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

import tensorflow as tf
hello = tf.constant('Hello, TensorFlow!')
sess = tf.Session()
print(sess.run(hello))

# import tensorflow as tf
# from keras import backend as K

# num_cores = 4
# GPU = 1
# CPU = 0

# if GPU:
#     num_GPU = 1
#     num_CPU = 1
# if CPU:
#     num_CPU = 1
#     num_GPU = 0

# config = tf.ConfigProto(intra_op_parallelism_threads=num_cores,\
```

```
#         inter_op_parallelism_threads=num_cores, allow_soft_placement=True,\
#         device_count = {'CPU' : num_CPU, 'GPU' : num_GPU})
# session = tf.Session(config=config)
# K.set_session(session)

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))
```

```
Using TensorFlow backend.
```

```
b'Hello, TensorFlow!'
Number of Categories:  10
Categories:  ['c0', 'c1', 'c2', 'c3', 'c4', 'c5', 'c6', 'c7', 'c8', 'c9']
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
```

```
['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 [02:19<00:00, 108.81it/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 [00:35<00:00, 106.93it/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_f
ormat=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 (MaxPooling2 (None, 55, 55, 10)        0
_________________________________________________________________
global_average_pooling2d_1 ( (None, 10)                0
_________________________________________________________________
dense_1 (Dense)              (None, 10)                110
=================================================================
Total params: 600
Trainable params: 600
Non-trainable params: 0
_________________________________________________________________
```

Train the Model

In [7]:
```
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='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=2)
    return history

history = train_model(100)
```

```
Train Targets [3 8 9 ..., 7 7 5]
Test Targets [5 6 2 ..., 8 6 5]
Train Targets One-hot encoded [[ 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 [[ 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
 - 213s - loss: 2.3059 - acc: 0.0995 - val_loss: 2.3036 - val_acc: 0.093
1

Epoch 00001: val_loss improved from inf to 2.30365, saving model to C:/U
sers/pushkar/ML/machine-learning/projects/capstone/saved_models/weights.
best.from_scratch.hdf5
Epoch 2/100
 - 64s - loss: 2.3043 - acc: 0.1000 - val_loss: 2.3076 - val_acc: 0.0957

Epoch 00002: val_loss did not improve
Epoch 3/100
 - 64s - loss: 2.3034 - acc: 0.1011 - val_loss: 2.3074 - val_acc: 0.0974

Epoch 00003: val_loss did not improve
Epoch 4/100
 - 64s - loss: 2.3025 - acc: 0.1065 - val_loss: 2.3010 - val_acc: 0.1062

Epoch 00004: val_loss improved from 2.30365 to 2.30102, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 5/100
 - 64s - loss: 2.3011 - acc: 0.1109 - val_loss: 2.3014 - val_acc: 0.0934

Epoch 00005: val_loss did not improve
Epoch 6/100
 - 66s - loss: 2.3004 - acc: 0.1102 - val_loss: 2.3002 - val_acc: 0.1299

Epoch 00006: val_loss improved from 2.30102 to 2.30019, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 7/100
 - 66s - loss: 2.2995 - acc: 0.1149 - val_loss: 2.3022 - val_acc: 0.0954

Epoch 00007: val_loss did not improve
Epoch 8/100
 - 64s - loss: 2.2980 - acc: 0.1156 - val_loss: 2.2987 - val_acc: 0.1191
```

```
Epoch 00008: val_loss improved from 2.30019 to 2.29867, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 9/100
 - 65s - loss: 2.2965 - acc: 0.1221 - val_loss: 2.2961 - val_acc: 0.1128

Epoch 00009: val_loss improved from 2.29867 to 2.29610, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 10/100
 - 68s - loss: 2.2961 - acc: 0.1157 - val_loss: 2.2949 - val_acc: 0.1118

Epoch 00010: val_loss improved from 2.29610 to 2.29488, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 11/100
 - 66s - loss: 2.2944 - acc: 0.1223 - val_loss: 2.2971 - val_acc: 0.1224

Epoch 00011: val_loss did not improve
Epoch 12/100
 - 69s - loss: 2.2932 - acc: 0.1248 - val_loss: 2.2930 - val_acc: 0.1283

Epoch 00012: val_loss improved from 2.29488 to 2.29302, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 13/100
 - 69s - loss: 2.2926 - acc: 0.1277 - val_loss: 2.2904 - val_acc: 0.1451

Epoch 00013: val_loss improved from 2.29302 to 2.29040, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 14/100
 - 66s - loss: 2.2904 - acc: 0.1330 - val_loss: 2.2895 - val_acc: 0.1158

