

# Distracted Driving Detection

## Load the Data

```
In [1]: #dictionary for distraction category to numerical value
catLabels = {
    'c0': 'safe driving',
    'c1': 'texting - right',
    'c2': 'talking on the phone - right',
    'c3': 'texting - left',
    'c4': 'talking on the phone - left',
    'c5': 'operating the radio',
    'c6': 'drinking',
    'c7': 'reaching behind',
    'c8': 'hair and makeup',
    'c9': 'talking to passenger'
}

def getClass(value):
    index = 'c' + str(value)
    return catLabels[index]
```

```

In [2]: from sklearn.datasets import load_files
        from keras.utils import np_utils
        import numpy as np
        from glob import glob
        import os
        from sklearn.model_selection import train_test_split

        def loadImages(path):
            data = load_files(path)
            files = data['filenames']
            targets = data['target']
            target_names = data['target_names']
            return files, targets, target_names

        path = "images/train"
        files, targets, target_names = loadImages(path)
        predict_files = np.array(glob("images/test/*"))[1:10]
        print('Number of Categories: ', len(target_names))
        print('Categories: ', target_names)
        print('Number of images by category: ')
        for c in target_names:
            print(c + ':' + str(len( os.listdir(path+'/'+c))))
        # train_data = np.vstack((files, targets)).T
        # print(train_data.shape)

        #Split the original training sets into training & validation sets
        train_files, test_files, train_targets, test_targets = train_test_split(files,
            targets, test_size=0.20, random_state=40)

        print(train_files.shape, test_files.shape, train_targets.shape, test_targets.s
            hape)
        print(len(test_files))

```

/usr/local/lib/python2.7/dist-packages/h5py/\_\_init\_\_.py:36: FutureWarning: Conversion of the second argument of issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 == np.dtype(float).type`.

from .\_conv import register\_converters as \_register\_converters  
Using TensorFlow backend.

```

('Number of Categories: ', 10)
('Categories: ', ['c0', 'c1', 'c2', 'c3', 'c4', 'c5', 'c6', 'c7', 'c8', 'c
9'])

```

Number of images by category:

```

c0:1900
c1:1900
c2:1900
c3:1900
c4:1900
c5:1900
c6:1900
c7:1900
c8:1900
c9:1900
((15200,), (3800,), (15200,), (3800,))
3800

```

# Data Analysis

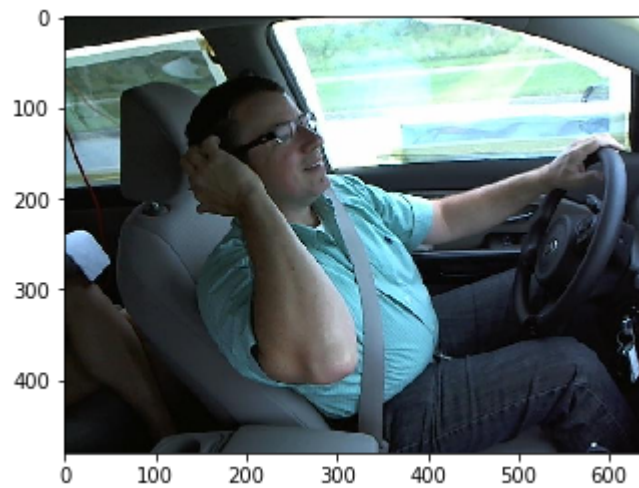
```
In [3]: import cv2
import matplotlib.pyplot as plt
%matplotlib inline

def displayImage(sample_image):
    gray = cv2.cvtColor(sample_image, cv2.COLOR_BGR2GRAY)

    # convert BGR image to RGB for plotting
    cv_rgb = cv2.cvtColor(sample_image, cv2.COLOR_BGR2RGB)
    plt.imshow(cv_rgb)
    plt.show()

for i in range(1,5):
    sample_image = cv2.imread(train_files[i])
    print(train_targets[i])
    print getClass(train_targets[i])
    displayImage(sample_image)
```

8  
hair and makeup



9  
talking to passenger



0  
safe driving



0  
safe driving



```
In [4]: #(nb_samples,rows,columns,channels)
#nb_samples - total number of images
# Resize image to 224x224
# Convert image to an array -> resized to a 4D tensor used by Keras CNN
# Tensor will be (1,224,224,3)

#Adopted from the Deep Learning Project
from keras.preprocessing import image
from tqdm import tqdm

def path_to_tensor(img_path):
    # Loads RGB image as PIL.Image.Image type
    img = image.load_img(img_path, target_size=(224, 224))
    # convert PIL.Image.Image type to 3D tensor with shape (224, 224, 3)
    x = image.img_to_array(img)
    # convert 3D tensor to 4D tensor with shape (1, 224, 224, 3) and return 4D tensor
    return np.expand_dims(x, axis=0)

def paths_to_tensor(img_paths):
    print (img_paths)
    list_of_tensors = [path_to_tensor(img_path) for img_path in tqdm(img_paths)]
    return np.vstack(list_of_tensors)
```

## Pre-Process the Data

```

In [5]: #Rescale the images

from PIL import ImageFile
ImageFile.LOAD_TRUNCATED_IMAGES = True

train_tensors = paths_to_tensor(train_files).astype('float32')/255
test_tensors = paths_to_tensor(test_files).astype('float32')/255
#predict_tensors = paths_to_tensor(predict_files).astype('float32')/255

0%|          | 9/15200 [00:00<02:55, 86.77it/s]

['images/train/c3/img_24663.jpg' 'images/train/c8/img_98810.jpg'
 'images/train/c9/img_67390.jpg' ... 'images/train/c7/img_31727.jpg'
 'images/train/c7/img_82756.jpg' 'images/train/c5/img_21995.jpg']

100%|██████████| 15200/15200 [01:15<00:00, 201.56it/s]
0%|          | 9/3800 [00:00<00:42, 88.45it/s]

['images/train/c5/img_68264.jpg' 'images/train/c6/img_69335.jpg'
 'images/train/c2/img_12280.jpg' ... 'images/train/c8/img_6916.jpg'
 'images/train/c6/img_21610.jpg' 'images/train/c5/img_46343.jpg']

100%|██████████| 3800/3800 [00:27<00:00, 135.99it/s]

