.py:384: The name tf.executing_eagerly_outside_functions is deprecated. Please use tf.compat.v1.executing_eagerly_outside_functions instead.

28/28 [=====] - 573s 20s/step - loss: 1.0431 - accuracy: 0.6228 - val_loss: 0.8805 - val_accuracy: 0.6250

Epoch 2/20

28/28 [=====] - 536s 19s/step - loss: 0.6536 - accuracy: 0.7232 - val_loss: 0.8401 - val_accuracy: 0.6786

Epoch 3/20

28/28 [=====] - 515s 18s/step - loss: 0.6204 - accuracy: 0.7254 - val_loss: 0.9023 - val_accuracy: 0.5625

Epoch 4/20

28/28 [=====] - 516s 18s/step - loss: 0.4798 - accuracy: 0.8013 - val_loss: 0.8090 - val_accuracy: 0.6339

Epoch 5/20

28/28 [=====] - 510s 18s/step - loss: 0.3719 - accuracy: 0.8371 - val_loss: 0.6904 - val_accuracy: 0.7232

Epoch 6/20

28/28 [=====] - 527s 19s/step - loss: 0.3579 - accuracy: 0.8616 - val_loss: 0.9053 - val_accuracy: 0.5982

Epoch 7/20

28/28 [=====] - 480s 17s/step - loss: 0.2431 - accuracy: 0.9174 - val_loss: 0.7004 - val_accuracy: 0.7054

Epoch 8/20

28/28 [=====] - 468s 17s/step - loss: 0.2265 - accuracy: 0.9107 - val_loss: 0.7885 - val_accuracy: 0.6250

Epoch 9/20

28/28 [=====] - 587s 21s/step - loss: 0.2115 - accuracy: 0.9174 - val_loss: 0.5063 - val_accuracy: 0.7679

Epoch 10/20

28/28 [=====] - 521s 18s/step - loss: 0.1707 - accuracy: 0.9375 - val_loss: 0.4563 - val_accuracy: 0.8125

Epoch 11/20

28/28 [=====] - 473s 17s/step - loss: 0.1505 - accuracy: 0.9531 - val_loss: 0.6761 - val_accuracy: 0.7321

Epoch 12/20

28/28 [=====] - 473s 17s/step - loss: 0.1719 - accuracy: 0.9442 - val_loss: 0.2698 - val_accuracy: 0.9107

Epoch 13/20

28/28 [=====] - 1405s 51s/step - loss: 0.1460 - accuracy: 0.9464 - val_loss: 0.3128 - val_accuracy: 0.8839

Epoch 14/20

28/28 [=====] - 480s 17s/step - loss: 0.1246 - accuracy: 0.9598 - val_loss: 0.5300 - val_accuracy: 0.8125

Epoch 15/20

28/28 [=====] - 457s 16s/step - loss: 0.1019 - accuracy: 0.9621 - val_loss: 0.4185 - val_accuracy: 0.8661

Epoch 16/20

28/28 [=====] - 463s 17s/step - loss: 0.0952 - accuracy: 0.9732 - val_loss: 0.3065 - val_accuracy: 0.9018

Epoch 17/20

28/28 [=====] - 3758s 139s/step - loss: 0.0729 - accuracy: 0.9821 - val_loss: 0.2923 - val_accuracy: 0.9107

