Assignment 1C - Question 2

Semantic Person Search

In [1]:

```
from keras.models import Sequential
from keras_preprocessing.image import ImageDataGenerator
from keras.layers import Dense, Activation, Flatten, Dropout, BatchNormalization
from keras.layers import Conv2D, MaxPooling2D
from keras import regularizers, optimizers
import pandas as pd
import numpy as np
import glob
import cv2
import matplotlib.pyplot as plt
import keras
from keras import layers
from PIL import Image

import os
import datetime
import numpy

import tensorflow as tf

from tensorflow import keras
from tensorflow.keras import layers
from tensorboard import notebook
from tensorflow.keras.preprocessing.image import Iterator

from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
import matplotlib.pyplot as plt

import pydot
import IPython
from IPython.display import SVG
from tensorflow.keras.utils import model_to_dot, plot_model
import imageio
import cv2
```

```
INFO:tensorflow:Enabling eager execution
INFO:tensorflow:Enabling v2 tensorshape
INFO:tensorflow:Enabling resource variables
INFO:tensorflow:Enabling tensor equality
INFO:tensorflow:Enabling control flow v2
```

In [2]:

```
train = pd.read_csv('CAB420_Assessment_1C_Data/Data/Q2/Q2/Train_Data/Train.csv')
test = pd.read_csv('CAB420_Assessment_1C_Data/Data/Q2/Q2/Test_Data/Test.csv')
```

In [3]:

```
train_img = []
gnd = []
files = glob.glob('CAB420_Assessment_1C_Data/Data/Q2/Q2/Train_Data/Originals/*.png')
for myfile in files:
    im = keras.preprocessing.image.load_img(myfile,target_size=(143,89))
    image = keras.preprocessing.image.img_to_array(im)
    gnd.append(myfile[58:])
    image = image[0:143,0:89]
    train_img.append(image)


train_img = np.array(train_img)
train_img = train_img.astype('float32') / 255.
```

In [4]:

```
print(train_img[0])
```

```
[[[0.27450982 0.2784314  0.24705882]
  [0.27450982 0.2784314  0.24705882]
  [0.27450982 0.2784314  0.24705882]
  ...
  [0.3372549  0.3372549  0.2901961 ]
  [0.3372549  0.3372549  0.2901961 ]
  [0.34117648 0.34117648 0.29411766]]

 [[0.2784314  0.28235295 0.2509804 ]
  [0.2784314  0.28235295 0.2509804 ]
  [0.2784314  0.28235295 0.2509804 ]
  ...
  [0.34509805 0.34509805 0.29803923]
  [0.34117648 0.34117648 0.29411766]
  [0.34901962 0.34901962 0.3019608 ]]

 [[0.28235295 0.28627452 0.25490198]
  [0.28235295 0.28627452 0.25490198]
  [0.28235295 0.28627452 0.25490198]
  ...
  [0.34509805 0.34509805 0.29803923]
  [0.34509805 0.34509805 0.29803923]
  [0.3529412  0.3529412  0.30588236]]

 ...

 [[0.30980393 0.30980393 0.2627451 ]
  [0.29803923 0.29803923 0.2509804 ]
  [0.30588236 0.30588236 0.25882354]
  ...
  [0.45882353 0.4509804  0.4       ]
  [0.45882353 0.4509804  0.4       ]
  [0.45490196 0.44705883 0.39607844]]

 [[0.30980393 0.30980393 0.2627451 ]
  [0.29803923 0.29803923 0.2509804 ]
  [0.30588236 0.30588236 0.25882354]
  ...
  [0.45490196 0.44705883 0.39607844]
  [0.4509804  0.44313726 0.39215687]
  [0.45490196 0.44705883 0.39607844]]

 [[0.30980393 0.30980393 0.2627451 ]
  [0.29803923 0.29803923 0.2509804 ]
  [0.3019608  0.3019608  0.25490198]
  ...
  [0.44705883 0.4509804  0.39607844]
  [0.44313726 0.44705883 0.39215687]
  [0.4509804  0.45490196 0.4       ]]]
```

In [5]:

```
test_img = []
test_gnd = []
files = glob.glob('CAB420_Assessment_1C_Data/Data/Q2/Q2/Test_Data/Originals/*.png')
for myfile in files:
    im = keras.preprocessing.image.load_img(myfile,target_size=(143,89))
    image = keras.preprocessing.image.img_to_array(im)
    test_gnd.append(myfile[58:])
    test_img.append(image)


test_img = np.array(test_img)
test_img = test_img.astype('float32') / 255.
```

In [6]:

```
train = train.drop(columns=['torcol2','torcol3','tortex','torcol3','legcol2','legcol3',
'legtex','pose'])


train_gender = train.iloc[:,1]
train_gender = np.asarray(train_gender)
train_tortyp = train.iloc[:,2]
train_tortyp = np.asarray(train_tortyp)
train_torcol = train.iloc[:,3]
train_torcol = np.asarray(train_torcol)
train_legtyp = train.iloc[:,4]
train_legtyp = np.asarray(train_legtyp)
train_legcol = train.iloc[:,5]
train_legcol = np.asarray(train_legcol)
train_luggage = train.iloc[:,6]
train_luggage = np.asarray(train_luggage)
```