```

## Baseline Model Architecture

```
In [6]: from keras.layers import Conv2D, MaxPooling2D, GlobalAveragePooling2D
from keras.layers import Dropout, Flatten, Dense
from keras.models import Sequential
from keras.utils import plot_model

model = Sequential()

### TODO: Define your architecture.
model.add(Conv2D(filters=10, kernel_size=(4,4), input_shape=(224,224,3)))
model.add(MaxPooling2D(pool_size=(4, 4), strides=None, padding='valid', data_format=None))
model.add(Conv2D(filters=10, kernel_size=(4,4), input_shape=(224,224,3)))
model.add(MaxPooling2D(pool_size=(4, 4), strides=None, padding='valid', data_format=None))
model.add(GlobalAveragePooling2D())
model.add(Dense(units=10, activation='softmax'))
model.summary()

model.compile(optimizer='rmsprop', loss='categorical_crossentropy', metrics=['accuracy'])

# from IPython.display import SVG
# from keras.utils.vis_utils import model_to_dot
# plot_model(model, to_file='model.png')
# SVG(model_to_dot(model).create(prog='dot', format='svg'))
```

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 221, 221, 10)	490
max_pooling2d_1 (MaxPooling2D)	(None, 55, 55, 10)	0
conv2d_2 (Conv2D)	(None, 52, 52, 10)	1610
max_pooling2d_2 (MaxPooling2D)	(None, 13, 13, 10)	0
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 10)	0
dense_1 (Dense)	(None, 10)	110
Total params: 2,210		
Trainable params: 2,210		
Non-trainable params: 0		

## Train the Model



```
In [17]: from keras.callbacks import ModelCheckpoint
from keras.utils import np_utils

print("Train Targets", train_targets)
print ("Test Targets", test_targets)
train_targets_onehot = np_utils.to_categorical(np.array(train_targets),10)
test_targets_onehot = np_utils.to_categorical(np.array(test_targets),10)
print ("Train Targets One-hot encoded", train_targets_onehot)
print ("Test Targets One-hot encoded", test_targets_onehot)

print(train_targets_onehot.shape)
print(test_targets_onehot.shape)

checkpointer = ModelCheckpoint(filepath='/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weights.best.from_scratch.hdf5',
                               verbose=1, save_best_only=True)

def train_model(_epochs):
    epochs = _epochs

    history = model.fit(train_tensors, train_targets_onehot, validation_split
                        =.20,
                        epochs=epochs, batch_size=32, callbacks=[checkpointer], verbose=1)
    return history

history = train_model(100)
```

```

('Train Targets', array([3, 8, 9, ..., 7, 7, 5]))
('Test Targets', array([5, 6, 2, ..., 8, 6, 5]))
('Train Targets One-hot encoded', array([[0., 0., 0., ..., 0., 0., 0.],
      [0., 0., 0., ..., 0., 1., 0.],
      [0., 0., 0., ..., 0., 0., 1.],
      ...,
      [0., 0., 0., ..., 1., 0., 0.],
      [0., 0., 0., ..., 1., 0., 0.],
      [0., 0., 0., ..., 0., 0., 0.])))
('Test Targets One-hot encoded', array([[0., 0., 0., ..., 0., 0., 0.],
      [0., 0., 0., ..., 0., 0., 0.],
      [0., 0., 1., ..., 0., 0., 0.],
      ...,
      [0., 0., 0., ..., 0., 1., 0.],
      [0., 0., 0., ..., 0., 0., 0.],
      [0., 0., 0., ..., 0., 0., 0.])))
(15200, 10)
(3800, 10)
Train on 12160 samples, validate on 3040 samples
Epoch 1/100
12160/12160 [=====] - 117s 10ms/step - loss: 1.6862 - acc: 0.3984 - val_loss: 1.7190 - val_acc: 0.3668

Epoch 00001: val_loss improved from inf to 1.71896, saving model to /mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weights.best.from_scratch.hdf5
Epoch 2/100
12160/12160 [=====] - 111s 9ms/step - loss: 1.6636 - acc: 0.4170 - val_loss: 1.6809 - val_acc: 0.3990

Epoch 00002: val_loss improved from 1.71896 to 1.68088, saving model to /mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weights.best.from_scratch.hdf5
Epoch 3/100
12160/12160 [=====] - 108s 9ms/step - loss: 1.6424 - acc: 0.4212 - val_loss: 1.6383 - val_acc: 0.4355

Epoch 00003: val_loss improved from 1.68088 to 1.63825, saving model to /mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weights.best.from_scratch.hdf5
Epoch 4/100
12160/12160 [=====] - 108s 9ms/step - loss: 1.6270 - acc: 0.4257 - val_loss: 1.6110 - val_acc: 0.4286

Epoch 00004: val_loss improved from 1.63825 to 1.61105, saving model to /mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weights.best.from_scratch.hdf5
Epoch 5/100
12160/12160 [=====] - 109s 9ms/step - loss: 1.6052 - acc: 0.4283 - val_loss: 1.6644 - val_acc: 0.4141

Epoch 00005: val_loss did not improve
Epoch 6/100
12160/12160 [=====] - 114s 9ms/step - loss: 1.5861 - acc: 0.4436 - val_loss: 1.5874 - val_acc: 0.4220

Epoch 00006: val_loss improved from 1.61105 to 1.58740, saving model to

```

```
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/  
weights.best.from_scratch.hdf5
```

```
Epoch 7/100
```

```
12160/12160 [=====] - 114s 9ms/step - loss: 1.5  
658 - acc: 0.4498 - val_loss: 1.5681 - val_acc: 0.4526
```

```
Epoch 00007: val_loss improved from 1.58740 to 1.56805, saving model to  
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/  
weights.best.from_scratch.hdf5
```

```
Epoch 8/100
```

```
12160/12160 [=====] - 110s 9ms/step - loss: 1.5  
512 - acc: 0.4553 - val_loss: 1.5469 - val_acc: 0.4539
```

```
Epoch 00008: val_loss improved from 1.56805 to 1.54691, saving model to  
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/  
weights.best.from_scratch.hdf5
```

```
Epoch 9/100
```

```
12160/12160 [=====] - 108s 9ms/step - loss: 1.5  
361 - acc: 0.4618 - val_loss: 1.5616 - val_acc: 0.4368
```

```
Epoch 00009: val_loss did not improve
```

```
Epoch 10/100
```

```
12160/12160 [=====] - 107s 9ms/step - loss: 1.5  
200 - acc: 0.4688 - val_loss: 1.5187 - val_acc: 0.4701
```

```
Epoch 00010: val_loss improved from 1.54691 to 1.51873, saving model to  
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/  
weights.best.from_scratch.hdf5
```

```
Epoch 11/100
```

```
12160/12160 [=====] - 108s 9ms/step - loss: 1.5  
028 - acc: 0.4757 - val_loss: 1.5215 - val_acc: 0.4668
```

```
Epoch 00011: val_loss did not improve
```

```
Epoch 12/100
```

```
12160/12160 [=====] - 37715s 3s/step - loss: 1.  
4888 - acc: 0.4840 - val_loss: 1.4997 - val_acc: 0.4875
```

```
Epoch 00012: val_loss improved from 1.51873 to 1.49970, saving model to  
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/  
weights.best.from_scratch.hdf5
```

```
Epoch 13/100
```

```
12160/12160 [=====] - 127s 10ms/step - loss: 1.  
4763 - acc: 0.4830 - val_loss: 1.4891 - val_acc: 0.4704
```

```
Epoch 00013: val_loss improved from 1.49970 to 1.48907, saving model to  
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/  
weights.best.from_scratch.hdf5
```

```
Epoch 14/100
```

```
12160/12160 [=====] - 135s 11ms/step - loss: 1.  
4592 - acc: 0.4918 - val_loss: 1.4652 - val_acc: 0.4799
```

```
Epoch 00014: val_loss improved from 1.48907 to 1.46520, saving model to  
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/  
weights.best.from_scratch.hdf5
```

```
Epoch 15/100
```

```
12160/12160 [=====] - 132s 11ms/step - loss: 1.  
4460 - acc: 0.4970 - val_loss: 1.4514 - val_acc: 0.5030
```