Epoch 00014: val_loss improved from 2.29040 to 2.28954, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 15/100
 - 67s - loss: 2.2889 - acc: 0.1299 - val_loss: 2.2874 - val_acc: 0.1368

Epoch 00015: val_loss improved from 2.28954 to 2.28744, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 16/100
 - 66s - loss: 2.2877 - acc: 0.1343 - val_loss: 2.2927 - val_acc: 0.1151

Epoch 00016: val_loss did not improve
Epoch 17/100
 - 68s - loss: 2.2861 - acc: 0.1369 - val_loss: 2.2890 - val_acc: 0.1395

Epoch 00017: val_loss did not improve
Epoch 18/100
 - 69s - loss: 2.2846 - acc: 0.1341 - val_loss: 2.2867 - val_acc: 0.1336

Epoch 00018: val_loss improved from 2.28744 to 2.28674, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
```

```
Epoch 19/100
 - 68s - loss: 2.2826 - acc: 0.1399 - val_loss: 2.2854 - val_acc: 0.1470

Epoch 00019: val_loss improved from 2.28674 to 2.28541, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 20/100
 - 67s - loss: 2.2815 - acc: 0.1419 - val_loss: 2.2842 - val_acc: 0.1408

Epoch 00020: val_loss improved from 2.28541 to 2.28422, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 21/100
 - 68s - loss: 2.2793 - acc: 0.1386 - val_loss: 2.2785 - val_acc: 0.1615

Epoch 00021: val_loss improved from 2.28422 to 2.27851, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 22/100
 - 67s - loss: 2.2790 - acc: 0.1433 - val_loss: 2.2775 - val_acc: 0.1451

Epoch 00022: val_loss improved from 2.27851 to 2.27748, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 23/100
 - 68s - loss: 2.2771 - acc: 0.1465 - val_loss: 2.2789 - val_acc: 0.1451

Epoch 00023: val_loss did not improve
Epoch 24/100
 - 66s - loss: 2.2754 - acc: 0.1492 - val_loss: 2.2750 - val_acc: 0.1520

Epoch 00024: val_loss improved from 2.27748 to 2.27502, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 25/100
 - 65s - loss: 2.2739 - acc: 0.1463 - val_loss: 2.2833 - val_acc: 0.1270

Epoch 00025: val_loss did not improve
Epoch 26/100
 - 65s - loss: 2.2725 - acc: 0.1486 - val_loss: 2.2790 - val_acc: 0.1447

Epoch 00026: val_loss did not improve
Epoch 27/100
 - 66s - loss: 2.2718 - acc: 0.1498 - val_loss: 2.2710 - val_acc: 0.1789

Epoch 00027: val_loss improved from 2.27502 to 2.27099, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 28/100
 - 66s - loss: 2.2697 - acc: 0.1542 - val_loss: 2.2955 - val_acc: 0.1309

Epoch 00028: val_loss did not improve
Epoch 29/100
 - 66s - loss: 2.2682 - acc: 0.1591 - val_loss: 2.2699 - val_acc: 0.1615

Epoch 00029: val_loss improved from 2.27099 to 2.26993, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
```

```
hts.best.from_scratch.hdf5
Epoch 30/100
 - 65s - loss: 2.2670 - acc: 0.1539 - val_loss: 2.2712 - val_acc: 0.1490

Epoch 00030: val_loss did not improve
Epoch 31/100
 - 66s - loss: 2.2662 - acc: 0.1594 - val_loss: 2.2697 - val_acc: 0.1378

Epoch 00031: val_loss improved from 2.26993 to 2.26970, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 32/100
 - 65s - loss: 2.2648 - acc: 0.1563 - val_loss: 2.2694 - val_acc: 0.1632

Epoch 00032: val_loss improved from 2.26970 to 2.26940, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 33/100
 - 65s - loss: 2.2640 - acc: 0.1572 - val_loss: 2.2643 - val_acc: 0.1562

Epoch 00033: val_loss improved from 2.26940 to 2.26430, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 34/100
 - 65s - loss: 2.2625 - acc: 0.1594 - val_loss: 2.2658 - val_acc: 0.1526

Epoch 00034: val_loss did not improve
Epoch 35/100
 - 65s - loss: 2.2604 - acc: 0.1615 - val_loss: 2.2698 - val_acc: 0.1444

Epoch 00035: val_loss did not improve
Epoch 36/100
 - 65s - loss: 2.2586 - acc: 0.1604 - val_loss: 2.2661 - val_acc: 0.1536

Epoch 00036: val_loss did not improve
Epoch 37/100
 - 68s - loss: 2.2578 - acc: 0.1623 - val_loss: 2.2609 - val_acc: 0.1697