Epoch 18/20

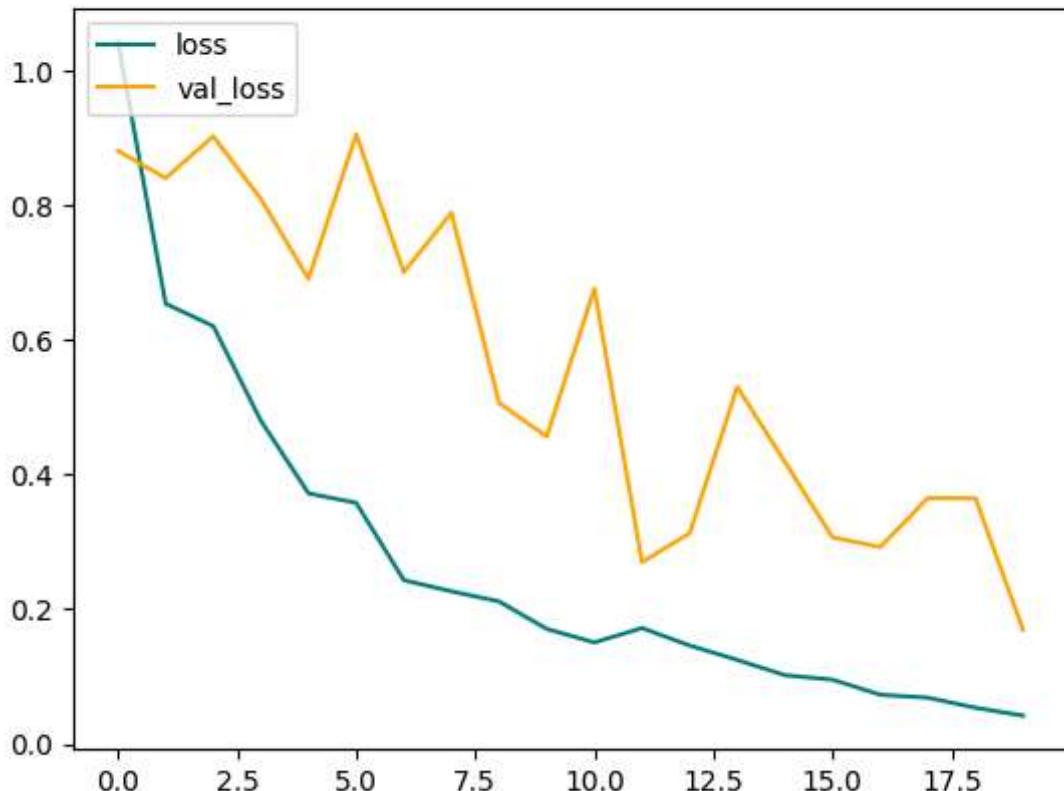
28/28 [=====] - 523s 18s/step - loss: 0.0685 - accuracy: 0.9754 - val_loss: 0.3649 - val_accuracy: 0.8661

Epoch 19/20

```
28/28 [=====] - 504s 18s/step - loss: 0.0535 - accuracy: 0.9821 - val_loss: 0.3647 - val_accuracy: 0.8750
Epoch 20/20
28/28 [=====] - 506s 18s/step - loss: 0.0419 - accuracy: 0.9911 - val_loss: 0.1693 - val_accuracy: 0.9286
```

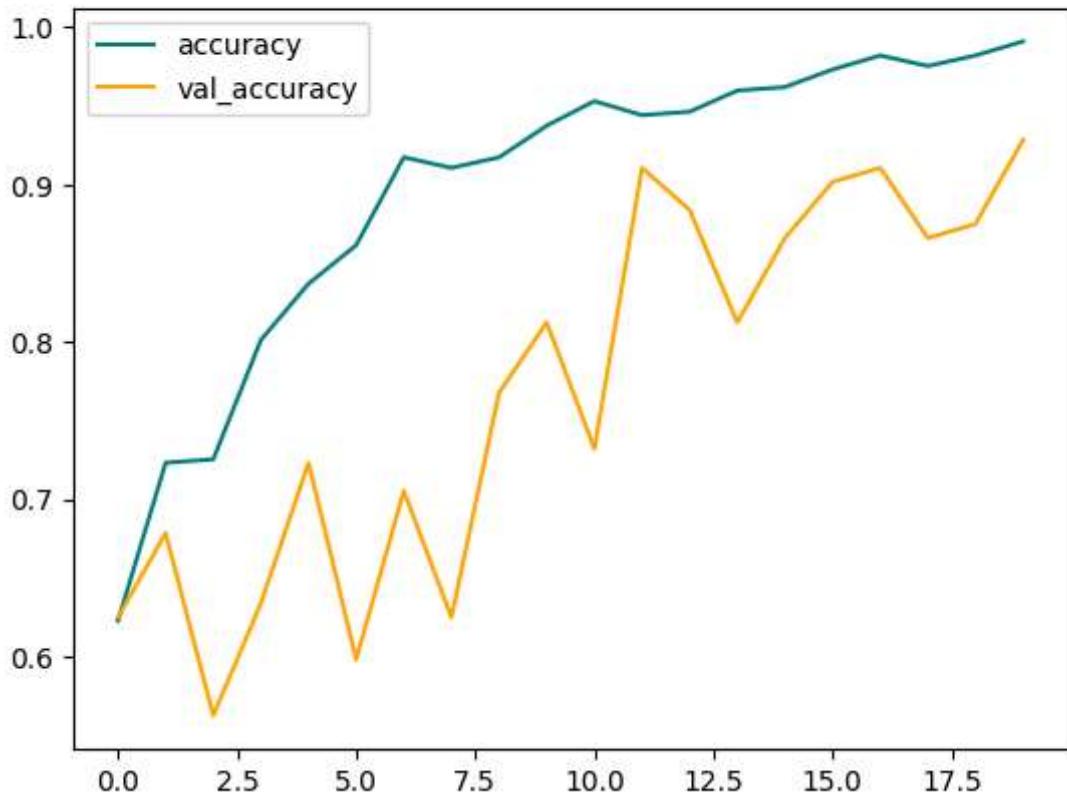
```
In [ ]: fig = plt.figure()
plt.plot(hist.history['loss'], color='teal', label='loss')
plt.plot(hist.history['val_loss'], color='orange', label='val_loss')
fig.suptitle('Loss', fontsize=20)
plt.legend(loc="upper left")
plt.show()
```

Loss



```
In [ ]: fig = plt.figure()
plt.plot(hist.history['accuracy'], color='teal', label='accuracy')
plt.plot(hist.history['val_accuracy'], color='orange', label='val_accuracy')
fig.suptitle('Accuracy', fontsize=20)
plt.legend(loc="upper left")
plt.show()
```

Accuracy



Saving the model

```
In [ ]: from tensorflow.keras.models import load_model
```

```
In [ ]: model.save(os.path.join('models', 'driverViolation.h5'))
```

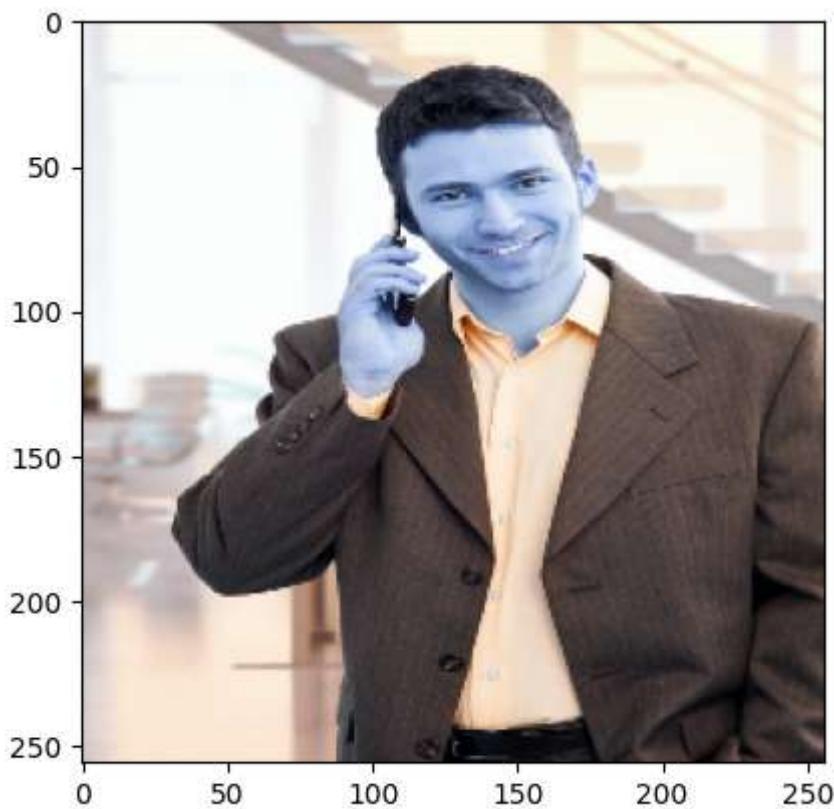
```
p:\MachineLearning\DriverBehaviorAnalysisAndViolationDetection\venv\Lib\site-packages\keras\src\nonconfigurable\training.py:3103: UserWarning: You are saving your model as an HDF5 file via `model.save()`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')`.  
... saving_api.save_model()
```

Testing

```
In [ ]: img = cv2.imread('ca33.jpg')  
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))  
plt.show()
```



```
In [ ]: resize = tf.image.resize(img, (256,256))
plt.imshow(resize.numpy().astype(int))
plt.show()
```



```
In [ ]: np.expand_dims(resize,0).shape
```

```
Out[ ]: (1, 256, 256, 3)
```

```
In [ ]: yhat = model.predict(np.expand_dims(resize/255, 0))
```

```
1/1 [=====] - 8s 8s/step
```

```
In [ ]: yhat
```

```
Out[ ]: array([[0.05515841, 0.231414 , 0.7134276 ]], dtype=float32)
```

Loading a model

```
In [ ]: img = cv2.imread('sad.jpg')
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
resize = tf.image.resize(img, (256,256))
plt.imshow(resize.numpy().astype(int))
plt.show()
```

```
In [ ]: np.expand_dims(resize,0).shape
```

```
In [ ]: new_model = load_model(os.path.join('models','imageclassifier.h5'))
```

```
In [ ]: new_model.predict(np.expand_dims(resize/255, 0))
```