Epoch 00015: val\_loss improved from 1.46520 to 1.45144, saving model to /mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/weights.best.from\_scratch.hdf5

Epoch 16/100

12160/12160 [=====] - 110s 9ms/step - loss: 1.4301 - acc: 0.5009 - val\_loss: 1.4430 - val\_acc: 0.5000

Epoch 00016: val\_loss improved from 1.45144 to 1.44296, saving model to /mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/weights.best.from\_scratch.hdf5

Epoch 17/100

12160/12160 [=====] - 115s 9ms/step - loss: 1.4187 - acc: 0.5086 - val\_loss: 1.4287 - val\_acc: 0.4961

Epoch 00017: val\_loss improved from 1.44296 to 1.42869, saving model to /mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/weights.best.from\_scratch.hdf5

Epoch 18/100

12160/12160 [=====] - 110s 9ms/step - loss: 1.4063 - acc: 0.5115 - val\_loss: 1.4062 - val\_acc: 0.5118

Epoch 00018: val\_loss improved from 1.42869 to 1.40616, saving model to /mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/weights.best.from\_scratch.hdf5

Epoch 19/100

12160/12160 [=====] - 110s 9ms/step - loss: 1.3945 - acc: 0.5190 - val\_loss: 1.4118 - val\_acc: 0.5089

Epoch 00019: val\_loss did not improve

Epoch 20/100

12160/12160 [=====] - 110s 9ms/step - loss: 1.3822 - acc: 0.5234 - val\_loss: 1.3822 - val\_acc: 0.5286

Epoch 00020: val\_loss improved from 1.40616 to 1.38217, saving model to /mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/weights.best.from\_scratch.hdf5

Epoch 21/100

12160/12160 [=====] - 114s 9ms/step - loss: 1.3672 - acc: 0.5269 - val\_loss: 1.4010 - val\_acc: 0.5036

Epoch 00021: val\_loss did not improve

Epoch 22/100

12160/12160 [=====] - 113s 9ms/step - loss: 1.3606 - acc: 0.5291 - val\_loss: 1.3826 - val\_acc: 0.5109

Epoch 00022: val\_loss did not improve

Epoch 23/100

12160/12160 [=====] - 109s 9ms/step - loss: 1.3490 - acc: 0.5350 - val\_loss: 1.4321 - val\_acc: 0.4882

Epoch 00023: val\_loss did not improve

Epoch 24/100

12160/12160 [=====] - 107s 9ms/step - loss: 1.3383 - acc: 0.5392 - val\_loss: 1.3549 - val\_acc: 0.5283

Epoch 00024: val\_loss improved from 1.38217 to 1.35495, saving model to

/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/  
weights.best.from\_scratch.hdf5

Epoch 25/100

12160/12160 [=====] - 110s 9ms/step - loss: 1.3  
274 - acc: 0.5451 - val\_loss: 1.3689 - val\_acc: 0.5276

Epoch 00025: val\_loss did not improve

Epoch 26/100

12160/12160 [=====] - 8494s 699ms/step - loss:  
1.3136 - acc: 0.5478 - val\_loss: 1.3126 - val\_acc: 0.5546

Epoch 00026: val\_loss improved from 1.35495 to 1.31257, saving model to  
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/  
weights.best.from\_scratch.hdf5

Epoch 27/100

12160/12160 [=====] - 124s 10ms/step - loss: 1.  
3073 - acc: 0.5556 - val\_loss: 1.3252 - val\_acc: 0.5395

Epoch 00027: val\_loss did not improve

Epoch 28/100

12160/12160 [=====] - 125s 10ms/step - loss: 1.  
2968 - acc: 0.5571 - val\_loss: 1.3250 - val\_acc: 0.5342

Epoch 00028: val\_loss did not improve

Epoch 29/100

12160/12160 [=====] - 125s 10ms/step - loss: 1.  
2863 - acc: 0.5593 - val\_loss: 1.3205 - val\_acc: 0.5382

Epoch 00029: val\_loss did not improve

Epoch 30/100

12160/12160 [=====] - 122s 10ms/step - loss: 1.  
2761 - acc: 0.5666 - val\_loss: 1.3025 - val\_acc: 0.5424

Epoch 00030: val\_loss improved from 1.31257 to 1.30249, saving model to  
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/  
weights.best.from\_scratch.hdf5

Epoch 31/100

12160/12160 [=====] - 117s 10ms/step - loss: 1.  
2660 - acc: 0.5678 - val\_loss: 1.2812 - val\_acc: 0.5572

Epoch 00031: val\_loss improved from 1.30249 to 1.28121, saving model to  
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/  
weights.best.from\_scratch.hdf5

Epoch 32/100

12160/12160 [=====] - 114s 9ms/step - loss: 1.2  
529 - acc: 0.5765 - val\_loss: 1.2786 - val\_acc: 0.5651

Epoch 00032: val\_loss improved from 1.28121 to 1.27861, saving model to  
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/  
weights.best.from\_scratch.hdf5

Epoch 33/100

12160/12160 [=====] - 111s 9ms/step - loss: 1.2  
488 - acc: 0.5800 - val\_loss: 1.2434 - val\_acc: 0.5852

Epoch 00033: val\_loss improved from 1.27861 to 1.24336, saving model to  
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/  
weights.best.from\_scratch.hdf5

Epoch 34/100  
12160/12160 [=====] - 111s 9ms/step - loss: 1.2  
407 - acc: 0.5756 - val\_loss: 1.2574 - val\_acc: 0.5648

Epoch 00034: val\_loss did not improve  
Epoch 35/100  
12160/12160 [=====] - 113s 9ms/step - loss: 1.2  
299 - acc: 0.5814 - val\_loss: 1.2239 - val\_acc: 0.5908

Epoch 00035: val\_loss improved from 1.24336 to 1.22385, saving model to  
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/  
weights.best.from\_scratch.hdf5  
Epoch 36/100  
12160/12160 [=====] - 111s 9ms/step - loss: 1.2  
216 - acc: 0.5829 - val\_loss: 1.2907 - val\_acc: 0.5382