In [7]:

```
test = test.drop(columns=['torcol2','torcol3','tortex','torcol3','legcol2','legcol3','l
egtex','pose'])

test_gender = test.iloc[:,1]
test_gender = np.asarray(test_gender)
test_tortyp = test.iloc[:,2]
test_tortyp = np.asarray(test_tortyp)
test_torcol = test.iloc[:,3]
test_torcol = np.asarray(test_torcol)
test_legtyp = test.iloc[:,4]
test_legtyp = np.asarray(test_legtyp)
test_legcol = test.iloc[:,5]
test_legcol = np.asarray(test_legcol)
test_luggage = test.iloc[:,6]
test_luggage = np.asarray(test_luggage)
```

In [8]:

```
#fig = plt.figure(figsize=[20, 20])
#for i in range(100):
#    ax = fig.add_subplot(10, 10, i + 1)
#    ax.imshow(train_img[i])
```

In [9]:

```
#,tortyp,torcol,legtyp,legcol,luggage]
#, train_tortyp,train_torcol,train_legtyp,train_legcol,train_luggage]
#, test_tortyp, test_torcol, test_legtyp, test_legcol, test_luggage]
```

In [10]:

```
inputs = keras.Input(shape=(143, 89, 3, ), name='img')
#x = inputs
x = layers.Conv2D(filters=8, kernel_size=(3,3), activation='relu')(inputs)
x = layers.MaxPool2D(pool_size=(2, 2))(x)
x = layers.Conv2D(filters=16, kernel_size=(3,3), activation='relu')(x)
x = layers.MaxPool2D(pool_size=(2, 2))(x)
x = layers.Conv2D(filters=32, kernel_size=(3,3), activation='relu')(x)
#x = layers.Flatten()(x)
x = layers.Conv2D(filters=64, kernel_size=(3,3), activation='relu')(x)
x = layers.Conv2D(filters=96, kernel_size=(3,3), activation='relu')(x)

x1 = layers.Dense(64, activation='relu')(x)
x1 = layers.MaxPool2D(pool_size=(2, 2))(x)
x1 = layers.Conv2D(filters=16, kernel_size=(3,3), activation='relu', padding='same')(x1
)
x1 = layers.MaxPool2D(pool_size=(2, 2))(x1)
x1 = layers.Conv2D(filters=32, kernel_size=(3,3), activation='relu', padding='same')(x1
)
x1 = layers.Flatten()(x1)
x1 = layers.Dense(64, activation='relu')(x1)
gender = layers.Dense(3, name='gender_out')(x1)

x2 = layers.Dense(64, activation='relu')(x)
x2 = layers.MaxPool2D(pool_size=(2, 2))(x)
x2 = layers.Conv2D(filters=16, kernel_size=(3,3), activation='relu', padding='same')(x2
)
x2 = layers.MaxPool2D(pool_size=(2, 2))(x2)
x2 = layers.Conv2D(filters=32, kernel_size=(3,3), activation='relu', padding='same')(x2
)
x2 = layers.Flatten()(x2)
x2 = layers.Dense(64, activation='relu')(x2)
tortyp = layers.Dense(3, name='tortyp_out')(x2)

x3 = layers.Dense(64, activation='relu')(x)
x3 = layers.MaxPool2D(pool_size=(2, 2))(x)
x3 = layers.Conv2D(filters=16, kernel_size=(3,3), activation='relu', padding='same')(x3
)
x3 = layers.MaxPool2D(pool_size=(2, 2))(x3)
x3 = layers.Conv2D(filters=32, kernel_size=(3,3), activation='relu', padding='same')(x3
)
x3 = layers.Flatten()(x3)
x3 = layers.Dense(64, activation='relu')(x3)
torcol = layers.Dense(11, name='torcol_out')(x3)

x4 = layers.Dense(64, activation='relu')(x)
x4 = layers.MaxPool2D(pool_size=(2, 2))(x)
x4 = layers.Conv2D(filters=16, kernel_size=(3,3), activation='relu', padding='same')(x4
)
x4 = layers.MaxPool2D(pool_size=(2, 2))(x4)
x4 = layers.Conv2D(filters=32, kernel_size=(3,3), activation='relu', padding='same')(x4
)
x4 = layers.Flatten()(x4)
x4 = layers.Dense(64, activation='relu')(x4)
legtyp = layers.Dense(3, name='legtyp_out')(x4)

x5 = layers.Dense(64, activation='relu')(x)
x5 = layers.MaxPool2D(pool_size=(2, 2))(x)
x5 = layers.Conv2D(filters=16, kernel_size=(3,3), activation='relu', padding='same')(x5
)
```

```
x5 = layers.MaxPool2D(pool_size=(2, 2))(x5)
x5 = layers.Conv2D(filters=32, kernel_size=(3,3), activation='relu', padding='same')(x5
)
x5 = layers.Flatten()(x5)
x5 = layers.Dense(64, activation='relu')(x5)
legcol = layers.Dense(11, name='legocol_out')(x5)

x6 = layers.Dense(64, activation='relu')(x)
x6 = layers.MaxPool2D(pool_size=(2, 2))(x)
x6 = layers.Conv2D(filters=16, kernel_size=(3,3), activation='relu', padding='same')(x6
)
x6 = layers.MaxPool2D(pool_size=(2, 2))(x6)
x6 = layers.Conv2D(filters=32, kernel_size=(3,3), activation='relu', padding='same')(x6
)
x6 = layers.Flatten()(x6)
x6 = layers.Dense(64, activation='relu')(x6)
luggage = layers.Dense(3, name='luggage_out')(x6)

model_cnn = keras.Model(inputs=inputs, outputs=[gender, tortyp, torcol, legtyp, legcol,
luggage], name='A1CQ2')


# inputs = keras.Input(shape=(50, 50, 3, ), name='img')
# x = layers.Conv2D(filters=8, kernel_size=(3,3), activation='relu', padding='same')(in
puts)
# x = layers.MaxPool2D(pool_size=(2, 2))(x)
# x = layers.Conv2D(filters=16, kernel_size=(3,3), activation='relu', padding='same')
(x)