Epoch 00037: val_loss improved from 2.26430 to 2.26091, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 38/100
 - 67s - loss: 2.2563 - acc: 0.1606 - val_loss: 2.2597 - val_acc: 0.1793

Epoch 00038: val_loss improved from 2.26091 to 2.25968, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 39/100
 - 65s - loss: 2.2562 - acc: 0.1637 - val_loss: 2.2588 - val_acc: 0.1681

Epoch 00039: val_loss improved from 2.25968 to 2.25877, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 40/100
 - 65s - loss: 2.2529 - acc: 0.1648 - val_loss: 2.2909 - val_acc: 0.1296

Epoch 00040: val_loss did not improve
```

```
Epoch 41/100
 - 64s - loss: 2.2533 - acc: 0.1683 - val_loss: 2.2570 - val_acc: 0.1490

Epoch 00041: val_loss improved from 2.25877 to 2.25703, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 42/100
 - 64s - loss: 2.2530 - acc: 0.1674 - val_loss: 2.2653 - val_acc: 0.1602

Epoch 00042: val_loss did not improve
Epoch 43/100
 - 65s - loss: 2.2522 - acc: 0.1647 - val_loss: 2.2579 - val_acc: 0.1707

Epoch 00043: val_loss did not improve
Epoch 44/100
 - 65s - loss: 2.2501 - acc: 0.1719 - val_loss: 2.2607 - val_acc: 0.1612

Epoch 00044: val_loss did not improve
Epoch 45/100
 - 65s - loss: 2.2496 - acc: 0.1702 - val_loss: 2.2505 - val_acc: 0.1714

Epoch 00045: val_loss improved from 2.25703 to 2.25046, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 46/100
 - 65s - loss: 2.2483 - acc: 0.1660 - val_loss: 2.2757 - val_acc: 0.1461

Epoch 00046: val_loss did not improve
Epoch 47/100
 - 67s - loss: 2.2478 - acc: 0.1703 - val_loss: 2.2572 - val_acc: 0.1618

Epoch 00047: val_loss did not improve
Epoch 48/100
 - 66s - loss: 2.2470 - acc: 0.1707 - val_loss: 2.2559 - val_acc: 0.1740

Epoch 00048: val_loss did not improve
Epoch 49/100
 - 64s - loss: 2.2464 - acc: 0.1718 - val_loss: 2.2498 - val_acc: 0.1806

Epoch 00049: val_loss improved from 2.25046 to 2.24976, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 50/100
 - 69s - loss: 2.2445 - acc: 0.1708 - val_loss: 2.2552 - val_acc: 0.1474

Epoch 00050: val_loss did not improve
Epoch 51/100
 - 66s - loss: 2.2438 - acc: 0.1675 - val_loss: 2.2486 - val_acc: 0.1760

Epoch 00051: val_loss improved from 2.24976 to 2.24856, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 52/100
 - 67s - loss: 2.2429 - acc: 0.1721 - val_loss: 2.2471 - val_acc: 0.1727

Epoch 00052: val_loss improved from 2.24856 to 2.24715, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
```

```
hts.best.from_scratch.hdf5
Epoch 53/100
 - 66s - loss: 2.2421 - acc: 0.1681 - val_loss: 2.2578 - val_acc: 0.1431

Epoch 00053: val_loss did not improve
Epoch 54/100
 - 66s - loss: 2.2417 - acc: 0.1729 - val_loss: 2.2507 - val_acc: 0.1668

Epoch 00054: val_loss did not improve
Epoch 55/100
 - 66s - loss: 2.2424 - acc: 0.1747 - val_loss: 2.2419 - val_acc: 0.1822

Epoch 00055: val_loss improved from 2.24715 to 2.24191, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 56/100
 - 66s - loss: 2.2397 - acc: 0.1735 - val_loss: 2.2455 - val_acc: 0.1648

Epoch 00056: val_loss did not improve
Epoch 57/100
 - 66s - loss: 2.2374 - acc: 0.1712 - val_loss: 2.2418 - val_acc: 0.1763

Epoch 00057: val_loss improved from 2.24191 to 2.24178, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 58/100
 - 65s - loss: 2.2382 - acc: 0.1682 - val_loss: 2.2425 - val_acc: 0.1714

Epoch 00058: val_loss did not improve
Epoch 59/100
 - 65s - loss: 2.2375 - acc: 0.1728 - val_loss: 2.2428 - val_acc: 0.1895

Epoch 00059: val_loss did not improve
Epoch 60/100
 - 65s - loss: 2.2361 - acc: 0.1754 - val_loss: 2.2453 - val_acc: 0.1717

Epoch 00060: val_loss did not improve
Epoch 61/100
 - 65s - loss: 2.2352 - acc: 0.1754 - val_loss: 2.2463 - val_acc: 0.1793