Epoch 00036: val\_loss did not improve  
Epoch 37/100  
12160/12160 [=====] - 112s 9ms/step - loss: 1.2  
128 - acc: 0.5881 - val\_loss: 1.2752 - val\_acc: 0.5388

Epoch 00037: val\_loss did not improve  
Epoch 38/100  
12160/12160 [=====] - 111s 9ms/step - loss: 1.2  
014 - acc: 0.5959 - val\_loss: 1.2417 - val\_acc: 0.5720

Epoch 00038: val\_loss did not improve  
Epoch 39/100  
12160/12160 [=====] - 111s 9ms/step - loss: 1.1  
951 - acc: 0.5959 - val\_loss: 1.2190 - val\_acc: 0.5895

Epoch 00039: val\_loss improved from 1.22385 to 1.21901, saving model to  
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/  
weights.best.from\_scratch.hdf5  
Epoch 40/100  
12160/12160 [=====] - 111s 9ms/step - loss: 1.1  
895 - acc: 0.5984 - val\_loss: 1.1848 - val\_acc: 0.6118

Epoch 00040: val\_loss improved from 1.21901 to 1.18480, saving model to  
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/  
weights.best.from\_scratch.hdf5  
Epoch 41/100  
12160/12160 [=====] - 112s 9ms/step - loss: 1.1  
794 - acc: 0.5979 - val\_loss: 1.1841 - val\_acc: 0.6003

Epoch 00041: val\_loss improved from 1.18480 to 1.18412, saving model to  
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/  
weights.best.from\_scratch.hdf5  
Epoch 42/100  
12160/12160 [=====] - 111s 9ms/step - loss: 1.1  
713 - acc: 0.6023 - val\_loss: 1.2103 - val\_acc: 0.5740

Epoch 00042: val\_loss did not improve  
Epoch 43/100  
12160/12160 [=====] - 111s 9ms/step - loss: 1.1  
679 - acc: 0.6024 - val\_loss: 1.1797 - val\_acc: 0.6023

Epoch 00043: val\_loss improved from 1.18412 to 1.17970, saving model to /mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/weights.best.from\_scratch.hdf5

Epoch 44/100

12160/12160 [=====] - 111s 9ms/step - loss: 1.1582 - acc: 0.6086 - val\_loss: 1.2136 - val\_acc: 0.5783

Epoch 00044: val\_loss did not improve

Epoch 45/100

12160/12160 [=====] - 111s 9ms/step - loss: 1.1528 - acc: 0.6078 - val\_loss: 1.1818 - val\_acc: 0.5938

Epoch 00045: val\_loss did not improve

Epoch 46/100

12160/12160 [=====] - 111s 9ms/step - loss: 1.1451 - acc: 0.6128 - val\_loss: 1.1527 - val\_acc: 0.6128

Epoch 00046: val\_loss improved from 1.17970 to 1.15271, saving model to /mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/weights.best.from\_scratch.hdf5

Epoch 47/100

12160/12160 [=====] - 111s 9ms/step - loss: 1.1372 - acc: 0.6150 - val\_loss: 1.1609 - val\_acc: 0.5970

Epoch 00047: val\_loss did not improve

Epoch 48/100

12160/12160 [=====] - 111s 9ms/step - loss: 1.1331 - acc: 0.6197 - val\_loss: 1.1420 - val\_acc: 0.6178

Epoch 00048: val\_loss improved from 1.15271 to 1.14201, saving model to /mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/weights.best.from\_scratch.hdf5

Epoch 49/100

12160/12160 [=====] - 111s 9ms/step - loss: 1.1215 - acc: 0.6215 - val\_loss: 1.2238 - val\_acc: 0.5632

Epoch 00049: val\_loss did not improve

Epoch 50/100

12160/12160 [=====] - 111s 9ms/step - loss: 1.1170 - acc: 0.6202 - val\_loss: 1.1259 - val\_acc: 0.6312

Epoch 00050: val\_loss improved from 1.14201 to 1.12588, saving model to /mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/weights.best.from\_scratch.hdf5

Epoch 51/100

12160/12160 [=====] - 111s 9ms/step - loss: 1.1093 - acc: 0.6268 - val\_loss: 1.1415 - val\_acc: 0.6079

Epoch 00051: val\_loss did not improve

Epoch 52/100

12160/12160 [=====] - 111s 9ms/step - loss: 1.1067 - acc: 0.6272 - val\_loss: 1.2132 - val\_acc: 0.5609

Epoch 00052: val\_loss did not improve

Epoch 53/100

12160/12160 [=====] - 111s 9ms/step - loss: 1.0961 - acc: 0.6285 - val\_loss: 1.1346 - val\_acc: 0.6076

Epoch 00053: val\_loss did not improve  
Epoch 54/100  
12160/12160 [=====] - 111s 9ms/step - loss: 1.0  
911 - acc: 0.6299 - val\_loss: 1.1268 - val\_acc: 0.6072

Epoch 00054: val\_loss did not improve  
Epoch 55/100  
12160/12160 [=====] - 110s 9ms/step - loss: 1.0  
847 - acc: 0.6378 - val\_loss: 1.1051 - val\_acc: 0.6188

Epoch 00055: val\_loss improved from 1.12588 to 1.10514, saving model to  
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/  
weights.best.from\_scratch.hdf5  
Epoch 56/100  
12160/12160 [=====] - 111s 9ms/step - loss: 1.0  
831 - acc: 0.6347 - val\_loss: 1.0908 - val\_acc: 0.6280

Epoch 00056: val\_loss improved from 1.10514 to 1.09080, saving model to  
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/  
weights.best.from\_scratch.hdf5  
Epoch 57/100  
12160/12160 [=====] - 111s 9ms/step - loss: 1.0  
725 - acc: 0.6363 - val\_loss: 1.0851 - val\_acc: 0.6385

Epoch 00057: val\_loss improved from 1.09080 to 1.08509, saving model to  
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/  
weights.best.from\_scratch.hdf5  
Epoch 58/100  
12160/12160 [=====] - 112s 9ms/step - loss: 1.0  
650 - acc: 0.6377 - val\_loss: 1.1117 - val\_acc: 0.6128

Epoch 00058: val\_loss did not improve  
Epoch 59/100  
12160/12160 [=====] - 111s 9ms/step - loss: 1.0  
644 - acc: 0.6447 - val\_loss: 1.0818 - val\_acc: 0.6296

Epoch 00059: val\_loss improved from 1.08509 to 1.08177, saving model to  
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/  
weights.best.from\_scratch.hdf5  
Epoch 60/100  
12160/12160 [=====] - 110s 9ms/step - loss: 1.0  
582 - acc: 0.6397 - val\_loss: 1.0965 - val\_acc: 0.6289