# x = layers.MaxPool2D(pool_size=(2, 2))(x)
# x = layers.Conv2D(filters=32, kernel_size=(3,3), activation='relu', padding='same')
(x)
# x = layers.Flatten()(x)
# x = layers.Dense(64, activation='relu')(x)
# outputs = layers.Dense(11, activation='softmax')(x)

# model_cnn = keras.Model(inputs=inputs, outputs=outputs, name='SVHN_CNN_Model')


model_cnn.summary()
```

```
Model: "A1CQ2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
img (InputLayer)             [(None, 143, 89, 3)]      0         
_________________________________________________________________
conv2d (Conv2D)              (None, 141, 87, 8)        224       
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 70, 43, 8)         0         
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 68, 41, 16)        1168      
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 34, 20, 16)        0         
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 18, 32)        4640      
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 30, 16, 64)        18496     
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 28, 14, 96)        55392     
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 14, 7, 96)         0         
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 14, 7, 96)         0         
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 14, 7, 96)         0         
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 14, 7, 96)         0         
_________________________________________________________________
max_pooling2d_10 (MaxPooling (None, 14, 7, 96)         0         
_________________________________________________________________
max_pooling2d_12 (MaxPooling (None, 14, 7, 96)         0         
_________________________________________________________________
```

```
conv2d_5 (Conv2D)               (None, 14, 7, 16)    13840       max_pooli
ng2d_2[0][0]
__________________________________________________________________________
_____
conv2d_7 (Conv2D)               (None, 14, 7, 16)    13840       max_pooli
ng2d_4[0][0]
__________________________________________________________________________
_____
conv2d_9 (Conv2D)               (None, 14, 7, 16)    13840       max_pooli
ng2d_6[0][0]
__________________________________________________________________________
_____
conv2d_11 (Conv2D)              (None, 14, 7, 16)    13840       max_pooli
ng2d_8[0][0]
__________________________________________________________________________
_____
conv2d_13 (Conv2D)              (None, 14, 7, 16)    13840       max_pooli
ng2d_10[0][0]
__________________________________________________________________________
_____
conv2d_15 (Conv2D)              (None, 14, 7, 16)    13840       max_pooli
ng2d_12[0][0]
__________________________________________________________________________
_____
max_pooling2d_3 (MaxPooling2D)  (None, 7, 3, 16)     0           conv2d_5
[0][0]
__________________________________________________________________________
_____
max_pooling2d_5 (MaxPooling2D)  (None, 7, 3, 16)     0           conv2d_7
[0][0]
__________________________________________________________________________
_____
max_pooling2d_7 (MaxPooling2D)  (None, 7, 3, 16)     0           conv2d_9
[0][0]
__________________________________________________________________________
_____
max_pooling2d_9 (MaxPooling2D)  (None, 7, 3, 16)     0           conv2d_11
[0][0]
__________________________________________________________________________
_____
max_pooling2d_11 (MaxPooling2D) (None, 7, 3, 16)     0           conv2d_13
[0][0]
__________________________________________________________________________
_____
max_pooling2d_13 (MaxPooling2D) (None, 7, 3, 16)     0           conv2d_15
[0][0]
__________________________________________________________________________
_____
conv2d_6 (Conv2D)               (None, 7, 3, 32)     4640        max_pooli
ng2d_3[0][0]
__________________________________________________________________________
_____
conv2d_8 (Conv2D)               (None, 7, 3, 32)     4640        max_pooli
ng2d_5[0][0]
__________________________________________________________________________
_____
conv2d_10 (Conv2D)              (None, 7, 3, 32)     4640        max_pooli
ng2d_7[0][0]
__________________________________________________________________________
_____
conv2d_12 (Conv2D)              (None, 7, 3, 32)     4640        max_pooli
```