Epoch 00061: val_loss did not improve
Epoch 62/100
 - 65s - loss: 2.2337 - acc: 0.1812 - val_loss: 2.2384 - val_acc: 0.1862

Epoch 00062: val_loss improved from 2.24178 to 2.23841, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 63/100
 - 66s - loss: 2.2340 - acc: 0.1736 - val_loss: 2.2402 - val_acc: 0.1750

Epoch 00063: val_loss did not improve
Epoch 64/100
 - 65s - loss: 2.2314 - acc: 0.1783 - val_loss: 2.2357 - val_acc: 0.1658

Epoch 00064: val_loss improved from 2.23841 to 2.23568, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
```

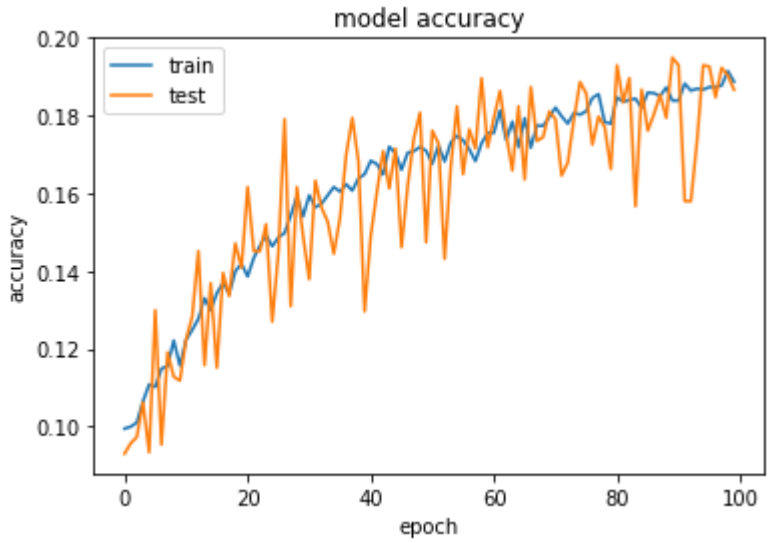
```
Epoch 65/100
 - 65s - loss: 2.2316 - acc: 0.1718 - val_loss: 2.2359 - val_acc: 0.1822

Epoch 00065: val_loss did not improve
Epoch 66/100
 - 64s - loss: 2.2289 - acc: 0.1792 - val_loss: 2.2356 - val_acc: 0.1635

Epoch 00066: val_loss improved from 2.23568 to 2.23563, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 67/100
 - 65s - loss: 2.2294 - acc: 0.1716 - val_loss: 2.2335 - val_acc: 0.1872

Epoch 00067: val_loss improved from 2.23563 to 2.23351, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 68/100
 - 64s - loss: 2.2274 - acc: 0.1773 - val_loss: 2.2354 - val_acc: 0.1734

Epoch 00068: val_loss did not improve
Epoch 69/100
 - 64s - loss: 2.2272 - acc: 0.1772 - val_loss: 2.2320 - val_acc: 0.1743

Epoch 00069: val_loss improved from 2.23351 to 2.23196, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 70/100
 - 65s - loss: 2.2261 - acc: 0.1795 - val_loss: 2.2294 - val_acc: 0.1809

Epoch 00070: val_loss improved from 2.23196 to 2.22936, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 71/100
 - 65s - loss: 2.2256 - acc: 0.1819 - val_loss: 2.2332 - val_acc: 0.1789

Epoch 00071: val_loss did not improve
Epoch 72/100
 - 65s - loss: 2.2243 - acc: 0.1797 - val_loss: 2.2584 - val_acc: 0.1645

Epoch 00072: val_loss did not improve
Epoch 73/100
 - 67s - loss: 2.2244 - acc: 0.1778 - val_loss: 2.2333 - val_acc: 0.1678

Epoch 00073: val_loss did not improve
Epoch 74/100
 - 69s - loss: 2.2244 - acc: 0.1805 - val_loss: 2.2435 - val_acc: 0.1789

Epoch 00074: val_loss did not improve
Epoch 75/100
 - 69s - loss: 2.2233 - acc: 0.1801 - val_loss: 2.2248 - val_acc: 0.1885

Epoch 00075: val_loss improved from 2.22936 to 2.22480, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 76/100
 - 66s - loss: 2.2224 - acc: 0.1810 - val_loss: 2.2245 - val_acc: 0.1855
```

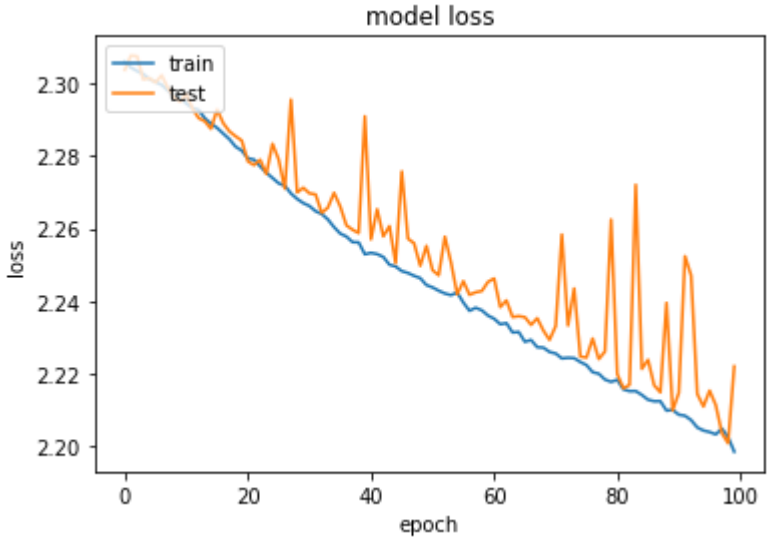
```
Epoch 00076: val_loss improved from 2.22480 to 2.22451, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 77/100
 - 65s - loss: 2.2205 - acc: 0.1844 - val_loss: 2.2298 - val_acc: 0.1724

Epoch 00077: val_loss did not improve
Epoch 78/100
 - 65s - loss: 2.2201 - acc: 0.1854 - val_loss: 2.2241 - val_acc: 0.1796

Epoch 00078: val_loss improved from 2.22451 to 2.22410, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 79/100
 - 65s - loss: 2.2185 - acc: 0.1782 - val_loss: 2.2262 - val_acc: 0.1770

Epoch 00079: val_loss did not improve
Epoch 80/100
 - 64s - loss: 2.2178 - acc: 0.1778 - val_loss: 2.2625 - val_acc: 0.1661

Epoch 00080: val_loss did not improve
Epoch 81/100
 - 64s - loss: 2.2184 - acc: 0.1845 - val_loss: 2.2200 - val_acc: 0.1928

Epoch 00081: val_loss improved from 2.22410 to 2.21999, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 82/100
 - 65s - loss: 2.2158 - acc: 0.1834 - val_loss: 2.2161 - val_acc: 0.1832

Epoch 00082: val_loss improved from 2.21999 to 2.21609, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 83/100
 - 65s - loss: 2.2153 - acc: 0.1839 - val_loss: 2.2170 - val_acc: 0.1895

Epoch 00083: val_loss did not improve
Epoch 84/100
 - 65s - loss: 2.2153 - acc: 0.1842 - val_loss: 2.2721 - val_acc: 0.1566

Epoch 00084: val_loss did not improve
Epoch 85/100
 - 65s - loss: 2.2142 - acc: 0.1818 - val_loss: 2.2214 - val_acc: 0.1865

Epoch 00085: val_loss did not improve
Epoch 86/100
 - 65s - loss: 2.2129 - acc: 0.1858 - val_loss: 2.2239 - val_acc: 0.1760

Epoch 00086: val_loss did not improve
Epoch 87/100
 - 65s - loss: 2.2125 - acc: 0.1856 - val_loss: 2.2169 - val_acc: 0.1803

Epoch 00087: val_loss did not improve
Epoch 88/100
 - 64s - loss: 2.2125 - acc: 0.1849 - val_loss: 2.2150 - val_acc: 0.1852

Epoch 00088: val_loss improved from 2.21609 to 2.21496, saving model to
```

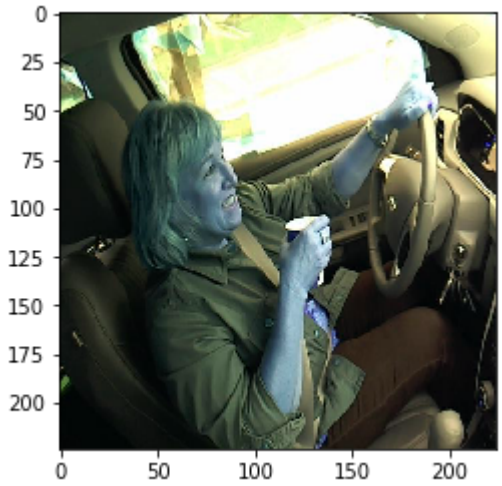
```
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 89/100
 - 64s - loss: 2.2099 - acc: 0.1871 - val_loss: 2.2396 - val_acc: 0.1793