Epoch 00060: val\_loss did not improve  
Epoch 61/100  
12160/12160 [=====] - 110s 9ms/step - loss: 1.0  
523 - acc: 0.6427 - val\_loss: 1.0549 - val\_acc: 0.6480

Epoch 00061: val\_loss improved from 1.08177 to 1.05491, saving model to  
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/  
weights.best.from\_scratch.hdf5  
Epoch 62/100  
12160/12160 [=====] - 110s 9ms/step - loss: 1.0  
455 - acc: 0.6464 - val\_loss: 1.0784 - val\_acc: 0.6276

Epoch 00062: val\_loss did not improve



Epoch 63/100  
12160/12160 [=====] - 111s 9ms/step - loss: 1.0  
386 - acc: 0.6484 - val\_loss: 1.0802 - val\_acc: 0.6322

Epoch 00063: val\_loss did not improve

Epoch 64/100  
12160/12160 [=====] - 111s 9ms/step - loss: 1.0  
361 - acc: 0.6495 - val\_loss: 1.0623 - val\_acc: 0.6319

Epoch 00064: val\_loss did not improve

Epoch 65/100  
12160/12160 [=====] - 111s 9ms/step - loss: 1.0  
290 - acc: 0.6523 - val\_loss: 1.0615 - val\_acc: 0.6362

Epoch 00065: val\_loss did not improve

Epoch 66/100  
12160/12160 [=====] - 111s 9ms/step - loss: 1.0  
258 - acc: 0.6537 - val\_loss: 1.0353 - val\_acc: 0.6599

Epoch 00066: val\_loss improved from 1.05491 to 1.03532, saving model to  
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/  
weights.best.from\_scratch.hdf5

Epoch 67/100  
12160/12160 [=====] - 110s 9ms/step - loss: 1.0  
196 - acc: 0.6556 - val\_loss: 1.0382 - val\_acc: 0.6487

Epoch 00067: val\_loss did not improve

Epoch 68/100  
12160/12160 [=====] - 110s 9ms/step - loss: 1.0  
132 - acc: 0.6602 - val\_loss: 1.0383 - val\_acc: 0.6368

Epoch 00068: val\_loss did not improve

Epoch 69/100  
12160/12160 [=====] - 110s 9ms/step - loss: 1.0  
091 - acc: 0.6639 - val\_loss: 1.0319 - val\_acc: 0.6556

Epoch 00069: val\_loss improved from 1.03532 to 1.03191, saving model to  
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/  
weights.best.from\_scratch.hdf5

Epoch 70/100  
12160/12160 [=====] - 110s 9ms/step - loss: 1.0  
063 - acc: 0.6613 - val\_loss: 1.0315 - val\_acc: 0.6490

Epoch 00070: val\_loss improved from 1.03191 to 1.03154, saving model to  
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/  
weights.best.from\_scratch.hdf5

Epoch 71/100  
12160/12160 [=====] - 110s 9ms/step - loss: 0.9  
998 - acc: 0.6663 - val\_loss: 1.0411 - val\_acc: 0.6474

Epoch 00071: val\_loss did not improve

Epoch 72/100  
12160/12160 [=====] - 110s 9ms/step - loss: 0.9  
963 - acc: 0.6640 - val\_loss: 1.0291 - val\_acc: 0.6467

Epoch 00072: val\_loss improved from 1.03154 to 1.02912, saving model to  
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/

```
weights.best.from_scratch.hdf5
Epoch 73/100
12160/12160 [=====] - 110s 9ms/step - loss: 0.9
909 - acc: 0.6686 - val_loss: 1.0052 - val_acc: 0.6655

Epoch 00073: val_loss improved from 1.02912 to 1.00525, saving model to
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/
weights.best.from_scratch.hdf5
Epoch 74/100
12160/12160 [=====] - 111s 9ms/step - loss: 0.9
875 - acc: 0.6669 - val_loss: 1.0343 - val_acc: 0.6467

Epoch 00074: val_loss did not improve
Epoch 75/100
12160/12160 [=====] - 110s 9ms/step - loss: 0.9
810 - acc: 0.6720 - val_loss: 1.0253 - val_acc: 0.6461

Epoch 00075: val_loss did not improve
Epoch 76/100
12160/12160 [=====] - 110s 9ms/step - loss: 0.9
791 - acc: 0.6660 - val_loss: 1.0085 - val_acc: 0.6628

Epoch 00076: val_loss did not improve
Epoch 77/100
12160/12160 [=====] - 112s 9ms/step - loss: 0.9
716 - acc: 0.6733 - val_loss: 1.0503 - val_acc: 0.6326

Epoch 00077: val_loss did not improve
Epoch 78/100
12160/12160 [=====] - 111s 9ms/step - loss: 0.9
701 - acc: 0.6751 - val_loss: 1.0136 - val_acc: 0.6576

Epoch 00078: val_loss did not improve
Epoch 79/100
12160/12160 [=====] - 110s 9ms/step - loss: 0.9
636 - acc: 0.6756 - val_loss: 0.9672 - val_acc: 0.6724

Epoch 00079: val_loss improved from 1.00525 to 0.96717, saving model to
/mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/
weights.best.from_scratch.hdf5
Epoch 80/100
12160/12160 [=====] - 110s 9ms/step - loss: 0.9
608 - acc: 0.6791 - val_loss: 0.9959 - val_acc: 0.6635

Epoch 00080: val_loss did not improve
Epoch 81/100
12160/12160 [=====] - 110s 9ms/step - loss: 0.9
548 - acc: 0.6816 - val_loss: 1.0224 - val_acc: 0.6414

Epoch 00081: val_loss did not improve
Epoch 82/100
12160/12160 [=====] - 110s 9ms/step - loss: 0.9
531 - acc: 0.6799 - val_loss: 0.9847 - val_acc: 0.6572

Epoch 00082: val_loss did not improve
Epoch 83/100
12160/12160 [=====] - 110s 9ms/step - loss: 0.9
```

488 - acc: 0.6771 - val\_loss: 1.0104 - val\_acc: 0.6546

Epoch 00083: val\_loss did not improve

Epoch 84/100

12160/12160 [=====] - 110s 9ms/step - loss: 0.9

451 - acc: 0.6877 - val\_loss: 0.9646 - val\_acc: 0.6747

Epoch 00084: val\_loss improved from 0.96717 to 0.96457, saving model to /mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/weights.best.from\_scratch.hdf5