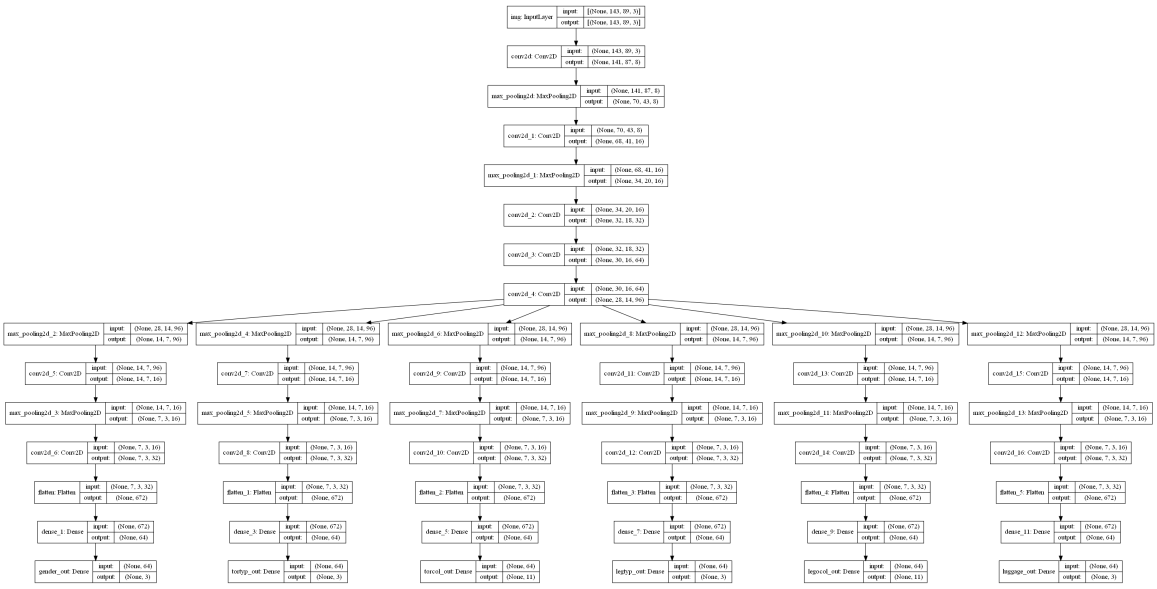
```
ng2d_9[0][0]
__________________________________________________________________________________________________
__________________________
conv2d_14 (Conv2D)              (None, 7, 3, 32)     4640        max_pooli
ng2d_11[0][0]
__________________________________________________________________________________________________
__________________________
conv2d_16 (Conv2D)              (None, 7, 3, 32)     4640        max_pooli
ng2d_13[0][0]
__________________________________________________________________________________________________
__________________________
flatten (Flatten)               (None, 672)          0           conv2d_6
[0][0]
__________________________________________________________________________________________________
__________________________
flatten_1 (Flatten)             (None, 672)          0           conv2d_8
[0][0]
__________________________________________________________________________________________________
__________________________
flatten_2 (Flatten)             (None, 672)          0           conv2d_10
[0][0]
__________________________________________________________________________________________________
__________________________
flatten_3 (Flatten)             (None, 672)          0           conv2d_12
[0][0]
__________________________________________________________________________________________________
__________________________
flatten_4 (Flatten)             (None, 672)          0           conv2d_14
[0][0]
__________________________________________________________________________________________________
__________________________
flatten_5 (Flatten)             (None, 672)          0           conv2d_16
[0][0]
__________________________________________________________________________________________________
__________________________
dense_1 (Dense)                 (None, 64)           43072       flatten
[0][0]
__________________________________________________________________________________________________
__________________________
dense_3 (Dense)                 (None, 64)           43072       flatten_1
[0][0]
__________________________________________________________________________________________________
__________________________
dense_5 (Dense)                 (None, 64)           43072       flatten_2
[0][0]
__________________________________________________________________________________________________
__________________________
dense_7 (Dense)                 (None, 64)           43072       flatten_3
[0][0]
__________________________________________________________________________________________________
__________________________
dense_9 (Dense)                 (None, 64)           43072       flatten_4
[0][0]
__________________________________________________________________________________________________
__________________________
dense_11 (Dense)                (None, 64)           43072       flatten_5
[0][0]
__________________________________________________________________________________________________
__________________________
gender_out (Dense)              (None, 3)            195         dense_1
[0][0]
```

```
__________________________________________________________________________________
_____________________
tortyp_out (Dense)              (None, 3)            195         dense_3
[0][0]
__________________________________________________________________________________
_____________________
torcol_out (Dense)              (None, 11)           715         dense_5
[0][0]
__________________________________________________________________________________
_____________________
legtyp_out (Dense)              (None, 3)            195         dense_7
[0][0]
__________________________________________________________________________________
_____________________
legocol_out (Dense)             (None, 11)           715         dense_9
[0][0]
__________________________________________________________________________________
_____________________
luggage_out (Dense)             (None, 3)            195         dense_11
[0][0]
==================================================================================
========================
Total params: 451,442
Trainable params: 451,442
Non-trainable params: 0
__________________________________________________________________________________
_____________________
```

In [11]:

```
plot_model(model_cnn, to_file='test_keras_plot_model.png', show_shapes=True)
IPython.display.Image('test_keras_plot_model.png')
```

Out[11]:



In [12]:

```
model_cnn.compile(loss=['mean_squared_error', keras.losses.SparseCategoricalCrossentrop
y(from_logits=True)],
              optimizer=keras.optimizers.RMSprop(),metrics=['accuracy'])
```