Epoch 00089: val_loss did not improve
Epoch 90/100
 - 64s - loss: 2.2101 - acc: 0.1837 - val_loss: 2.2100 - val_acc: 0.1947

Epoch 00090: val_loss improved from 2.21496 to 2.20996, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 91/100
 - 64s - loss: 2.2088 - acc: 0.1837 - val_loss: 2.2148 - val_acc: 0.1928

Epoch 00091: val_loss did not improve
Epoch 92/100
 - 64s - loss: 2.2085 - acc: 0.1881 - val_loss: 2.2524 - val_acc: 0.1579

Epoch 00092: val_loss did not improve
Epoch 93/100
 - 64s - loss: 2.2073 - acc: 0.1863 - val_loss: 2.2471 - val_acc: 0.1579

Epoch 00093: val_loss did not improve
Epoch 94/100
 - 64s - loss: 2.2053 - acc: 0.1868 - val_loss: 2.2145 - val_acc: 0.1734

Epoch 00094: val_loss did not improve
Epoch 95/100
 - 64s - loss: 2.2044 - acc: 0.1865 - val_loss: 2.2111 - val_acc: 0.1928

Epoch 00095: val_loss did not improve
Epoch 96/100
 - 64s - loss: 2.2040 - acc: 0.1872 - val_loss: 2.2154 - val_acc: 0.1924

Epoch 00096: val_loss did not improve
Epoch 97/100
 - 64s - loss: 2.2034 - acc: 0.1872 - val_loss: 2.2114 - val_acc: 0.1845

Epoch 00097: val_loss did not improve
Epoch 98/100
 - 64s - loss: 2.2050 - acc: 0.1876 - val_loss: 2.2037 - val_acc: 0.1921

Epoch 00098: val_loss improved from 2.20996 to 2.20372, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 99/100
 - 64s - loss: 2.2027 - acc: 0.1913 - val_loss: 2.2010 - val_acc: 0.1901

Epoch 00099: val_loss improved from 2.20372 to 2.20095, saving model to
C:/Users/pushkar/ML/machine-learning/projects/capstone/saved_models/weig
hts.best.from_scratch.hdf5
Epoch 100/100
 - 64s - loss: 2.1986 - acc: 0.1886 - val_loss: 2.2221 - val_acc: 0.1865

Epoch 00100: val_loss did not improve
```

In [8]:
```
import matplotlib.pyplot as plt
import numpy as py

print (history)

# 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(tenso
r, 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()
```

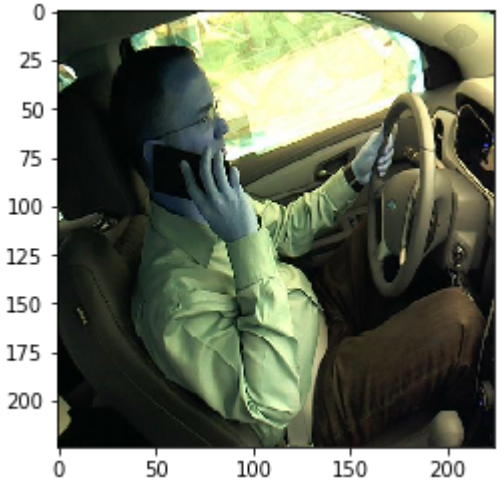
```
<keras.callbacks.History object at 0x000001DF4CF3F6A0>
```



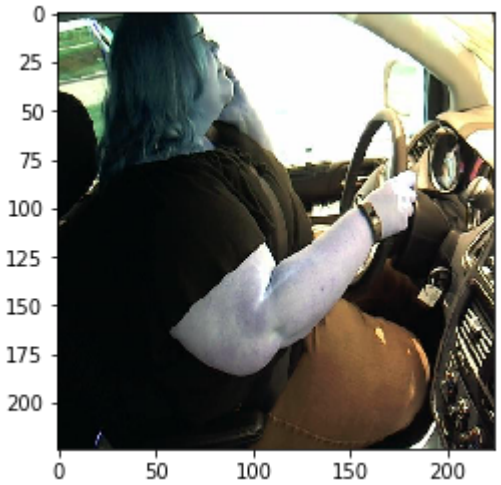




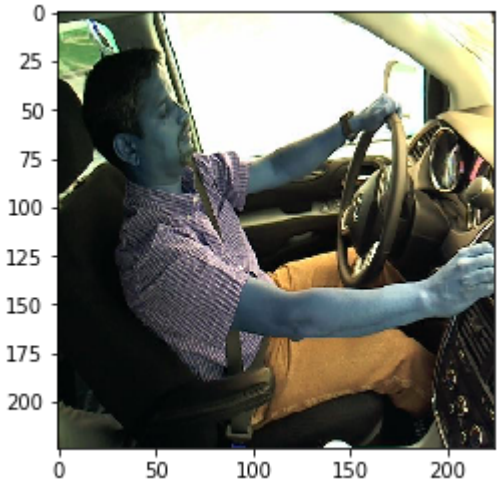

```
Predicted class safe driving
Actual Class drinking
```



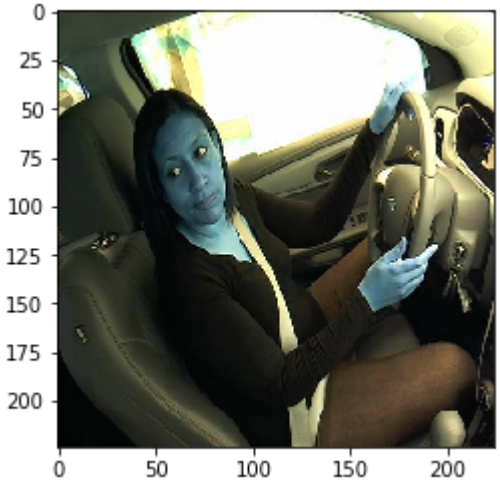

```
Predicted class safe driving
Actual Class talking on the phone - right
```



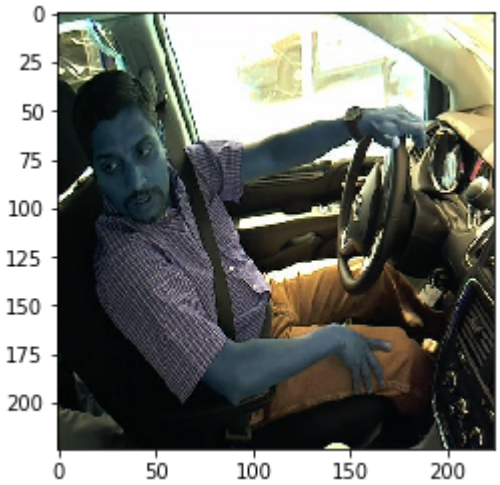

```
Predicted class drinking
Actual Class talking on the phone - left
```



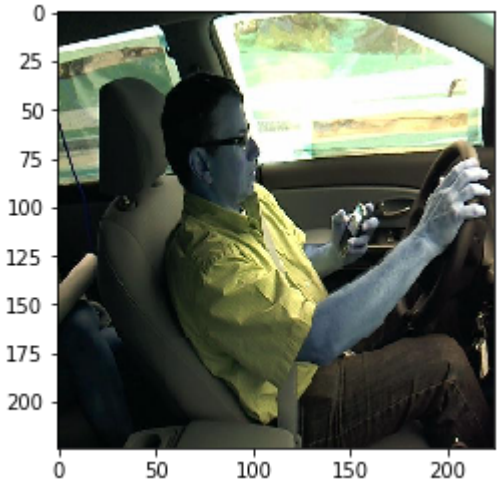

```
Predicted class talking to passenger
Actual Class operating the radio
```



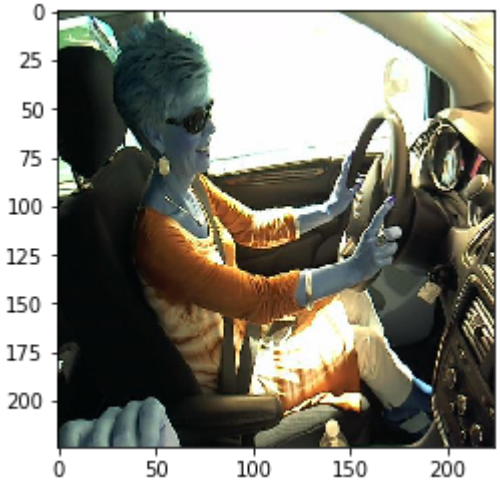

```
Predicted class safe driving
Actual Class talking to passenger
```



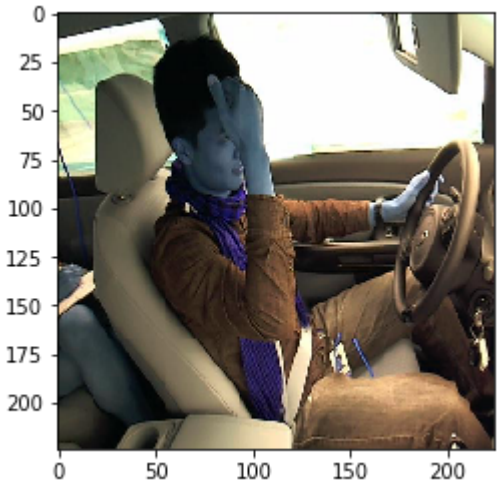

```
Predicted class talking to passenger
Actual Class talking to passenger
```



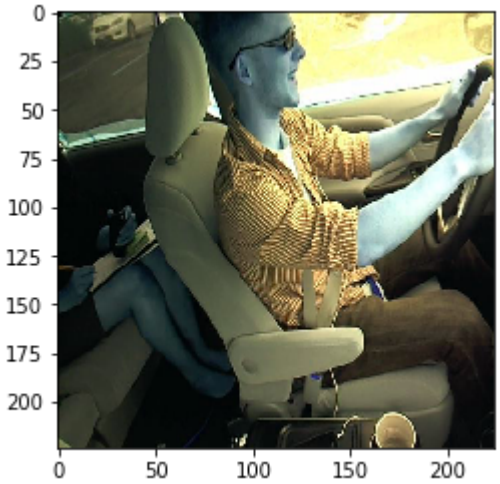

```
Predicted class safe driving
Actual Class texting - left
```



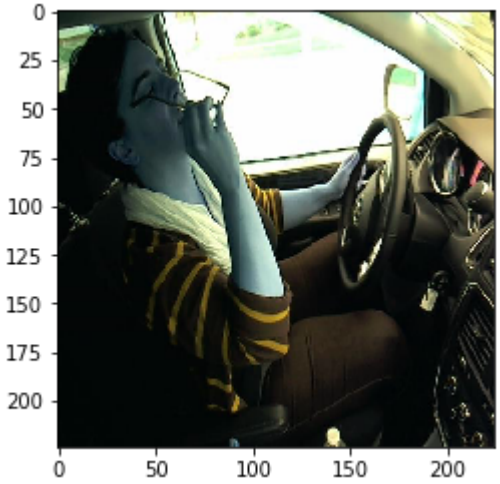

```
Predicted class talking to passenger
Actual Class safe driving
```



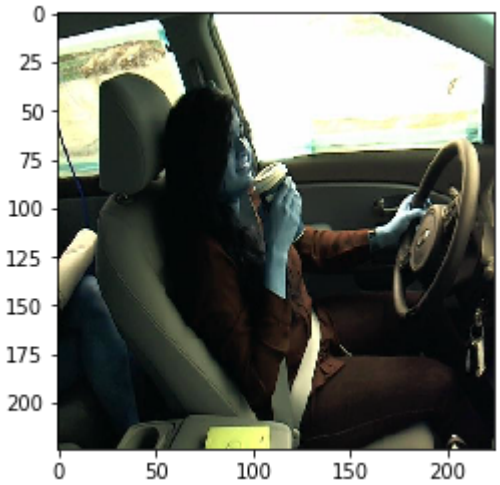

```
Predicted class operating the radio
Actual Class hair and makeup
```



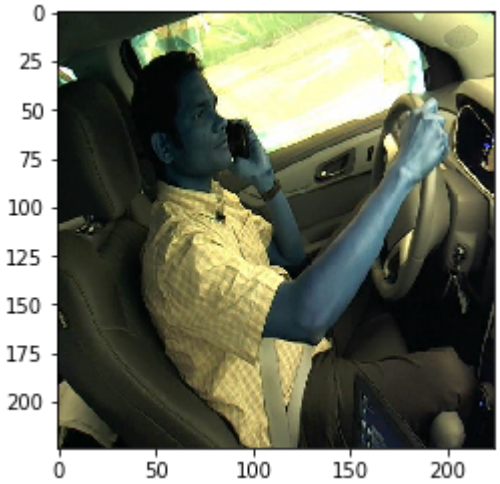

```
Predicted class safe driving
Actual Class safe driving
```



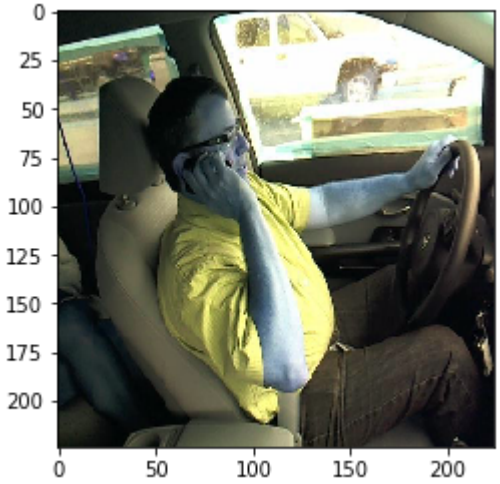

```
Predicted class safe driving
Actual Class hair and makeup
```




```
Predicted class safe driving
Actual Class drinking
```




```
Predicted class safe driving
Actual Class talking on the phone - left
```




```
Predicted class operating the radio
Actual Class talking on the phone - right
```