Epoch 85/100

12160/12160 [=====] - 110s 9ms/step - loss: 0.9

404 - acc: 0.6857 - val\_loss: 1.0071 - val\_acc: 0.6447

Epoch 00085: val\_loss did not improve

Epoch 86/100

12160/12160 [=====] - 110s 9ms/step - loss: 0.9

374 - acc: 0.6865 - val\_loss: 0.9982 - val\_acc: 0.6569

Epoch 00086: val\_loss did not improve

Epoch 87/100

12160/12160 [=====] - 110s 9ms/step - loss: 0.9

305 - acc: 0.6874 - val\_loss: 0.9890 - val\_acc: 0.6589

Epoch 00087: val\_loss did not improve

Epoch 88/100

12160/12160 [=====] - 110s 9ms/step - loss: 0.9

265 - acc: 0.6894 - val\_loss: 0.9636 - val\_acc: 0.6809

Epoch 00088: val\_loss improved from 0.96457 to 0.96358, saving model to /mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/weights.best.from\_scratch.hdf5

Epoch 89/100

12160/12160 [=====] - 110s 9ms/step - loss: 0.9

223 - acc: 0.6907 - val\_loss: 0.9669 - val\_acc: 0.6671

Epoch 00089: val\_loss did not improve

Epoch 90/100

12160/12160 [=====] - 110s 9ms/step - loss: 0.9

194 - acc: 0.6896 - val\_loss: 0.9508 - val\_acc: 0.6773

Epoch 00090: val\_loss improved from 0.96358 to 0.95077, saving model to /mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/weights.best.from\_scratch.hdf5

Epoch 91/100

12160/12160 [=====] - 110s 9ms/step - loss: 0.9

141 - acc: 0.6943 - val\_loss: 0.9354 - val\_acc: 0.6826

Epoch 00091: val\_loss improved from 0.95077 to 0.93536, saving model to /mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/weights.best.from\_scratch.hdf5

Epoch 92/100

12160/12160 [=====] - 110s 9ms/step - loss: 0.9

137 - acc: 0.6977 - val\_loss: 0.9546 - val\_acc: 0.6770

Epoch 00092: val\_loss did not improve

Epoch 93/100

12160/12160 [=====] - 110s 9ms/step - loss: 0.9081 - acc: 0.6984 - val\_loss: 0.9201 - val\_acc: 0.6980

Epoch 00093: val\_loss improved from 0.93536 to 0.92013, saving model to /mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/weights.best.from\_scratch.hdf5

Epoch 94/100

12160/12160 [=====] - 110s 9ms/step - loss: 0.9047 - acc: 0.6954 - val\_loss: 0.9619 - val\_acc: 0.6704

Epoch 00094: val\_loss did not improve

Epoch 95/100

12160/12160 [=====] - 110s 9ms/step - loss: 0.9019 - acc: 0.6975 - val\_loss: 0.9224 - val\_acc: 0.6868

Epoch 00095: val\_loss did not improve

Epoch 96/100

12160/12160 [=====] - 109s 9ms/step - loss: 0.9005 - acc: 0.6954 - val\_loss: 0.9397 - val\_acc: 0.6895

Epoch 00096: val\_loss did not improve

Epoch 97/100

12160/12160 [=====] - 110s 9ms/step - loss: 0.8947 - acc: 0.7001 - val\_loss: 0.9636 - val\_acc: 0.6783

Epoch 00097: val\_loss did not improve

Epoch 98/100

12160/12160 [=====] - 110s 9ms/step - loss: 0.8927 - acc: 0.7025 - val\_loss: 0.9344 - val\_acc: 0.6849

Epoch 00098: val\_loss did not improve

Epoch 99/100

12160/12160 [=====] - 111s 9ms/step - loss: 0.8900 - acc: 0.7035 - val\_loss: 0.9190 - val\_acc: 0.6868

Epoch 00099: val\_loss improved from 0.92013 to 0.91903, saving model to /mnt/c/Users/pushkar/ML/machine-learning/projects/capstone/saved\_models/weights.best.from\_scratch.hdf5

Epoch 100/100

12160/12160 [=====] - 111s 9ms/step - loss: 0.8880 - acc: 0.7052 - val\_loss: 0.9556 - val\_acc: 0.6845