In [13]:

```
history = model_cnn.fit(train_img, [train_gender, train_tortyp, train_torcol, train_leg
typ, train_legcol, train_luggage],
                    batch_size=64,
                    epochs=20,
                    validation_data=(test_img, [test_gender, test_tortyp, test_torcol,
test_legtyp, test_legcol, test_luggage]),verbose=True)
```

```
Epoch 1/20
WARNING:tensorflow:Gradients do not exist for variables ['conv2d_9/kernel:
0', 'conv2d_9/bias:0', 'conv2d_11/kernel:0', 'conv2d_11/bias:0', 'conv2d_1
3/kernel:0', 'conv2d_13/bias:0', 'conv2d_15/kernel:0', 'conv2d_15/bias:0',
'conv2d_10/kernel:0', 'conv2d_10/bias:0', 'conv2d_12/kernel:0', 'conv2d_1
2/bias:0', 'conv2d_14/kernel:0', 'conv2d_14/bias:0', 'conv2d_16/kernel:0',
'conv2d_16/bias:0', 'dense_5/kernel:0', 'dense_5/bias:0', 'dense_7/kernel:
0', 'dense_7/bias:0', 'dense_9/kernel:0', 'dense_9/bias:0', 'dense_11/kern
el:0', 'dense_11/bias:0', 'torcol_out/kernel:0', 'torcol_out/bias:0', 'leg
typ_out/kernel:0', 'legtyp_out/bias:0', 'legocol_out/kernel:0', 'legocol_o
ut/bias:0', 'luggage_out/kernel:0', 'luggage_out/bias:0'] when minimizing
the loss.
WARNING:tensorflow:Gradients do not exist for variables ['conv2d_9/kernel:
0', 'conv2d_9/bias:0', 'conv2d_11/kernel:0', 'conv2d_11/bias:0', 'conv2d_1
3/kernel:0', 'conv2d_13/bias:0', 'conv2d_15/kernel:0', 'conv2d_15/bias:0',
'conv2d_10/kernel:0', 'conv2d_10/bias:0', 'conv2d_12/kernel:0', 'conv2d_1
2/bias:0', 'conv2d_14/kernel:0', 'conv2d_14/bias:0', 'conv2d_16/kernel:0',
'conv2d_16/bias:0', 'dense_5/kernel:0', 'dense_5/bias:0', 'dense_7/kernel:
0', 'dense_7/bias:0', 'dense_9/kernel:0', 'dense_9/bias:0', 'dense_11/kern
el:0', 'dense_11/bias:0', 'torcol_out/kernel:0', 'torcol_out/bias:0', 'leg
typ_out/kernel:0', 'legtyp_out/bias:0', 'legocol_out/kernel:0', 'legocol_o
ut/bias:0', 'luggage_out/kernel:0', 'luggage_out/bias:0'] when minimizing
the loss.
9/9 [==============================] - 17s 359ms/step - loss: 1.2433 - gen
der_out_loss: 0.3611 - tortyp_out_loss: 0.8822 - gender_out_accuracy: 0.35
23 - tortyp_out_accuracy: 0.5454 - torcol_out_accuracy: 0.0349 - legtyp_ou
t_accuracy: 0.3078 - legocol_out_accuracy: 0.2551 - luggage_out_accuracy:
0.0305 - val_loss: 2.9064 - val_gender_out_loss: 2.0362 - val_tortyp_out_l
oss: 0.8701 - val_gender_out_accuracy: 0.0561 - val_tortyp_out_accuracy:
0.3827 - val_torcol_out_accuracy: 0.0408 - val_legtyp_out_accuracy: 0.4184
- val_legocol_out_accuracy: 0.0663 - val_luggage_out_accuracy: 0.0000e+00
Epoch 2/20
9/9 [==============================] - 1s 137ms/step - loss: 1.5336 - gend
er_out_loss: 0.7965 - tortyp_out_loss: 0.7371 - gender_out_accuracy: 0.326
3 - tortyp_out_accuracy: 0.5276 - torcol_out_accuracy: 0.0608 - legtyp_out
_accuracy: 0.3192 - legocol_out_accuracy: 0.2576 - luggage_out_accuracy:
0.0089 - val_loss: 1.0938 - val_gender_out_loss: 0.4206 - val_tortyp_out_l
oss: 0.6732 - val_gender_out_accuracy: 0.0510 - val_tortyp_out_accuracy:
0.6173 - val_torcol_out_accuracy: 0.0357 - val_legtyp_out_accuracy: 0.4184
- val_legocol_out_accuracy: 0.1224 - val_luggage_out_accuracy: 0.0255
Epoch 3/20
9/9 [==============================] - 1s 95ms/step - loss: 0.9598 - gende
r_out_loss: 0.2794 - tortyp_out_loss: 0.6804 - gender_out_accuracy: 0.3512
- tortyp_out_accuracy: 0.5816 - torcol_out_accuracy: 0.0429 - legtyp_out_a
ccuracy: 0.3218 - legocol_out_accuracy: 0.2087 - luggage_out_accuracy: 0.0
306 - val_loss: 1.1419 - val_gender_out_loss: 0.3402 - val_tortyp_out_los