Epoch 00100: val\_loss did not improve

```

In [18]: import matplotlib.pyplot as plt
import numpy as np

# summarize history for accuracy
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

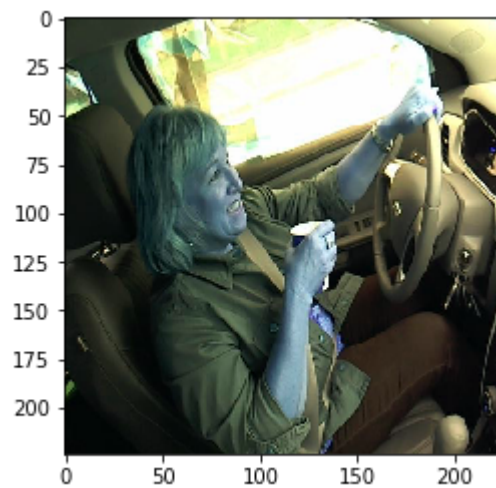
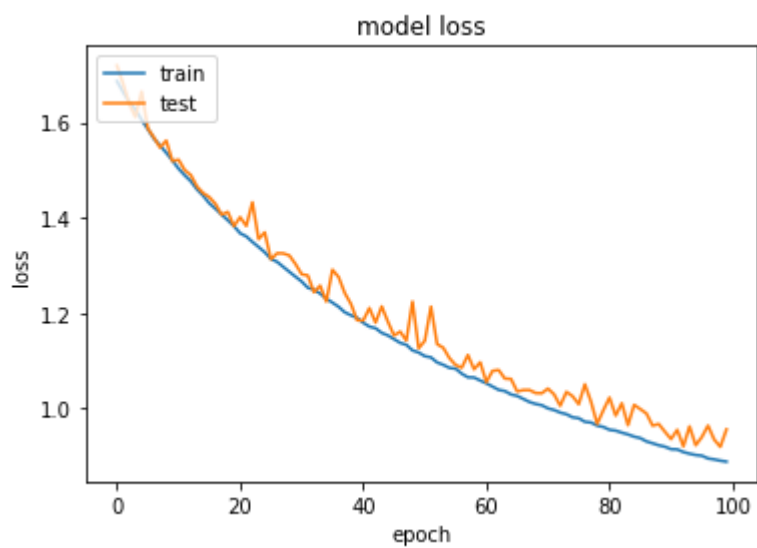
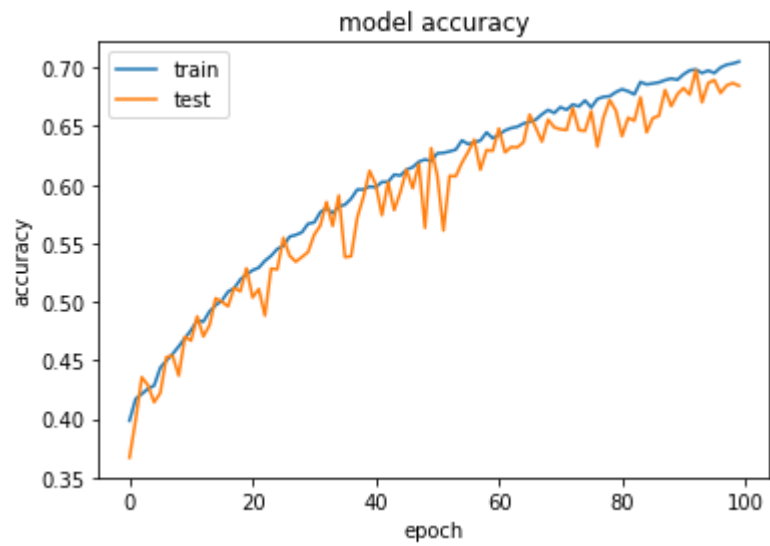
p = model.predict(test_tensors)
#print (p)
z=np.argmax(p,axis=1)
#print("z = ", z)
for i in range(1,15):
    img = np.squeeze(np.array(test_tensors[i]))
    displayImage(img)
    print("Predicted class", getClass(z[i]))
    print ("Actual Class", getClass(test_targets[i]))

# def predict_distraction:
#     # get index of predicted distraction for each image in test set
#     distraction_predictions = [np.argmax(model.predict(np.expand_dims(tensor, axis=0))) for tensor in test_tensors]

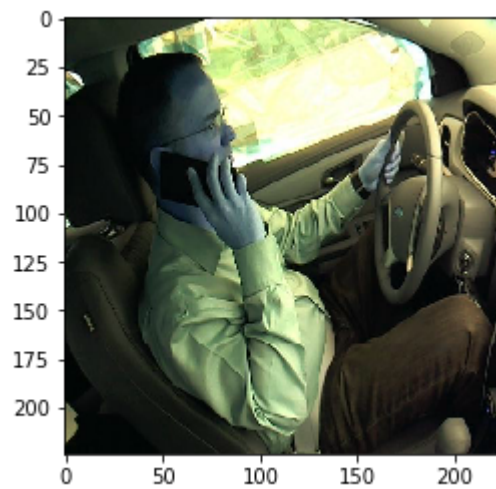
#     # report test accuracy
#     test_accuracy = 100*np.sum(np.array(distraction_predictions)==np.argmax(test_targets, axis=0))/len(distraction_predictions)
#     print('Test accuracy: %.4f%%' % test_accuracy)
#     return test_accuracy

# predict_distraction()

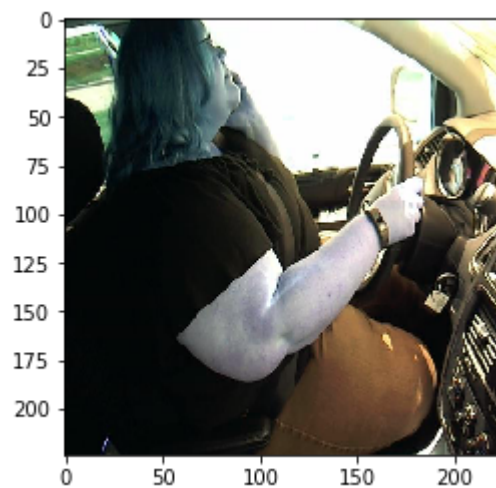
```



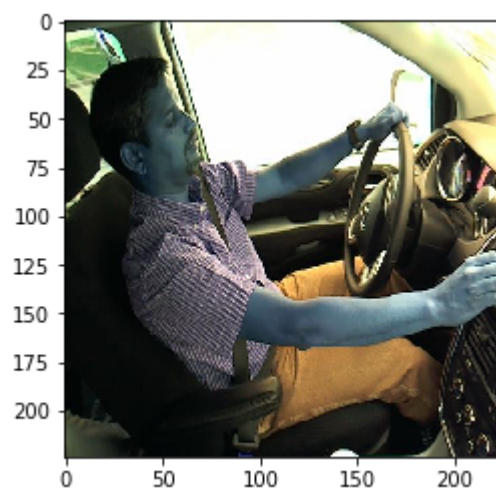
('Predicted class', 'drinking')  
('Actual Class', 'drinking')



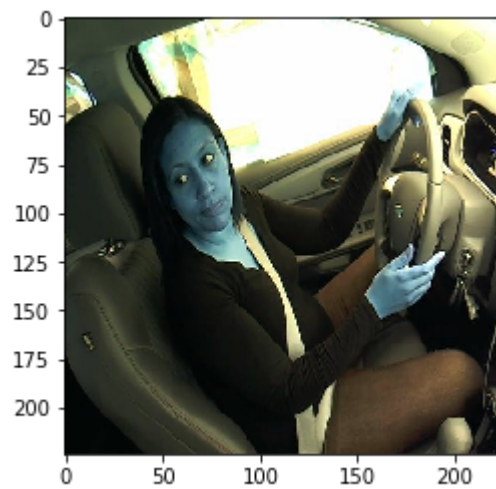
('Predicted class', 'hair and makeup')  
( 'Actual Class', 'talking on the phone - right')



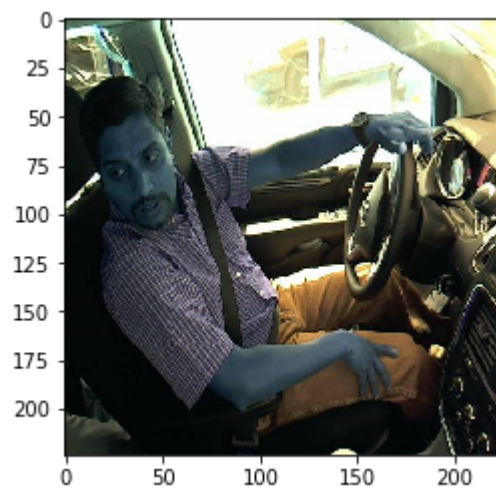
('Predicted class', 'talking to passenger')  
( 'Actual Class', 'talking on the phone - left')



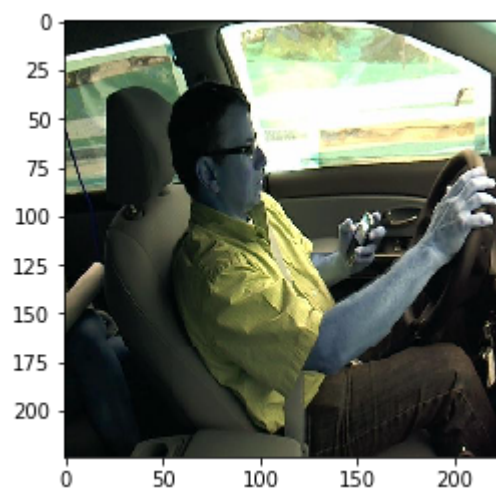
('Predicted class', 'operating the radio')  
( 'Actual Class', 'operating the radio')



('Predicted class', 'hair and makeup')  
( 'Actual Class', 'talking to passenger')

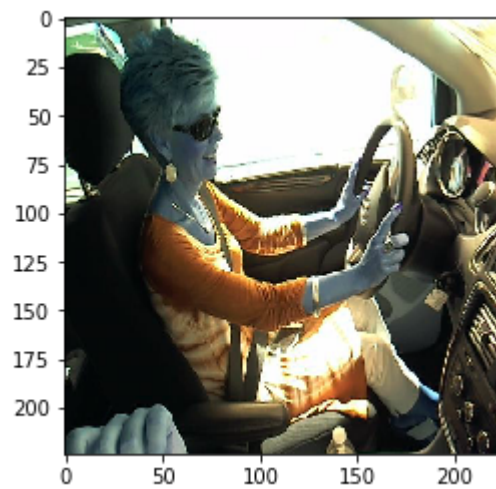


('Predicted class', 'talking to passenger')  
( 'Actual Class', 'talking to passenger')

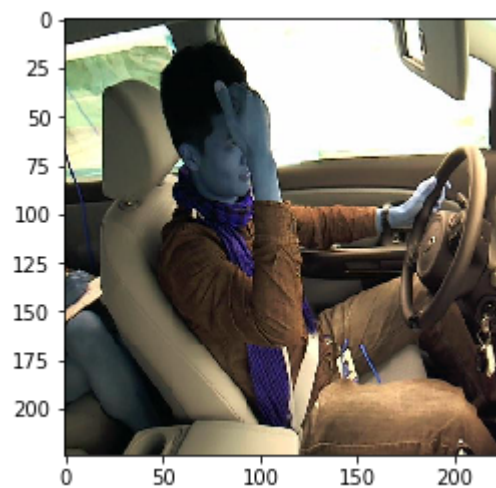


('Predicted class', 'talking on the phone - left')  
( 'Actual Class', 'texting - left')

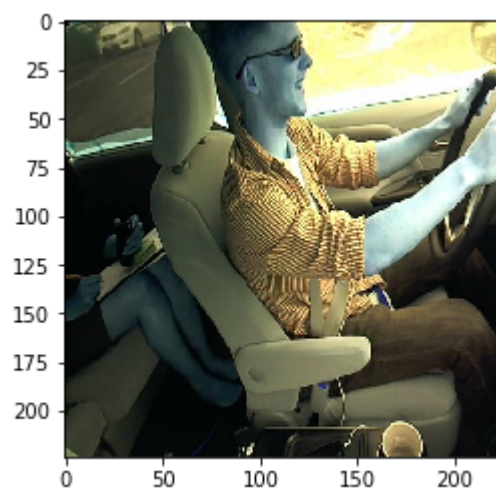




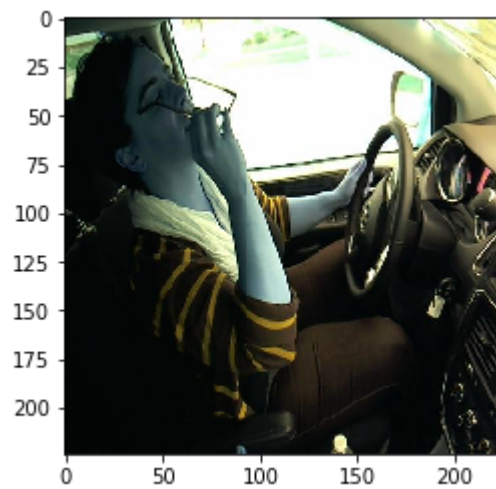
('Predicted class', 'talking to passenger')  
( 'Actual Class', 'safe driving')



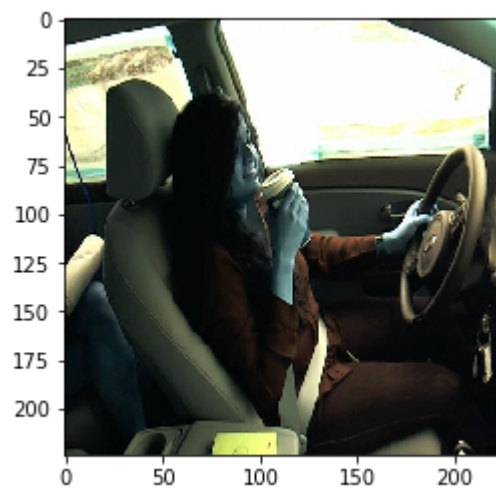
('Predicted class', 'hair and makeup')  
( 'Actual Class', 'hair and makeup')



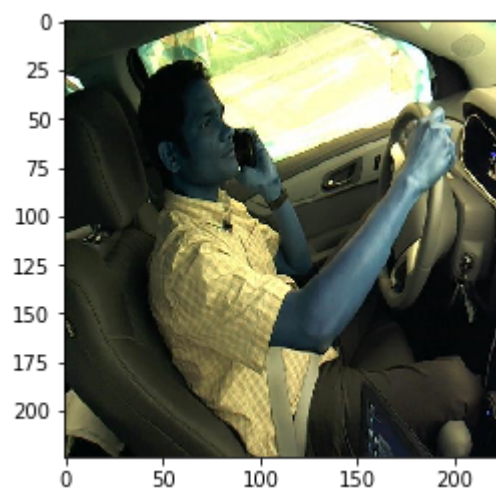
('Predicted class', 'safe driving')  
( 'Actual Class', 'safe driving')



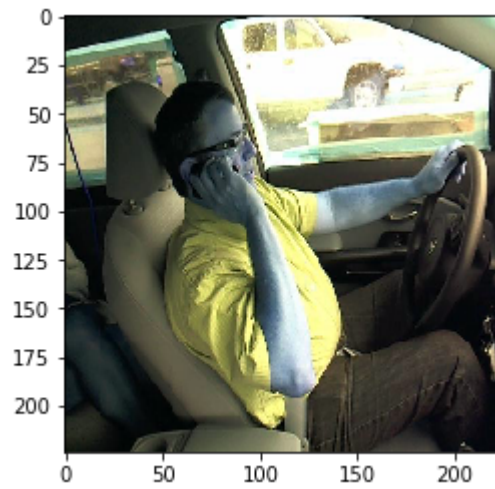
('Predicted class', 'hair and makeup')  
( 'Actual Class', 'hair and makeup')



('Predicted class', 'drinking')  
( 'Actual Class', 'drinking')



('Predicted class', 'texting - left')  
( 'Actual Class', 'talking on the phone - left')



('Predicted class', 'drinking')  
( 'Actual Class', 'talking on the phone - right')