s: 0.8017 - val_gender_out_accuracy: 0.3571 - val_tortyp_out_accuracy: 0.3
827 - val_torcol_out_accuracy: 0.0255 - val_legtyp_out_accuracy: 0.4184 -
val_legocol_out_accuracy: 0.1071 - val_luggage_out_accuracy: 0.0153
Epoch 4/20
9/9 [==============================] - 1s 98ms/step - loss: 0.9706 - gende
r_out_loss: 0.2680 - tortyp_out_loss: 0.7025 - gender_out_accuracy: 0.4075
- tortyp_out_accuracy: 0.5424 - torcol_out_accuracy: 0.0525 - legtyp_out_a
ccuracy: 0.3147 - legocol_out_accuracy: 0.2452 - luggage_out_accuracy: 0.0
343 - val_loss: 1.1812 - val_gender_out_loss: 0.4016 - val_tortyp_out_los
s: 0.7796 - val_gender_out_accuracy: 0.3571 - val_tortyp_out_accuracy: 0.3
827 - val_torcol_out_accuracy: 0.0459 - val_legtyp_out_accuracy: 0.4184 -
val_legocol_out_accuracy: 0.1786 - val_luggage_out_accuracy: 0.0000e+00
Epoch 5/20
9/9 [==============================] - 1s 98ms/step - loss: 1.0153 - gende
r_out_loss: 0.2984 - tortyp_out_loss: 0.7170 - gender_out_accuracy: 0.3933
```

```
- tortyp_out_accuracy: 0.4797 - torcol_out_accuracy: 0.0532 - legtyp_out_a
ccuracy: 0.3368 - legocol_out_accuracy: 0.2536 - luggage_out_accuracy: 0.0
069 - val_loss: 1.3014 - val_gender_out_loss: 0.3573 - val_tortyp_out_los
s: 0.9441 - val_gender_out_accuracy: 0.2653 - val_tortyp_out_accuracy: 0.3
827 - val_torcol_out_accuracy: 0.0561 - val_legtyp_out_accuracy: 0.4184 -
val_legocol_out_accuracy: 0.0714 - val_luggage_out_accuracy: 0.0000e+00
Epoch 6/20
9/9 [==============================] - 1s 117ms/step - loss: 0.9882 - gend
er_out_loss: 0.2726 - tortyp_out_loss: 0.7156 - gender_out_accuracy: 0.354
8 - tortyp_out_accuracy: 0.5346 - torcol_out_accuracy: 0.0482 - legtyp_out
_accuracy: 0.2918 - legocol_out_accuracy: 0.2041 - luggage_out_accuracy:
0.0014 - val_loss: 1.0569 - val_gender_out_loss: 0.3404 - val_tortyp_out_l
oss: 0.7165 - val_gender_out_accuracy: 0.3980 - val_tortyp_out_accuracy:
0.6173 - val_torcol_out_accuracy: 0.0561 - val_legtyp_out_accuracy: 0.4184
- val_legocol_out_accuracy: 0.0765 - val_luggage_out_accuracy: 0.0153
Epoch 7/20
9/9 [==============================] - 1s 87ms/step - loss: 0.9546 - gende
r_out_loss: 0.2754 - tortyp_out_loss: 0.6792 - gender_out_accuracy: 0.3803
- tortyp_out_accuracy: 0.6096 - torcol_out_accuracy: 0.0425 - legtyp_out_a
ccuracy: 0.3263 - legocol_out_accuracy: 0.1749 - luggage_out_accuracy: 0.0
317 - val_loss: 1.2165 - val_gender_out_loss: 0.4141 - val_tortyp_out_los
s: 0.8024 - val_gender_out_accuracy: 0.4898 - val_tortyp_out_accuracy: 0.3
827 - val_torcol_out_accuracy: 0.0459 - val_legtyp_out_accuracy: 0.4184 -
val_legocol_out_accuracy: 0.0765 - val_luggage_out_accuracy: 0.0153
Epoch 8/20
9/9 [==============================] - 1s 93ms/step - loss: 0.9933 - gende
r_out_loss: 0.2949 - tortyp_out_loss: 0.6984 - gender_out_accuracy: 0.5067
- tortyp_out_accuracy: 0.5501 - torcol_out_accuracy: 0.0389 - legtyp_out_a
ccuracy: 0.3238 - legocol_out_accuracy: 0.2323 - luggage_out_accuracy: 0.0
113 - val_loss: 1.1749 - val_gender_out_loss: 0.3965 - val_tortyp_out_los
s: 0.7783 - val_gender_out_accuracy: 0.4490 - val_tortyp_out_accuracy: 0.3
827 - val_torcol_out_accuracy: 0.0561 - val_legtyp_out_accuracy: 0.4184 -
val_legocol_out_accuracy: 0.1582 - val_luggage_out_accuracy: 0.0204
Epoch 9/20
9/9 [==============================] - 1s 107ms/step - loss: 0.9902 - gend
er_out_loss: 0.2750 - tortyp_out_loss: 0.7152 - gender_out_accuracy: 0.344
8 - tortyp_out_accuracy: 0.5038 - torcol_out_accuracy: 0.0565 - legtyp_out
_accuracy: 0.3070 - legocol_out_accuracy: 0.1359 - luggage_out_accuracy:
0.0283 - val_loss: 1.0212 - val_gender_out_loss: 0.3270 - val_tortyp_out_l
oss: 0.6942 - val_gender_out_accuracy: 0.1888 - val_tortyp_out_accuracy:
0.6173 - val_torcol_out_accuracy: 0.0510 - val_legtyp_out_accuracy: 0.4184
- val_legocol_out_accuracy: 0.0357 - val_luggage_out_accuracy: 0.0051
Epoch 10/20
9/9 [==============================] - 1s 106ms/step - loss: 0.9345 - gend
er_out_loss: 0.2529 - tortyp_out_loss: 0.6816 - gender_out_accuracy: 0.273
8 - tortyp_out_accuracy: 0.6042 - torcol_out_accuracy: 0.0586 - legtyp_out
_accuracy: 0.3392 - legocol_out_accuracy: 0.1430 - luggage_out_accuracy:
0.0527 - val_loss: 1.0668 - val_gender_out_loss: 0.3257 - val_tortyp_out_l
oss: 0.7410 - val_gender_out_accuracy: 0.3827 - val_tortyp_out_accuracy:
0.3929 - val_torcol_out_accuracy: 0.0459 - val_legtyp_out_accuracy: 0.4184
- val_legocol_out_accuracy: 0.1224 - val_luggage_out_accuracy: 0.0102
Epoch 11/20
9/9 [==============================] - 1s 92ms/step - loss: 0.9281 - gende
r_out_loss: 0.2594 - tortyp_out_loss: 0.6687 - gender_out_accuracy: 0.3893
- tortyp_out_accuracy: 0.6158 - torcol_out_accuracy: 0.0601 - legtyp_out_a
ccuracy: 0.3230 - legocol_out_accuracy: 0.1909 - luggage_out_accuracy: 0.0
406 - val_loss: 1.0123 - val_gender_out_loss: 0.3330 - val_tortyp_out_los
s: 0.6793 - val_gender_out_accuracy: 0.1378 - val_tortyp_out_accuracy: 0.6
173 - val_torcol_out_accuracy: 0.0612 - val_legtyp_out_accuracy: 0.4184 -
val_legocol_out_accuracy: 0.2143 - val_luggage_out_accuracy: 0.0204
Epoch 12/20
```

```
9/9 [==============================] - 1s 100ms/step - loss: 0.8994 - gend
er_out_loss: 0.2511 - tortyp_out_loss: 0.6483 - gender_out_accuracy: 0.334
9 - tortyp_out_accuracy: 0.6024 - torcol_out_accuracy: 0.0315 - legtyp_out
_accuracy: 0.2930 - legocol_out_accuracy: 0.3054 - luggage_out_accuracy:
0.0643 - val_loss: 1.0024 - val_gender_out_loss: 0.3262 - val_tortyp_out_l
oss: 0.6762 - val_gender_out_accuracy: 0.4388 - val_tortyp_out_accuracy:
0.6173 - val_torcol_out_accuracy: 0.0765 - val_legtyp_out_accuracy: 0.4184
- val_legocol_out_accuracy: 0.2551 - val_luggage_out_accuracy: 0.0306
Epoch 13/20
9/9 [==============================] - 1s 96ms/step - loss: 0.8940 - gende
r_out_loss: 0.2487 - tortyp_out_loss: 0.6453 - gender_out_accuracy: 0.3913
- tortyp_out_accuracy: 0.6119 - torcol_out_accuracy: 0.0565 - legtyp_out_a
ccuracy: 0.2898 - legocol_out_accuracy: 0.3197 - luggage_out_accuracy: 0.0
915 - val_loss: 1.0069 - val_gender_out_loss: 0.3287 - val_tortyp_out_los
s: 0.6781 - val_gender_out_accuracy: 0.0510 - val_tortyp_out_accuracy: 0.5
867 - val_torcol_out_accuracy: 0.0561 - val_legtyp_out_accuracy: 0.4184 -
val_legocol_out_accuracy: 0.2347 - val_luggage_out_accuracy: 0.0408
Epoch 14/20
9/9 [==============================] - 1s 103ms/step - loss: 0.9010 - gend
er_out_loss: 0.2603 - tortyp_out_loss: 0.6407 - gender_out_accuracy: 0.213
2 - tortyp_out_accuracy: 0.6374 - torcol_out_accuracy: 0.0600 - legtyp_out
_accuracy: 0.3091 - legocol_out_accuracy: 0.2994 - luggage_out_accuracy:
0.0994 - val_loss: 1.5336 - val_gender_out_loss: 0.3452 - val_tortyp_out_l
oss: 1.1884 - val_gender_out_accuracy: 0.4796 - val_tortyp_out_accuracy:
0.3827 - val_torcol_out_accuracy: 0.0663 - val_legtyp_out_accuracy: 0.4184
- val_legocol_out_accuracy: 0.2806 - val_luggage_out_accuracy: 0.0510
Epoch 15/20
9/9 [==============================] - 1s 92ms/step - loss: 0.9921 - gende
r_out_loss: 0.2689 - tortyp_out_loss: 0.7231 - gender_out_accuracy: 0.3549
- tortyp_out_accuracy: 0.5787 - torcol_out_accuracy: 0.0576 - legtyp_out_a
ccuracy: 0.3045 - legocol_out_accuracy: 0.3383 - luggage_out_accuracy: 0.1
111 - val_loss: 1.1502 - val_gender_out_loss: 0.3382 - val_tortyp_out_los
s: 0.8120 - val_gender_out_accuracy: 0.3214 - val_tortyp_out_accuracy: 0.6
122 - val_torcol_out_accuracy: 0.0765 - val_legtyp_out_accuracy: 0.4184 -
val_legocol_out_accuracy: 0.2704 - val_luggage_out_accuracy: 0.1020
Epoch 16/20
9/9 [==============================] - 1s 100ms/step - loss: 0.9757 - gend
er_out_loss: 0.2543 - tortyp_out_loss: 0.7214 - gender_out_accuracy: 0.256
0 - tortyp_out_accuracy: 0.6254 - torcol_out_accuracy: 0.0453 - legtyp_out
_accuracy: 0.3029 - legocol_out_accuracy: 0.3446 - luggage_out_accuracy:
0.1199 - val_loss: 1.0714 - val_gender_out_loss: 0.3348 - val_tortyp_out_l
oss: 0.7365 - val_gender_out_accuracy: 0.2602 - val_tortyp_out_accuracy:
0.4694 - val_torcol_out_accuracy: 0.0816 - val_legtyp_out_accuracy: 0.4184
- val_legocol_out_accuracy: 0.3061 - val_luggage_out_accuracy: 0.1122
Epoch 17/20
9/9 [==============================] - 1s 100ms/step - loss: 0.8332 - gend
er_out_loss: 0.2504 - tortyp_out_loss: 0.5828 - gender_out_accuracy: 0.343
3 - tortyp_out_accuracy: 0.7126 - torcol_out_accuracy: 0.0472 - legtyp_out
_accuracy: 0.3305 - legocol_out_accuracy: 0.3275 - luggage_out_accuracy:
0.1893 - val_loss: 1.2140 - val_gender_out_loss: 0.3420 - val_tortyp_out_l
oss: 0.8720 - val_gender_out_accuracy: 0.2551 - val_tortyp_out_accuracy:
0.6071 - val_torcol_out_accuracy: 0.0612 - val_legtyp_out_accuracy: 0.4184
- val_legocol_out_accuracy: 0.3061 - val_luggage_out_accuracy: 0.0714
Epoch 18/20
9/9 [==============================] - 1s 103ms/step - loss: 0.9383 - gend
er_out_loss: 0.2382 - tortyp_out_loss: 0.7001 - gender_out_accuracy: 0.363
8 - tortyp_out_accuracy: 0.6955 - torcol_out_accuracy: 0.0454 - legtyp_out
_accuracy: 0.3191 - legocol_out_accuracy: 0.2904 - luggage_out_accuracy:
0.1434 - val_loss: 1.1138 - val_gender_out_loss: 0.3421 - val_tortyp_out_l
oss: 0.7717 - val_gender_out_accuracy: 0.4694 - val_tortyp_out_accuracy:
0.4541 - val_torcol_out_accuracy: 0.0765 - val_legtyp_out_accuracy: 0.4184
```

```
- val_legocol_out_accuracy: 0.2857 - val_luggage_out_accuracy: 0.0714
Epoch 19/20
9/9 [==============================] - 1s 102ms/step - loss: 0.7948 - gend
er_out_loss: 0.2368 - tortyp_out_loss: 0.5580 - gender_out_accuracy: 0.411
9 - tortyp_out_accuracy: 0.7083 - torcol_out_accuracy: 0.0431 - legtyp_out
_accuracy: 0.3022 - legocol_out_accuracy: 0.3051 - luggage_out_accuracy:
0.1374 - val_loss: 1.1429 - val_gender_out_loss: 0.3618 - val_tortyp_out_l
oss: 0.7810 - val_gender_out_accuracy: 0.4541 - val_tortyp_out_accuracy:
0.6071 - val_torcol_out_accuracy: 0.0714 - val_legtyp_out_accuracy: 0.4184
- val_legocol_out_accuracy: 0.2959 - val_luggage_out_accuracy: 0.0918
Epoch 20/20
9/9 [==============================] - 1s 97ms/step - loss: 0.8428 - gende
r_out_loss: 0.2505 - tortyp_out_loss: 0.5923 - gender_out_accuracy: 0.3933
- tortyp_out_accuracy: 0.7135 - torcol_out_accuracy: 0.0534 - legtyp_out_a
ccuracy: 0.3281 - legocol_out_accuracy: 0.3025 - luggage_out_accuracy: 0.1
260 - val_loss: 1.1572 - val_gender_out_loss: 0.3613 - val_tortyp_out_los
s: 0.7960 - val_gender_out_accuracy: 0.3061 - val_tortyp_out_accuracy: 0.6
020 - val_torcol_out_accuracy: 0.0408 - val_legtyp_out_accuracy: 0.4184 -
val_legocol_out_accuracy: 0.3010 - val_luggage_out_accuracy: 0.0918
```

In [ ]: