3dcarprojection

December 7, 2019

Yesterday, I read this recent article on medium about facial keypoint detection. The article suggests that deep learning methods can easily be used to perform this task. It ends by suggesting that everyone should try it, since the data needed and the toolkits are all open source. This article is my attempt, since I've been interested in face detection for a long time and written about it before.

This is the outline of what we'll try:

- loading the data
- analyzing the data
- building a Keras model
- checking the results
- applying the method to a fun problem

1 Loading the data

The data we will use comes from a Kaggle challenge called *Facial Keypoints Detection*. I've downloaded the *.csv* file and put it in a *data/* directory. Let's use pandas to read it.

```
In [2]: import pandas as pd

In [3]: df = pd.read_csv('vehiclereid_baseline/test1.txt', sep=",", header=None)

In [4]: df.head()

Out[4]:                                          0   1   2   3   4    5   6    7  \
        0  VeRi/image_train/0181_c001_00034295_0.jpg  -1  -1  -1  -1  367  89  190
        1  VeRi/image_train/0181_c001_00034315_0.jpg  -1  -1  -1  -1  231  58  153
        2  VeRi/image_train/0181_c001_00034340_0.jpg  -1  -1  -1  -1  130  56  114
        3  VeRi/image_train/0181_c012_00034760_0.jpg  -1  -1  -1  -1  213  71  163
        4  VeRi/image_train/0181_c012_00034765_0.jpg  -1  -1  -1  -1  192  66  151

             8   9  ...  32  33   34   35   36  37   38  39   40  41
        0  205  -1  ...  77  14  148   91  180  44  160  46  176   7
        1  132  -1  ...  32  14   94  108  108  55  100  58  114   7
        2   92  -1  ...  19  18   64   96   69  55   62  54   72   7
        3  141  -1  ...  44  22  100  123  113  67  106  70  123   7
        4  124  -1  ...  36  25   89  116   97  68   93  73  108   7

        [5 rows x 42 columns]
```

```
In [5]: df.shape
```

```
Out[5]: (499, 42)
```

2 Analyzing the data

The `Image` column contains the face data for which the 30 first columns represent the keypoint data (15 x-coordinates and 15 y-coordinates). Let's try to get a feel for the data. First, let's display some faces.

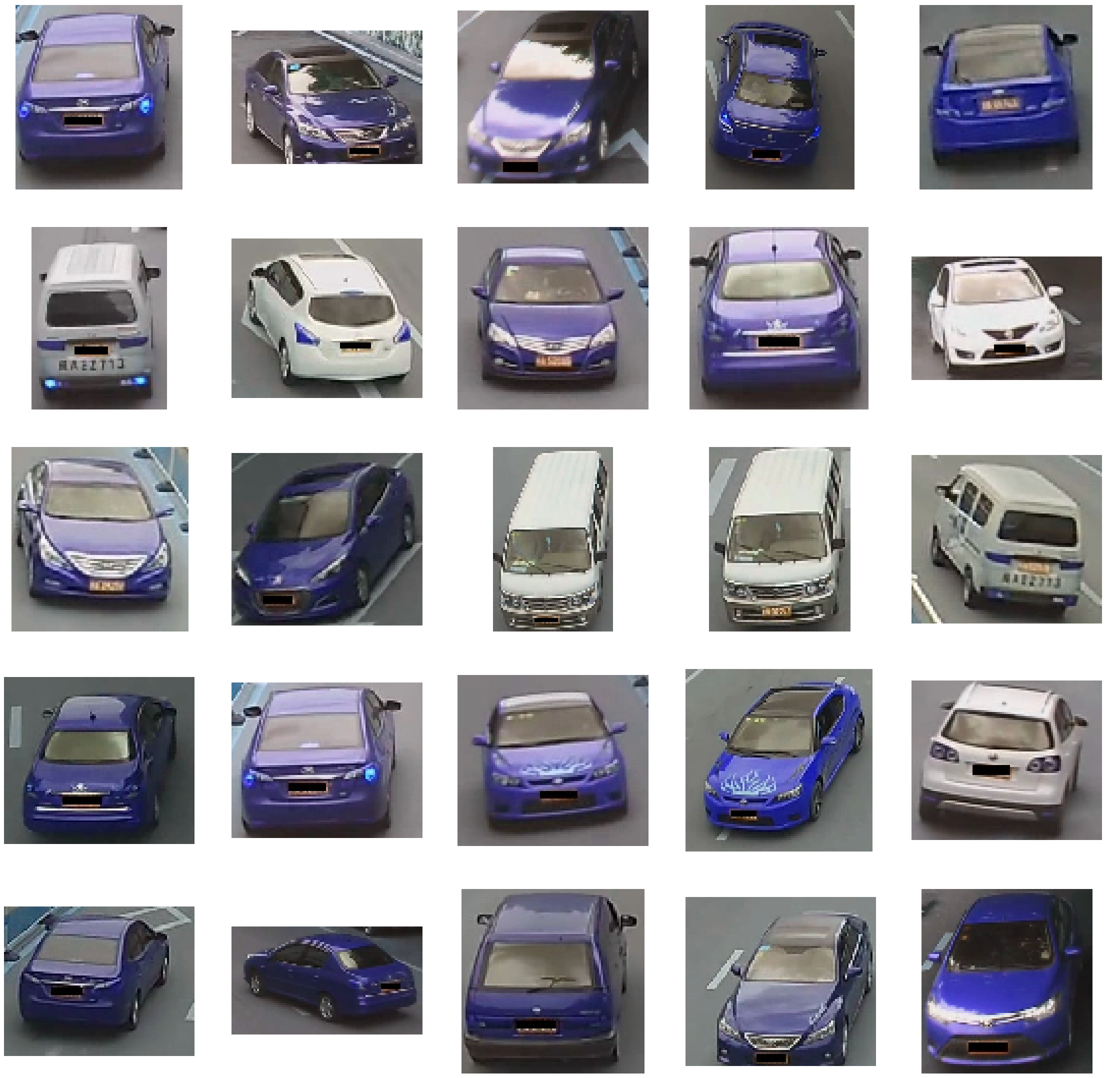
```
In [6]: import numpy as np
        import matplotlib.pyplot as plt
        import cv2
        %matplotlib inline
```

```
In [40]: def string2image(string):
             """Converts a string to a numpy array."""
             img = cv2.imread(string)
             #img = cv2.resize(img,(336,336))
             return img
             #return np.array([int(item) for item in string.split()]).reshape((96, 96))

         def plot_faces(nrows=5, ncols=5):
             """Randomly displays some faces from the training data."""
             selection = np.random.choice(df.index, size=(nrows*ncols), replace=False)
             image_strings = df.loc[selection][0]
             fig, axes = plt.subplots(figsize=(10, 10), nrows=nrows, ncols=ncols)
             for string, ax in zip(image_strings, axes.ravel()):
                 print(string)
                 ax.imshow(string2image(string), cmap='gray')
                 ax.axis('off')
```

```
In [41]: plot_faces()
```

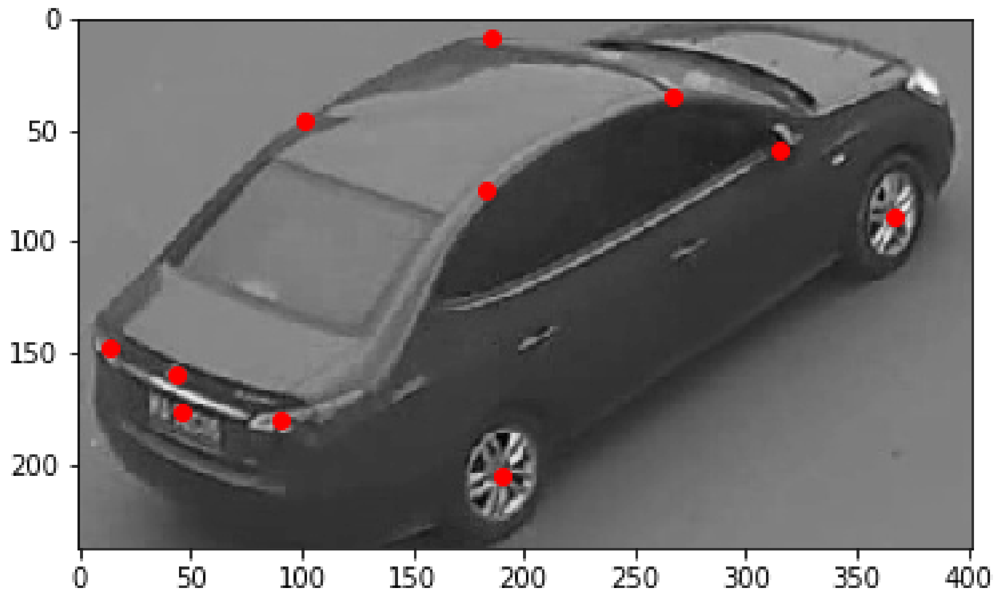
```
VeRi/image_train/0181_c014_00036570_0.jpg
VeRi/image_train/0185_c009_00042525_0.jpg
VeRi/image_train/0185_c008_00042440_0.jpg
VeRi/image_train/0189_c010_00003830_0.jpg
VeRi/image_train/0191_c017_00031640_0.jpg
VeRi/image_train/0297_c014_00077460_0.jpg
VeRi/image_train/0293_c006_00001500_0.jpg
VeRi/image_train/0181_c013_00034610_0.jpg
VeRi/image_train/0190_c014_00005570_0.jpg
VeRi/image_train/0293_c009_00000425_0.jpg
VeRi/image_train/0184_c015_00016750_0.jpg
VeRi/image_train/0189_c017_00000755_0.jpg
VeRi/image_train/0287_c001_00033100_0.jpg
VeRi/image_train/0287_c001_00033080_0.jpg
```

```
VeRi/image_train/0297_c015_00068010_0.jpg
VeRi/image_train/0190_c017_00072295_0.jpg
VeRi/image_train/0181_c014_00036580_0.jpg
VeRi/image_train/0191_c019_00031820_0.jpg
VeRi/image_train/0191_c010_00027390_0.jpg
VeRi/image_train/0289_c002_00049960_0.jpg
VeRi/image_train/0181_c012_00034775_0.jpg
VeRi/image_train/0190_c019_00004155_0.jpg
VeRi/image_train/0194_c003_00052155_0.jpg
VeRi/image_train/0185_c010_00043035_0.jpg
VeRi/image_train/0186_c008_00035660_0.jpg
```



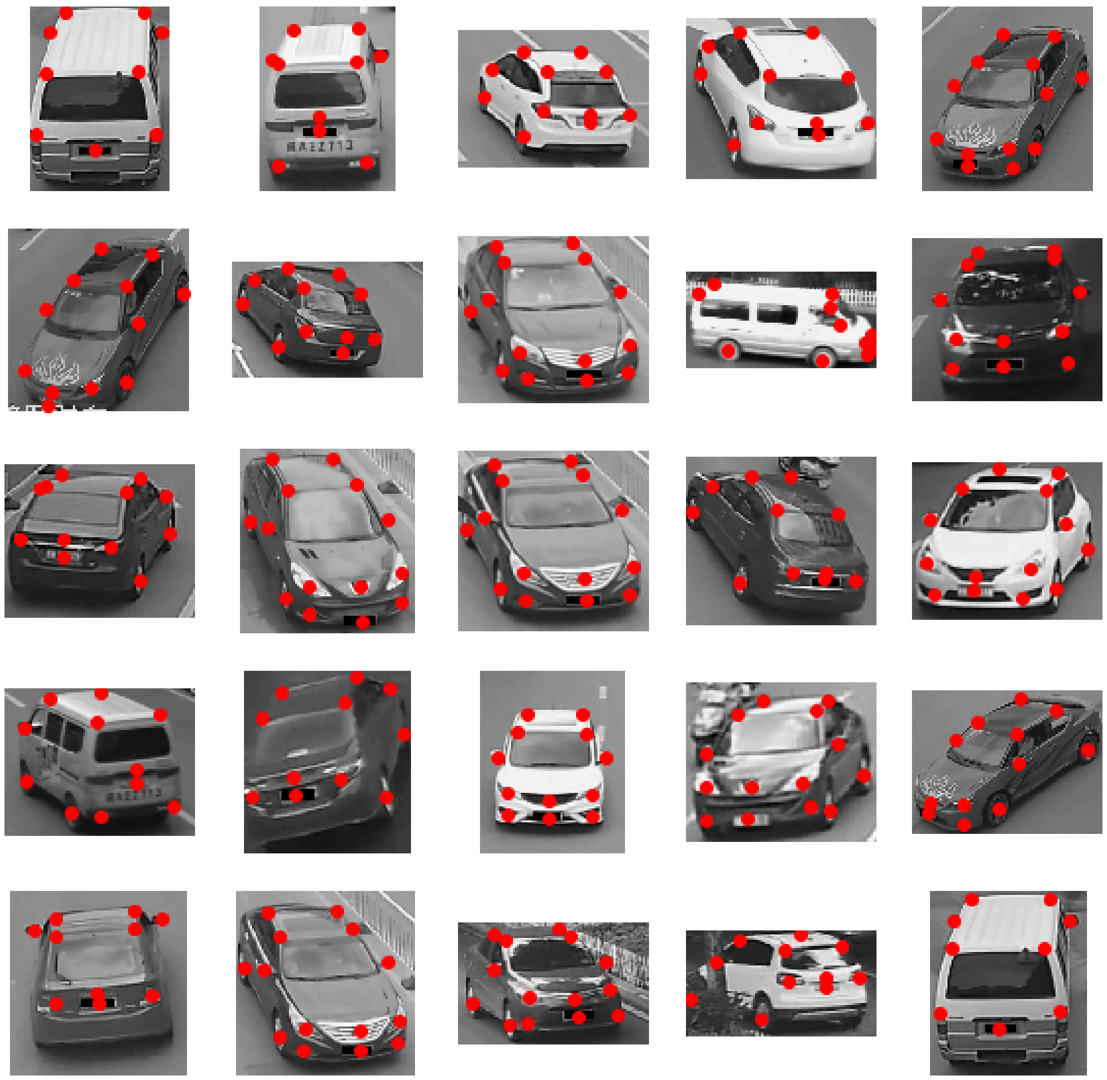
Let's now add to that plot the facial keypoints that were tagged. First, let's do an example :

```
In [9]: keypoint_cols = list(df.columns)[1:-1]
In [10]: def xy_plotfilter(xy):
             y = np.empty((0,2))

             for x in xy:
                 if x[0]!=-1 and x[1]!=-1:
                     y = np.append(y,[x],axis=0)
             return y

In [11]: xy = df.iloc[0][keypoint_cols].values.reshape((20, 2))

         xy = xy_plotfilter(xy)
         print(xy)

[[367 89]
 [190 205]
 [315 59]
 [267 35]
 [185 8]
 [101 46]
 [183 77]
 [14 148]
 [91 180]
 [44 160]
 [46 176]]


In [12]: plt.plot(xy[:, 0], xy[:, 1], 'ro')
         plt.imshow(string2image(df.iloc[0][0]), cmap='gray')

Out[12]: <matplotlib.image.AxesImage at 0x7fab04ebc400>
```



Now, let's add this to the function we wrote before.

```
In [13]: def plot_faces_with_keypoints(nrows=5, ncols=5):
             """Randomly displays some faces from the training data with their keypoints."""
             selection = np.random.choice(df.index, size=(nrows*ncols), replace=False)
             image_strings = df.loc[selection][0]
             keypoint_cols = list(df.columns)[1:-1]
             keypoints = df.loc[selection][keypoint_cols]
             fig, axes = plt.subplots(figsize=(10, 10), nrows=nrows, ncols=ncols)
             for string, (iloc, keypoint), ax in zip(image_strings, keypoints.iterrows(), axes
                 xy = keypoint.values.reshape((20, 2))
                 xy = xy_plotfilter(xy)
                 ax.imshow(string2image(string), cmap='gray')
                 ax.plot(xy[:, 0], xy[:, 1], 'ro')
                 ax.axis('off')

In [14]: plot_faces_with_keypoints()
```



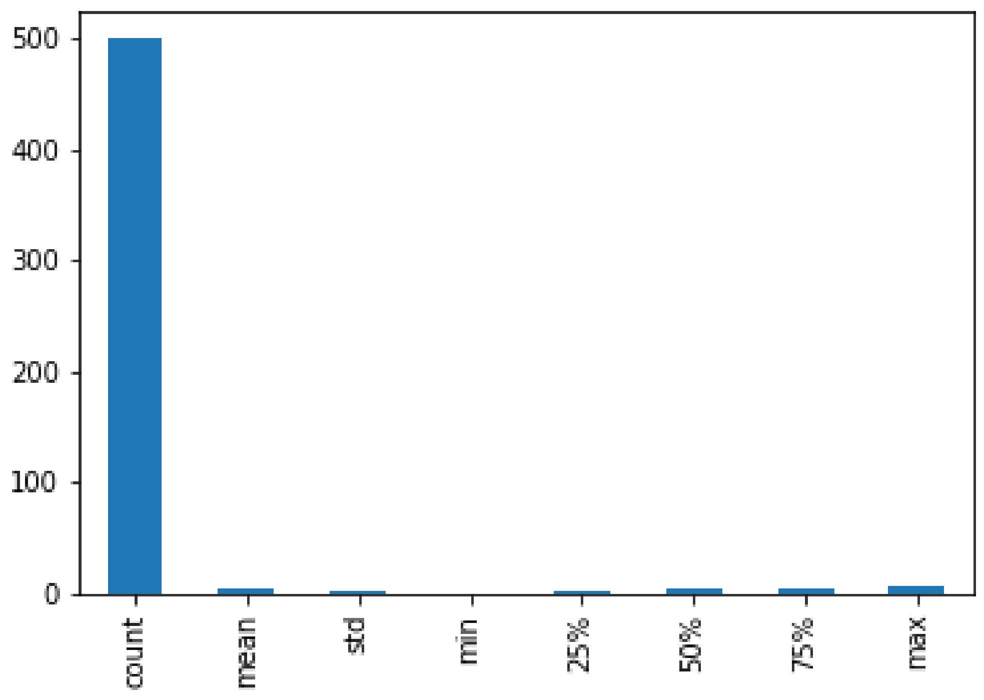
We can make several observations from this image:

- some images are high resolution, some are low
- some images have all 15 keypoints, while some have only a few

Let’s do some statistics about the keypoints to investigate that last observation :

```
In [15]: df.describe().loc[:][41].plot.bar()
```

```
Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x7fab0454ab38>
```



What this plot tells us is that in this dataset, only 2000 images are "high quality" with all key-points, while 5000 other images are "low quality" with only 4 keypoints labelled.

Let's start training the data with the high quality images and see how far we get.

```
In [16]: fully_annotated = df.dropna()

In [17]: fully_annotated.shape

Out[17]: (499, 42)

In [18]: fully_annotated.head()

Out[18]:                                          0   1   2   3   4    5   6    7  \
         0  VeRi/image_train/0181_c001_00034295_0.jpg  -1  -1  -1  -1  367  89  190
         1  VeRi/image_train/0181_c001_00034315_0.jpg  -1  -1  -1  -1  231  58  153
         2  VeRi/image_train/0181_c001_00034340_0.jpg  -1  -1  -1  -1  130  56  114
         3  VeRi/image_train/0181_c012_00034760_0.jpg  -1  -1  -1  -1  213  71  163
         4  VeRi/image_train/0181_c012_00034765_0.jpg  -1  -1  -1  -1  192  66  151

              8   9  ...  32  33   34   35   36  37   38  39   40  41
         0  205  -1  ...  77  14  148   91  180  44  160  46  176   7
         1  132  -1  ...  32  14   94  108  108  55  100  58  114   7
         2   92  -1  ...  19  18   64   96   69  55   62  54   72   7
         3  141  -1  ...  44  22  100  123  113  67  106  70  123   7
         4  124  -1  ...  36  25   89  116   97  68   93  73  108   7

         [5 rows x 42 columns]
```

3 Building a Keras model

Now on to the machine learning part. Let's build a Keras model with our data. Actually, before we do that, let's do some preprocessing first, using the scikit-learn pipelines (inspired by this great post on scalable Machine Learning by Tom Augspurger).

The idea behind pipelining is that it allows you to easily keep track of the data transformations applied to our data. We need two scalings: one for the input and one for the output. Since I couldn't get the scaling to work for 3d image data, we will only use a pipeline for our outputs.

```
In [19]: X = []
         i = 0
         y = np.vstack(fully_annotated[fully_annotated.columns[1:-1]].values)
         for string in fully_annotated[:][0]:
             img = string2image(string)
             for j in range(0,40,2):
                 if y[i][j] != -1:
                     y[i][j] = y[i][j] / img.shape[1] * 336
                 if y[i][j+1] != -1:
                     y[i][j+1] = y[i][j+1] / img.shape[0] * 336
             img = cv2.resize(img,(336,336))
             X.append(img)
             #print(img.shape)
             #X = np.stack((X,[img])).astype(np.float)[:, :, :, np.newaxis]
             #X = np.concatenate((X,img)).astype(np.float)[:, :, :, np.newaxis]
             i+=1
         X = np.array(X)
         X = X.reshape(499, 336,336,1)
         X.shape

         #X = np.stack([cv2.resize(string2image(string),(336,336)) for string in fully_annotat

Out[19]: (499, 336, 336, 1)

In [20]: for i in range(0,5):
             print(y[i])

[ -1  -1  -1  -1 306 125 158 289  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1
  -1  -1  -1  -1 263  83 223  49 154  11  84  64 152 108  11 208  76 254
  36 225  38 248]
[ -1  -1  -1  -1 318 121 210 275  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1
  -1  -1  -1  -1 291  58 236  18 128   4  82  48 196  66  19 196 148 225
  75 208  79 237]
[ -1  -1  -1  -1 314 165 275 271  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1
  -1  -1  62  50 314  64 256  20  99  14  79  50 241  56  43 188 232 203
 132 182 130 212]
[ -1  -1  -1  -1 307 140 235 278  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1
  -1  -1  93  41 294  75 248  35 118  21  83  65 214  86  31 197 177 223
  96 209 100 243]
[ -1  -1  -1  -1 286 134 225 252  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1
```

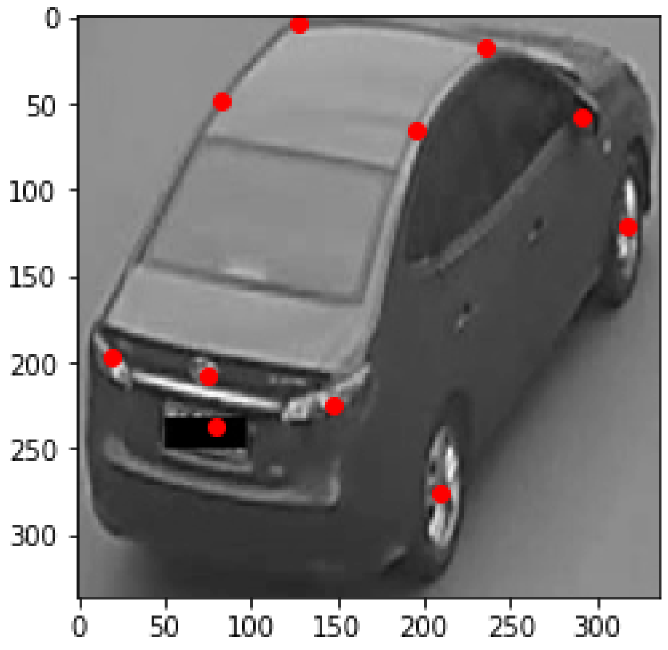
```
  -1  -1  83  40 274  67 235  32 110  20  83  52 207  73  37 181 173 197
 101 189 109 219]
```

```
In [21]: xy1 = y[1].reshape((20, 2))
         xy1 = xy_plotfilter(xy1)
         xy1
```

```
Out[21]: array([[318., 121.],
                [210., 275.],
                [291.,  58.],
                [236.,  18.],
                [128.,   4.],
                [ 82.,  48.],
                [196.,  66.],
                [ 19., 196.],
                [148., 225.],
                [ 75., 208.],
                [ 79., 237.]])
```

```
In [22]: #keypoint_cols = list(df.columns)[1:-1]

         plt.plot(xy1[:, 0], xy1[:, 1], 'ro')
         img1 = string2image(fully_annotated[:][0][1])
         img1 = cv2.resize(img1,(336,336))
         plt.imshow(img1, cmap='gray')
```

```
Out[22]: <matplotlib.image.AxesImage at 0x7fab044dbe80>
```



```
In [23]: X.shape, X.dtype

Out[23]: ((499, 336, 336, 1), dtype('uint8'))

In [24]: y.shape, y.dtype

Out[24]: ((499, 40), dtype('int64'))

In [25]: X_train = X / 255.0

In [26]: from sklearn.pipeline import make_pipeline
         from sklearn.preprocessing import MinMaxScaler

         output_pipe = make_pipeline(
             MinMaxScaler(feature_range=(-1, 1))
         )

         y_train = output_pipe.fit_transform(y)
         y_train.shape

Out[26]: (499, 40)

In [27]: for i in range(0,5):
             print(y_train[i])
```

```
[-1.         -1.         -1.         -1.          0.88343558 -0.18446602
  0.          0.84713376 -1.         -1.         -1.         -1.
 -1.         -1.         -1.         -1.         -1.         -1.
 -1.         -1.         -1.         -1.          0.6146789  -0.328
  0.47854785 -0.46236559  0.02649007 -0.87301587 -0.43143813 -0.23976608
 -0.05555556  0.26744186 -0.9266055   0.33121019 -0.53892216  0.63461538
 -0.76357827  0.65567766 -0.75316456  0.61165049]
[-1.         -1.         -1.         -1.          0.95705521 -0.21035599
  0.32704403  0.75796178 -1.         -1.         -1.         -1.
 -1.         -1.         -1.         -1.         -1.         -1.
 -1.         -1.         -1.         -1.          0.78593272 -0.528
  0.56435644 -0.79569892 -0.14569536 -0.94708995 -0.44481605 -0.42690058
  0.21604938 -0.22093023 -0.87767584  0.25477707 -0.10778443  0.44871795
 -0.514377    0.53113553 -0.49367089  0.54045307]
[-1.         -1.         -1.         -1.          0.93251534  0.07443366
  0.73584906  0.73248408 -1.         -1.         -1.         -1.
 -1.         -1.         -1.         -1.         -1.         -1.
 -1.         -1.         -0.61111111 -0.49253731  0.9266055  -0.48
  0.69636964 -0.77419355 -0.33774834 -0.84126984 -0.46488294 -0.40350877
  0.49382716 -0.3372093  -0.73088685  0.20382166  0.39520958  0.30769231
 -0.15015974  0.34065934 -0.17088608  0.37864078]
```

```
[-1.         -1.         -1.         -1.          0.88957055 -0.08737864
  0.48427673  0.77707006 -1.         -1.         -1.         -1.
 -1.         -1.         -1.         -1.         -1.         -1.
 -1.         -1.         -0.41975309 -0.58208955  0.80428135 -0.392
  0.64356436 -0.61290323 -0.21192053 -0.76719577 -0.43812709 -0.22807018
  0.32716049  0.01162791 -0.80428135  0.2611465   0.06586826  0.43589744
 -0.38019169  0.53846154 -0.36075949  0.57928803]
[-1.         -1.         -1.         -1.          0.7607362  -0.12621359
  0.42138365  0.61146497 -1.         -1.         -1.         -1.
 -1.         -1.         -1.         -1.         -1.         -1.
 -1.         -1.         -0.48148148 -0.5920398   0.68195719 -0.456
  0.55775578 -0.64516129 -0.26490066 -0.77777778 -0.43812709 -0.38011696
  0.28395062 -0.13953488 -0.7675841   0.15923567  0.04191617  0.26923077
 -0.34824281  0.39194139 -0.30379747  0.42394822]
```

4 Transfer learning for 8 classes

```
In [28]: #Importing the ResNet50 model
         from keras.applications.resnet50 import ResNet50, preprocess_input

         #Loading the ResNet50 model with pre-trained ImageNet weights
         model = ResNet50(weights='imagenet', include_top=False, input_shape=(336, 336, 3))
```

```
WARNING: Logging before flag parsing goes to stderr.
W1130 23:10:07.787857 140373928044352 deprecation_wrapper.py:119] From /home/william/anaconda3

W1130 23:10:07.816876 140373928044352 deprecation_wrapper.py:119] From /home/william/anaconda3

W1130 23:10:07.823477 140373928044352 deprecation_wrapper.py:119] From /home/william/anaconda3

W1130 23:10:07.845628 140373928044352 deprecation_wrapper.py:119] From /home/william/anaconda3

W1130 23:10:07.846690 140373928044352 deprecation_wrapper.py:119] From /home/william/anaconda3

W1130 23:10:07.972143 140373928044352 deprecation_wrapper.py:119] From /home/william/anaconda3

W1130 23:10:08.036344 140373928044352 deprecation_wrapper.py:119] From /home/william/anaconda3

/home/william/anaconda3/envs/tfcpu/lib/python3.6/site-packages/keras_applications/resnet50.py:
  warnings.warn('The output shape of `ResNet50(include_top=False)` '
```

```
In [29]: #Reshaping the training data
         X_train_new = np.array([np.resize(X_train[i], (336, 336, 3)) for i in range(0, len(X_

         #Preprocessing the data, so that it can be fed to the pre-trained ResNet50 model.
         resnet_train_input = preprocess_input(X_train_new)
         #Creating bottleneck features for the training data
         train_features = model.predict(resnet_train_input)

         #Saving the bottleneck features
         np.savez('resnet_features_train', features=train_features)
```

```
In [30]: img = X_train_new[0:2]#.reshape(1, -1)
         img1 = preprocess_input(img)
         print(img1.shape)
         print(resnet_train_input.shape)
```

```
(2, 336, 336, 3)
(499, 336, 336, 3)
```

```
In [56]: #Reshaping the testing data
         X_test_new = np.array([imresize(X_test[i], (336, 336, 3)) for i in range(0, len(X_tes

         #Preprocessing the data, so that it can be fed to the pre-trained ResNet50 model.
         resnet_test_input = preprocess_input(X_test_new)

         #Creating bottleneck features for the testing data
         test_features = model.predict(resnet_test_input)

         #Saving the bottleneck features
         np.savez('resnet_features_test', features=test_features)
```

```
        ---------------------------------------------------------------------------

        NameError                                 Traceback (most recent call last)

        <ipython-input-56-463595940c9c> in <module>
          1 #Reshaping the testing data
    ----> 2 X_test_new = np.array([imresize(X_test[i], (336, 336, 3)) for i in range(0, len(X_
          3
          4 #Preprocessing the data, so that it can be fed to the pre-trained ResNet50 model.
          5 resnet_test_input = preprocess_input(X_test_new)


        NameError: name 'X_test' is not defined
```

```
In [96]: from keras.models import Sequential
         from keras.layers import Dense, Conv2D, MaxPooling2D
         from keras.layers import Dropout, Flatten, GlobalAveragePooling2D
```

```
In [114]: model = Sequential()
          model.add(GlobalAveragePooling2D(input_shape=(336,336,3)))
          model.add(Dropout(0.3))
          model.add(Dense(8, activation='softmax'))
          model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
global_average_pooling2d_4 ( (None, 3)                 0         
_________________________________________________________________
dropout_3 (Dropout)          (None, 3)                 0         
_________________________________________________________________
dense_5 (Dense)              (None, 8)                 32        
=================================================================
Total params: 32
Trainable params: 32
Non-trainable params: 0
_________________________________________________________________
```

```
In [115]: model.compile(loss='categorical_crossentropy', optimizer='adam',
                        metrics=['accuracy'])
```

```
In [33]: y_train1 = np.vstack(fully_annotated[fully_annotated.columns[-1]].values)
         y_train1 = np.resize(y_train1, (499))
         num_classes = 8
         from keras.utils import np_utils
         y_train1 = np_utils.to_categorical(y_train1, num_classes)
         y_train1
```

```
Out[33]: array([[0., 0., 0., ..., 0., 0., 1.],
                [0., 0., 0., ..., 0., 0., 1.],
                [0., 0., 0., ..., 0., 0., 1.],
                ...,
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.],
                [0., 0., 0., ..., 0., 0., 0.]], dtype=float32)
```

```
In [117]: import keras
          checkpointer = keras.callbacks.ModelCheckpoint(filepath='scratchmodel.best.hdf5',
                                         verbose=1,save_best_only=True)
```

```
In [118]: model.fit(train_features, y_train1, batch_size=32, epochs=200,
                    validation_split=0.2, callbacks=[checkpointer], verbose=1, shuffle=True)
```

```
        ---------------------------------------------------------------------------

        ValueError                                Traceback (most recent call last)
```

```
    <ipython-input-118-248459b92ea6> in <module>
      1 model.fit(train_features, y_train1, batch_size=32, epochs=200,
----> 2             validation_split=0.2, callbacks=[checkpointer], verbose=1, shuffle=True)


    ~/anaconda3/envs/tfcpu/lib/python3.6/site-packages/keras/engine/training.py in fit(sel
    950             sample_weight=sample_weight,
    951             class_weight=class_weight,
--> 952             batch_size=batch_size)
    953         # Prepare validation data.
    954         do_validation = False


    ~/anaconda3/envs/tfcpu/lib/python3.6/site-packages/keras/engine/training.py in _standa
    749             feed_input_shapes,
    750             check_batch_axis=False,  # Don't enforce the batch size.
--> 751             exception_prefix='input')
    752
    753         if y is not None:


    ~/anaconda3/envs/tfcpu/lib/python3.6/site-packages/keras/engine/training_utils.py in s
    136                                 ': expected ' + names[i] + ' to have shape ' +
    137                                 str(shape) + ' but got array with shape ' +
--> 138                                 str(data_shape))
    139     return data
    140


    ValueError: Error when checking input: expected global_average_pooling2d_4_input to ha
```

```
In [103]: X_train[0:2].shape
```

```
Out[103]: (2, 336, 336, 1)
```

```
In [110]: from keras.applications.resnet50 import ResNet50, preprocess_input
          model_input_shape = (1,)+model.get_input_shape_at(0)[1:]
          img = X_train_new[0:2]#.reshape(1, -1)
          img1 = preprocess_input(X_train_new)
          print(img1.shape)
          #prediction = model.predict(img1)
          #predictions = model.predict(img)
          #xy_predictions = output_pipe.inverse_transform(predictions).reshape(20, 2)
          #plt.imshow(X_train[2, :, :, 0], cmap='gray')
          #plt.plot(xy_predictions[:, 0], xy_predictions[:, 1], 'b*')
          #prediction
          score  = model.evaluate(train_features, y_train1)
          score[1]
```

```
(499, 336, 336, 3)
499/499 [==============================] - 1s 1ms/step
```

```
Out[110]: 0.791583166810458
```

```
In [112]: test_predictions = model.predict(X_train_new)
```

```
        ---------------------------------------------------------------------------

        ValueError                                Traceback (most recent call last)

        <ipython-input-112-098aee91bc3f> in <module>
    ----> 1 test_predictions = model.predict(X_train_new)


        ~/anaconda3/envs/tfcpu/lib/python3.6/site-packages/keras/engine/training.py in predict
       1147                                  'argument.')
       1148         # Validate user data.
    -> 1149         x, _, _ = self._standardize_user_data(x)
       1150         if self.stateful:
       1151             if x[0].shape[0] > batch_size and x[0].shape[0] % batch_size != 0:


        ~/anaconda3/envs/tfcpu/lib/python3.6/site-packages/keras/engine/training.py in _standa
        749             feed_input_shapes,
        750             check_batch_axis=False,  # Don't enforce the batch size.
    --> 751             exception_prefix='input')
        752
        753         if y is not None:


        ~/anaconda3/envs/tfcpu/lib/python3.6/site-packages/keras/engine/training_utils.py in s
        136                                 ': expected ' + names[i] + ' to have shape ' +
        137                                 str(shape) + ' but got array with shape ' +
    --> 138                                 str(data_shape))
        139     return data
        140


        ValueError: Error when checking input: expected global_average_pooling2d_3_input to ha
```

```
In [31]: from keras.models import Model
         from keras.optimizers import Adam
         from keras.layers import GlobalAveragePooling2D
         from keras.layers import Dense
         from keras.applications.inception_v3 import InceptionV3
         from keras.utils.np_utils import to_categorical

         # Get the InceptionV3 model so we can do transfer learning
         base_inception = InceptionV3(weights='imagenet', include_top=False,
                                      input_shape=(336, 336, 3))

         # Add a global spatial average pooling layer
         out = base_inception.output
         out = GlobalAveragePooling2D()(out)
         out = Dense(512, activation='relu')(out)
         out = Dense(512, activation='relu')(out)
         total_classes = 8
         predictions = Dense(total_classes, activation='softmax')(out)

         model = Model(inputs=base_inception.input, outputs=predictions)

         # only if we want to freeze layers
         for layer in base_inception.layers:
             layer.trainable = False

         # Compile
         model.compile(Adam(lr=.0001), loss='categorical_crossentropy', metrics=['accuracy'])
         model.summary()
```

```
W1130 23:12:33.649125 140373928044352 deprecation_wrapper.py:119] From /home/william/anaconda3/

W1130 23:12:46.188709 140373928044352 deprecation_wrapper.py:119] From /home/william/anaconda3/



________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
================================================================================================
input_2 (InputLayer)            (None, 336, 336, 3)  0

________________________________________________________________________________________________
conv2d_1 (Conv2D)               (None, 167, 167, 32) 864         input_2[0][0]

________________________________________________________________________________________________
batch_normalization_1 (BatchNor (None, 167, 167, 32) 96          conv2d_1[0][0]

________________________________________________________________________________________________
activation_50 (Activation)      (None, 167, 167, 32) 0           batch_normalization_1[0][0]

________________________________________________________________________________________________
conv2d_2 (Conv2D)               (None, 165, 165, 32) 9216        activation_50[0][0]

________________________________________________________________________________________________
batch_normalization_2 (BatchNor (None, 165, 165, 32) 96          conv2d_2[0][0]

________________________________________________________________________________________________
activation_51 (Activation)      (None, 165, 165, 32) 0           batch_normalization_2[0][0]

________________________________________________________________________________________________
conv2d_3 (Conv2D)               (None, 165, 165, 64) 18432       activation_51[0][0]
```

```
----------------------------------------------------------------------------------------------
batch_normalization_3 (BatchNor (None, 165, 165, 64) 192         conv2d_3[0][0]

----------------------------------------------------------------------------------------------
activation_52 (Activation)      (None, 165, 165, 64) 0           batch_normalization_3[0][0]

----------------------------------------------------------------------------------------------
max_pooling2d_2 (MaxPooling2D)  (None, 82, 82, 64)   0           activation_52[0][0]

----------------------------------------------------------------------------------------------
conv2d_4 (Conv2D)               (None, 82, 82, 80)   5120        max_pooling2d_2[0][0]

----------------------------------------------------------------------------------------------
batch_normalization_4 (BatchNor (None, 82, 82, 80)   240         conv2d_4[0][0]

----------------------------------------------------------------------------------------------
activation_53 (Activation)      (None, 82, 82, 80)   0           batch_normalization_4[0][0]

----------------------------------------------------------------------------------------------
conv2d_5 (Conv2D)               (None, 80, 80, 192)  138240      activation_53[0][0]

----------------------------------------------------------------------------------------------
batch_normalization_5 (BatchNor (None, 80, 80, 192)  576         conv2d_5[0][0]

----------------------------------------------------------------------------------------------
activation_54 (Activation)      (None, 80, 80, 192)  0           batch_normalization_5[0][0]

----------------------------------------------------------------------------------------------
max_pooling2d_3 (MaxPooling2D)  (None, 39, 39, 192)  0           activation_54[0][0]

----------------------------------------------------------------------------------------------
conv2d_9 (Conv2D)               (None, 39, 39, 64)   12288       max_pooling2d_3[0][0]

----------------------------------------------------------------------------------------------
batch_normalization_9 (BatchNor (None, 39, 39, 64)   192         conv2d_9[0][0]

----------------------------------------------------------------------------------------------
activation_58 (Activation)      (None, 39, 39, 64)   0           batch_normalization_9[0][0]

----------------------------------------------------------------------------------------------
conv2d_7 (Conv2D)               (None, 39, 39, 48)   9216        max_pooling2d_3[0][0]

----------------------------------------------------------------------------------------------
conv2d_10 (Conv2D)              (None, 39, 39, 96)   55296       activation_58[0][0]

----------------------------------------------------------------------------------------------
batch_normalization_7 (BatchNor (None, 39, 39, 48)   144         conv2d_7[0][0]

----------------------------------------------------------------------------------------------
batch_normalization_10 (BatchNo (None, 39, 39, 96)   288         conv2d_10[0][0]

----------------------------------------------------------------------------------------------
activation_56 (Activation)      (None, 39, 39, 48)   0           batch_normalization_7[0][0]

----------------------------------------------------------------------------------------------
activation_59 (Activation)      (None, 39, 39, 96)   0           batch_normalization_10[0][0]

----------------------------------------------------------------------------------------------
average_pooling2d_1 (AveragePoo (None, 39, 39, 192)  0           max_pooling2d_3[0][0]

----------------------------------------------------------------------------------------------
conv2d_6 (Conv2D)               (None, 39, 39, 64)   12288       max_pooling2d_3[0][0]

----------------------------------------------------------------------------------------------
conv2d_8 (Conv2D)               (None, 39, 39, 64)   76800       activation_56[0][0]

----------------------------------------------------------------------------------------------
conv2d_11 (Conv2D)              (None, 39, 39, 96)   82944       activation_59[0][0]

----------------------------------------------------------------------------------------------
conv2d_12 (Conv2D)              (None, 39, 39, 32)   6144        average_pooling2d_1[0][0]
```

```
________________________________________________________________________________________________
batch_normalization_6 (BatchNor (None, 39, 39, 64)   192         conv2d_6[0][0]
________________________________________________________________________________________________
batch_normalization_8 (BatchNor (None, 39, 39, 64)   192         conv2d_8[0][0]
________________________________________________________________________________________________
batch_normalization_11 (BatchNo (None, 39, 39, 96)   288         conv2d_11[0][0]
________________________________________________________________________________________________
batch_normalization_12 (BatchNo (None, 39, 39, 32)   96          conv2d_12[0][0]
________________________________________________________________________________________________
activation_55 (Activation)      (None, 39, 39, 64)   0           batch_normalization_6[0][0]
________________________________________________________________________________________________
activation_57 (Activation)      (None, 39, 39, 64)   0           batch_normalization_8[0][0]
________________________________________________________________________________________________
activation_60 (Activation)      (None, 39, 39, 96)   0           batch_normalization_11[0][0]
________________________________________________________________________________________________
activation_61 (Activation)      (None, 39, 39, 32)   0           batch_normalization_12[0][0]
________________________________________________________________________________________________
mixed0 (Concatenate)            (None, 39, 39, 256)  0           activation_55[0][0]
                                                                 activation_57[0][0]
                                                                 activation_60[0][0]
                                                                 activation_61[0][0]
________________________________________________________________________________________________
conv2d_16 (Conv2D)              (None, 39, 39, 64)   16384       mixed0[0][0]
________________________________________________________________________________________________
batch_normalization_16 (BatchNo (None, 39, 39, 64)   192         conv2d_16[0][0]
________________________________________________________________________________________________
activation_65 (Activation)      (None, 39, 39, 64)   0           batch_normalization_16[0][0]
________________________________________________________________________________________________
conv2d_14 (Conv2D)              (None, 39, 39, 48)   12288       mixed0[0][0]
________________________________________________________________________________________________
conv2d_17 (Conv2D)              (None, 39, 39, 96)   55296       activation_65[0][0]
________________________________________________________________________________________________
batch_normalization_14 (BatchNo (None, 39, 39, 48)   144         conv2d_14[0][0]
________________________________________________________________________________________________
batch_normalization_17 (BatchNo (None, 39, 39, 96)   288         conv2d_17[0][0]
________________________________________________________________________________________________
activation_63 (Activation)      (None, 39, 39, 48)   0           batch_normalization_14[0][0]
________________________________________________________________________________________________
activation_66 (Activation)      (None, 39, 39, 96)   0           batch_normalization_17[0][0]
________________________________________________________________________________________________
average_pooling2d_2 (AveragePoo (None, 39, 39, 256)  0           mixed0[0][0]
________________________________________________________________________________________________
conv2d_13 (Conv2D)              (None, 39, 39, 64)   16384       mixed0[0][0]
________________________________________________________________________________________________
conv2d_15 (Conv2D)              (None, 39, 39, 64)   76800       activation_63[0][0]
________________________________________________________________________________________________
conv2d_18 (Conv2D)              (None, 39, 39, 96)   82944       activation_66[0][0]
________________________________________________________________________________________________
```

```
conv2d_19 (Conv2D)             (None, 39, 39, 64)   16384       average_pooling2d_2[0][0]
__________________________________________________________________________________________________
batch_normalization_13 (BatchNo (None, 39, 39, 64)   192         conv2d_13[0][0]
__________________________________________________________________________________________________
batch_normalization_15 (BatchNo (None, 39, 39, 64)   192         conv2d_15[0][0]
__________________________________________________________________________________________________
batch_normalization_18 (BatchNo (None, 39, 39, 96)   288         conv2d_18[0][0]
__________________________________________________________________________________________________
batch_normalization_19 (BatchNo (None, 39, 39, 64)   192         conv2d_19[0][0]
__________________________________________________________________________________________________
activation_62 (Activation)      (None, 39, 39, 64)   0           batch_normalization_13[0][0]
__________________________________________________________________________________________________
activation_64 (Activation)      (None, 39, 39, 64)   0           batch_normalization_15[0][0]
__________________________________________________________________________________________________
activation_67 (Activation)      (None, 39, 39, 96)   0           batch_normalization_18[0][0]
__________________________________________________________________________________________________
activation_68 (Activation)      (None, 39, 39, 64)   0           batch_normalization_19[0][0]
__________________________________________________________________________________________________
mixed1 (Concatenate)            (None, 39, 39, 288)  0           activation_62[0][0]
                                                                 activation_64[0][0]
                                                                 activation_67[0][0]
                                                                 activation_68[0][0]
__________________________________________________________________________________________________
conv2d_23 (Conv2D)             (None, 39, 39, 64)   18432       mixed1[0][0]
__________________________________________________________________________________________________
batch_normalization_23 (BatchNo (None, 39, 39, 64)   192         conv2d_23[0][0]
__________________________________________________________________________________________________
activation_72 (Activation)      (None, 39, 39, 64)   0           batch_normalization_23[0][0]
__________________________________________________________________________________________________
conv2d_21 (Conv2D)             (None, 39, 39, 48)   13824       mixed1[0][0]
__________________________________________________________________________________________________
conv2d_24 (Conv2D)             (None, 39, 39, 96)   55296       activation_72[0][0]
__________________________________________________________________________________________________
batch_normalization_21 (BatchNo (None, 39, 39, 48)   144         conv2d_21[0][0]
__________________________________________________________________________________________________
batch_normalization_24 (BatchNo (None, 39, 39, 96)   288         conv2d_24[0][0]
__________________________________________________________________________________________________
activation_70 (Activation)      (None, 39, 39, 48)   0           batch_normalization_21[0][0]
__________________________________________________________________________________________________
activation_73 (Activation)      (None, 39, 39, 96)   0           batch_normalization_24[0][0]
__________________________________________________________________________________________________
average_pooling2d_3 (AveragePoo (None, 39, 39, 288)  0           mixed1[0][0]
__________________________________________________________________________________________________
conv2d_20 (Conv2D)             (None, 39, 39, 64)   18432       mixed1[0][0]
__________________________________________________________________________________________________
conv2d_22 (Conv2D)             (None, 39, 39, 64)   76800       activation_70[0][0]
__________________________________________________________________________________________________
conv2d_25 (Conv2D)             (None, 39, 39, 96)   82944       activation_73[0][0]
```

```
----------------------------------------------------------------------------------------------------
conv2d_26 (Conv2D)              (None, 39, 39, 64)   18432       average_pooling2d_3[0][0]

----------------------------------------------------------------------------------------------------
batch_normalization_20 (BatchNo (None, 39, 39, 64)   192         conv2d_20[0][0]

----------------------------------------------------------------------------------------------------
batch_normalization_22 (BatchNo (None, 39, 39, 64)   192         conv2d_22[0][0]

----------------------------------------------------------------------------------------------------
batch_normalization_25 (BatchNo (None, 39, 39, 96)   288         conv2d_25[0][0]

----------------------------------------------------------------------------------------------------
batch_normalization_26 (BatchNo (None, 39, 39, 64)   192         conv2d_26[0][0]

----------------------------------------------------------------------------------------------------
activation_69 (Activation)      (None, 39, 39, 64)   0           batch_normalization_20[0][0]

----------------------------------------------------------------------------------------------------
activation_71 (Activation)      (None, 39, 39, 64)   0           batch_normalization_22[0][0]

----------------------------------------------------------------------------------------------------
activation_74 (Activation)      (None, 39, 39, 96)   0           batch_normalization_25[0][0]

----------------------------------------------------------------------------------------------------
activation_75 (Activation)      (None, 39, 39, 64)   0           batch_normalization_26[0][0]

----------------------------------------------------------------------------------------------------
mixed2 (Concatenate)            (None, 39, 39, 288)  0           activation_69[0][0]
                                                                 activation_71[0][0]
                                                                 activation_74[0][0]
                                                                 activation_75[0][0]

----------------------------------------------------------------------------------------------------
conv2d_28 (Conv2D)              (None, 39, 39, 64)   18432       mixed2[0][0]

----------------------------------------------------------------------------------------------------
batch_normalization_28 (BatchNo (None, 39, 39, 64)   192         conv2d_28[0][0]

----------------------------------------------------------------------------------------------------
activation_77 (Activation)      (None, 39, 39, 64)   0           batch_normalization_28[0][0]

----------------------------------------------------------------------------------------------------
conv2d_29 (Conv2D)              (None, 39, 39, 96)   55296       activation_77[0][0]

----------------------------------------------------------------------------------------------------
batch_normalization_29 (BatchNo (None, 39, 39, 96)   288         conv2d_29[0][0]

----------------------------------------------------------------------------------------------------
activation_78 (Activation)      (None, 39, 39, 96)   0           batch_normalization_29[0][0]

----------------------------------------------------------------------------------------------------
conv2d_27 (Conv2D)              (None, 19, 19, 384)  995328      mixed2[0][0]

----------------------------------------------------------------------------------------------------
conv2d_30 (Conv2D)              (None, 19, 19, 96)   82944       activation_78[0][0]

----------------------------------------------------------------------------------------------------
batch_normalization_27 (BatchNo (None, 19, 19, 384)  1152        conv2d_27[0][0]

----------------------------------------------------------------------------------------------------
batch_normalization_30 (BatchNo (None, 19, 19, 96)   288         conv2d_30[0][0]

----------------------------------------------------------------------------------------------------
activation_76 (Activation)      (None, 19, 19, 384)  0           batch_normalization_27[0][0]

----------------------------------------------------------------------------------------------------
activation_79 (Activation)      (None, 19, 19, 96)   0           batch_normalization_30[0][0]

----------------------------------------------------------------------------------------------------
```

```
max_pooling2d_4 (MaxPooling2D)  (None, 19, 19, 288)  0           mixed2[0][0]
__________________________________________________________________________________________________
mixed3 (Concatenate)            (None, 19, 19, 768)  0           activation_76[0][0]
                                                                 activation_79[0][0]
                                                                 max_pooling2d_4[0][0]
__________________________________________________________________________________________________
conv2d_35 (Conv2D)              (None, 19, 19, 128)  98304       mixed3[0][0]
__________________________________________________________________________________________________
batch_normalization_35 (BatchNo (None, 19, 19, 128)  384         conv2d_35[0][0]
__________________________________________________________________________________________________
activation_84 (Activation)      (None, 19, 19, 128)  0           batch_normalization_35[0][0]
__________________________________________________________________________________________________
conv2d_36 (Conv2D)              (None, 19, 19, 128)  114688      activation_84[0][0]
__________________________________________________________________________________________________
batch_normalization_36 (BatchNo (None, 19, 19, 128)  384         conv2d_36[0][0]
__________________________________________________________________________________________________
activation_85 (Activation)      (None, 19, 19, 128)  0           batch_normalization_36[0][0]
__________________________________________________________________________________________________
conv2d_32 (Conv2D)              (None, 19, 19, 128)  98304       mixed3[0][0]
__________________________________________________________________________________________________
conv2d_37 (Conv2D)              (None, 19, 19, 128)  114688      activation_85[0][0]
__________________________________________________________________________________________________
batch_normalization_32 (BatchNo (None, 19, 19, 128)  384         conv2d_32[0][0]
__________________________________________________________________________________________________
batch_normalization_37 (BatchNo (None, 19, 19, 128)  384         conv2d_37[0][0]
__________________________________________________________________________________________________
activation_81 (Activation)      (None, 19, 19, 128)  0           batch_normalization_32[0][0]
__________________________________________________________________________________________________
activation_86 (Activation)      (None, 19, 19, 128)  0           batch_normalization_37[0][0]
__________________________________________________________________________________________________
conv2d_33 (Conv2D)              (None, 19, 19, 128)  114688      activation_81[0][0]
__________________________________________________________________________________________________
conv2d_38 (Conv2D)              (None, 19, 19, 128)  114688      activation_86[0][0]
__________________________________________________________________________________________________
batch_normalization_33 (BatchNo (None, 19, 19, 128)  384         conv2d_33[0][0]
__________________________________________________________________________________________________
batch_normalization_38 (BatchNo (None, 19, 19, 128)  384         conv2d_38[0][0]
__________________________________________________________________________________________________
activation_82 (Activation)      (None, 19, 19, 128)  0           batch_normalization_33[0][0]
__________________________________________________________________________________________________
activation_87 (Activation)      (None, 19, 19, 128)  0           batch_normalization_38[0][0]
__________________________________________________________________________________________________
average_pooling2d_4 (AveragePoo (None, 19, 19, 768)  0           mixed3[0][0]
__________________________________________________________________________________________________
conv2d_31 (Conv2D)              (None, 19, 19, 192)  147456      mixed3[0][0]
__________________________________________________________________________________________________
conv2d_34 (Conv2D)              (None, 19, 19, 192)  172032      activation_82[0][0]
__________________________________________________________________________________________________
```

```
conv2d_39 (Conv2D)             (None, 19, 19, 192)  172032      activation_87[0][0]
__________________________________________________________________________________________________
conv2d_40 (Conv2D)             (None, 19, 19, 192)  147456      average_pooling2d_4[0][0]
__________________________________________________________________________________________________
batch_normalization_31 (BatchNo (None, 19, 19, 192)  576         conv2d_31[0][0]
__________________________________________________________________________________________________
batch_normalization_34 (BatchNo (None, 19, 19, 192)  576         conv2d_34[0][0]
__________________________________________________________________________________________________
batch_normalization_39 (BatchNo (None, 19, 19, 192)  576         conv2d_39[0][0]
__________________________________________________________________________________________________
batch_normalization_40 (BatchNo (None, 19, 19, 192)  576         conv2d_40[0][0]
__________________________________________________________________________________________________
activation_80 (Activation)     (None, 19, 19, 192)  0           batch_normalization_31[0][0]
__________________________________________________________________________________________________
activation_83 (Activation)     (None, 19, 19, 192)  0           batch_normalization_34[0][0]
__________________________________________________________________________________________________
activation_88 (Activation)     (None, 19, 19, 192)  0           batch_normalization_39[0][0]
__________________________________________________________________________________________________
activation_89 (Activation)     (None, 19, 19, 192)  0           batch_normalization_40[0][0]
__________________________________________________________________________________________________
mixed4 (Concatenate)           (None, 19, 19, 768)  0           activation_80[0][0]
                                                                 activation_83[0][0]
                                                                 activation_88[0][0]
                                                                 activation_89[0][0]
__________________________________________________________________________________________________
conv2d_45 (Conv2D)             (None, 19, 19, 160)  122880      mixed4[0][0]
__________________________________________________________________________________________________
batch_normalization_45 (BatchNo (None, 19, 19, 160)  480         conv2d_45[0][0]
__________________________________________________________________________________________________
activation_94 (Activation)     (None, 19, 19, 160)  0           batch_normalization_45[0][0]
__________________________________________________________________________________________________
conv2d_46 (Conv2D)             (None, 19, 19, 160)  179200      activation_94[0][0]
__________________________________________________________________________________________________
batch_normalization_46 (BatchNo (None, 19, 19, 160)  480         conv2d_46[0][0]
__________________________________________________________________________________________________
activation_95 (Activation)     (None, 19, 19, 160)  0           batch_normalization_46[0][0]
__________________________________________________________________________________________________
conv2d_42 (Conv2D)             (None, 19, 19, 160)  122880      mixed4[0][0]
__________________________________________________________________________________________________
conv2d_47 (Conv2D)             (None, 19, 19, 160)  179200      activation_95[0][0]
__________________________________________________________________________________________________
batch_normalization_42 (BatchNo (None, 19, 19, 160)  480         conv2d_42[0][0]
__________________________________________________________________________________________________
batch_normalization_47 (BatchNo (None, 19, 19, 160)  480         conv2d_47[0][0]
__________________________________________________________________________________________________
activation_91 (Activation)     (None, 19, 19, 160)  0           batch_normalization_42[0][0]
__________________________________________________________________________________________________
activation_96 (Activation)     (None, 19, 19, 160)  0           batch_normalization_47[0][0]
```

```
__________________________________________________________________________________________________
conv2d_43 (Conv2D)              (None, 19, 19, 160)  179200      activation_91[0][0]
__________________________________________________________________________________________________
conv2d_48 (Conv2D)              (None, 19, 19, 160)  179200      activation_96[0][0]
__________________________________________________________________________________________________
batch_normalization_43 (BatchNo (None, 19, 19, 160)  480         conv2d_43[0][0]
__________________________________________________________________________________________________
batch_normalization_48 (BatchNo (None, 19, 19, 160)  480         conv2d_48[0][0]
__________________________________________________________________________________________________
activation_92 (Activation)      (None, 19, 19, 160)  0           batch_normalization_43[0][0]
__________________________________________________________________________________________________
activation_97 (Activation)      (None, 19, 19, 160)  0           batch_normalization_48[0][0]
__________________________________________________________________________________________________
average_pooling2d_5 (AveragePoo (None, 19, 19, 768)  0           mixed4[0][0]
__________________________________________________________________________________________________
conv2d_41 (Conv2D)              (None, 19, 19, 192)  147456      mixed4[0][0]
__________________________________________________________________________________________________
conv2d_44 (Conv2D)              (None, 19, 19, 192)  215040      activation_92[0][0]
__________________________________________________________________________________________________
conv2d_49 (Conv2D)              (None, 19, 19, 192)  215040      activation_97[0][0]
__________________________________________________________________________________________________
conv2d_50 (Conv2D)              (None, 19, 19, 192)  147456      average_pooling2d_5[0][0]
__________________________________________________________________________________________________
batch_normalization_41 (BatchNo (None, 19, 19, 192)  576         conv2d_41[0][0]
__________________________________________________________________________________________________
batch_normalization_44 (BatchNo (None, 19, 19, 192)  576         conv2d_44[0][0]
__________________________________________________________________________________________________
batch_normalization_49 (BatchNo (None, 19, 19, 192)  576         conv2d_49[0][0]
__________________________________________________________________________________________________
batch_normalization_50 (BatchNo (None, 19, 19, 192)  576         conv2d_50[0][0]
__________________________________________________________________________________________________
activation_90 (Activation)      (None, 19, 19, 192)  0           batch_normalization_41[0][0]
__________________________________________________________________________________________________
activation_93 (Activation)      (None, 19, 19, 192)  0           batch_normalization_44[0][0]
__________________________________________________________________________________________________
activation_98 (Activation)      (None, 19, 19, 192)  0           batch_normalization_49[0][0]
__________________________________________________________________________________________________
activation_99 (Activation)      (None, 19, 19, 192)  0           batch_normalization_50[0][0]
__________________________________________________________________________________________________
mixed5 (Concatenate)            (None, 19, 19, 768)  0           activation_90[0][0]
                                                                 activation_93[0][0]
                                                                 activation_98[0][0]
                                                                 activation_99[0][0]
__________________________________________________________________________________________________
conv2d_55 (Conv2D)              (None, 19, 19, 160)  122880      mixed5[0][0]
__________________________________________________________________________________________________
batch_normalization_55 (BatchNo (None, 19, 19, 160)  480         conv2d_55[0][0]
__________________________________________________________________________________________________
```

```
activation_104 (Activation)     (None, 19, 19, 160)  0           batch_normalization_55[0][0]
__________________________________________________________________________________________________
conv2d_56 (Conv2D)              (None, 19, 19, 160)  179200      activation_104[0][0]
__________________________________________________________________________________________________
batch_normalization_56 (BatchNo (None, 19, 19, 160)  480         conv2d_56[0][0]
__________________________________________________________________________________________________
activation_105 (Activation)     (None, 19, 19, 160)  0           batch_normalization_56[0][0]
__________________________________________________________________________________________________
conv2d_52 (Conv2D)              (None, 19, 19, 160)  122880      mixed5[0][0]
__________________________________________________________________________________________________
conv2d_57 (Conv2D)              (None, 19, 19, 160)  179200      activation_105[0][0]
__________________________________________________________________________________________________
batch_normalization_52 (BatchNo (None, 19, 19, 160)  480         conv2d_52[0][0]
__________________________________________________________________________________________________
batch_normalization_57 (BatchNo (None, 19, 19, 160)  480         conv2d_57[0][0]
__________________________________________________________________________________________________
activation_101 (Activation)     (None, 19, 19, 160)  0           batch_normalization_52[0][0]
__________________________________________________________________________________________________
activation_106 (Activation)     (None, 19, 19, 160)  0           batch_normalization_57[0][0]
__________________________________________________________________________________________________
conv2d_53 (Conv2D)              (None, 19, 19, 160)  179200      activation_101[0][0]
__________________________________________________________________________________________________
conv2d_58 (Conv2D)              (None, 19, 19, 160)  179200      activation_106[0][0]
__________________________________________________________________________________________________
batch_normalization_53 (BatchNo (None, 19, 19, 160)  480         conv2d_53[0][0]
__________________________________________________________________________________________________
batch_normalization_58 (BatchNo (None, 19, 19, 160)  480         conv2d_58[0][0]
__________________________________________________________________________________________________
activation_102 (Activation)     (None, 19, 19, 160)  0           batch_normalization_53[0][0]
__________________________________________________________________________________________________
activation_107 (Activation)     (None, 19, 19, 160)  0           batch_normalization_58[0][0]
__________________________________________________________________________________________________
average_pooling2d_6 (AveragePoo (None, 19, 19, 768)  0           mixed5[0][0]
__________________________________________________________________________________________________
conv2d_51 (Conv2D)              (None, 19, 19, 192)  147456      mixed5[0][0]
__________________________________________________________________________________________________
conv2d_54 (Conv2D)              (None, 19, 19, 192)  215040      activation_102[0][0]
__________________________________________________________________________________________________
conv2d_59 (Conv2D)              (None, 19, 19, 192)  215040      activation_107[0][0]
__________________________________________________________________________________________________
conv2d_60 (Conv2D)              (None, 19, 19, 192)  147456      average_pooling2d_6[0][0]
__________________________________________________________________________________________________
batch_normalization_51 (BatchNo (None, 19, 19, 192)  576         conv2d_51[0][0]
__________________________________________________________________________________________________
batch_normalization_54 (BatchNo (None, 19, 19, 192)  576         conv2d_54[0][0]
__________________________________________________________________________________________________
batch_normalization_59 (BatchNo (None, 19, 19, 192)  576         conv2d_59[0][0]
__________________________________________________________________________________________________
```

```
batch_normalization_60 (BatchNo (None, 19, 19, 192)  576         conv2d_60[0][0]
__________________________________________________________________________________________________
activation_100 (Activation)     (None, 19, 19, 192)  0           batch_normalization_51[0][0]
__________________________________________________________________________________________________
activation_103 (Activation)     (None, 19, 19, 192)  0           batch_normalization_54[0][0]
__________________________________________________________________________________________________
activation_108 (Activation)     (None, 19, 19, 192)  0           batch_normalization_59[0][0]
__________________________________________________________________________________________________
activation_109 (Activation)     (None, 19, 19, 192)  0           batch_normalization_60[0][0]
__________________________________________________________________________________________________
mixed6 (Concatenate)            (None, 19, 19, 768)  0           activation_100[0][0]
                                                                 activation_103[0][0]
                                                                 activation_108[0][0]
                                                                 activation_109[0][0]
__________________________________________________________________________________________________
conv2d_65 (Conv2D)              (None, 19, 19, 192)  147456      mixed6[0][0]
__________________________________________________________________________________________________
batch_normalization_65 (BatchNo (None, 19, 19, 192)  576         conv2d_65[0][0]
__________________________________________________________________________________________________
activation_114 (Activation)     (None, 19, 19, 192)  0           batch_normalization_65[0][0]
__________________________________________________________________________________________________
conv2d_66 (Conv2D)              (None, 19, 19, 192)  258048      activation_114[0][0]
__________________________________________________________________________________________________
batch_normalization_66 (BatchNo (None, 19, 19, 192)  576         conv2d_66[0][0]
__________________________________________________________________________________________________
activation_115 (Activation)     (None, 19, 19, 192)  0           batch_normalization_66[0][0]
__________________________________________________________________________________________________
conv2d_62 (Conv2D)              (None, 19, 19, 192)  147456      mixed6[0][0]
__________________________________________________________________________________________________
conv2d_67 (Conv2D)              (None, 19, 19, 192)  258048      activation_115[0][0]
__________________________________________________________________________________________________
batch_normalization_62 (BatchNo (None, 19, 19, 192)  576         conv2d_62[0][0]
__________________________________________________________________________________________________
batch_normalization_67 (BatchNo (None, 19, 19, 192)  576         conv2d_67[0][0]
__________________________________________________________________________________________________
activation_111 (Activation)     (None, 19, 19, 192)  0           batch_normalization_62[0][0]
__________________________________________________________________________________________________
activation_116 (Activation)     (None, 19, 19, 192)  0           batch_normalization_67[0][0]
__________________________________________________________________________________________________
conv2d_63 (Conv2D)              (None, 19, 19, 192)  258048      activation_111[0][0]
__________________________________________________________________________________________________
conv2d_68 (Conv2D)              (None, 19, 19, 192)  258048      activation_116[0][0]
__________________________________________________________________________________________________
batch_normalization_63 (BatchNo (None, 19, 19, 192)  576         conv2d_63[0][0]
__________________________________________________________________________________________________
batch_normalization_68 (BatchNo (None, 19, 19, 192)  576         conv2d_68[0][0]
__________________________________________________________________________________________________
activation_112 (Activation)     (None, 19, 19, 192)  0           batch_normalization_63[0][0]
```

```
______________________________________________________________________________________________
activation_117 (Activation)     (None, 19, 19, 192)  0           batch_normalization_68[0][0]

______________________________________________________________________________________________
average_pooling2d_7 (AveragePoo (None, 19, 19, 768)  0           mixed6[0][0]

______________________________________________________________________________________________
conv2d_61 (Conv2D)              (None, 19, 19, 192)  147456      mixed6[0][0]

______________________________________________________________________________________________
conv2d_64 (Conv2D)              (None, 19, 19, 192)  258048      activation_112[0][0]

______________________________________________________________________________________________
conv2d_69 (Conv2D)              (None, 19, 19, 192)  258048      activation_117[0][0]

______________________________________________________________________________________________
conv2d_70 (Conv2D)              (None, 19, 19, 192)  147456      average_pooling2d_7[0][0]

______________________________________________________________________________________________
batch_normalization_61 (BatchNo (None, 19, 19, 192)  576         conv2d_61[0][0]

______________________________________________________________________________________________
batch_normalization_64 (BatchNo (None, 19, 19, 192)  576         conv2d_64[0][0]

______________________________________________________________________________________________
batch_normalization_69 (BatchNo (None, 19, 19, 192)  576         conv2d_69[0][0]

______________________________________________________________________________________________
batch_normalization_70 (BatchNo (None, 19, 19, 192)  576         conv2d_70[0][0]

______________________________________________________________________________________________
activation_110 (Activation)     (None, 19, 19, 192)  0           batch_normalization_61[0][0]

______________________________________________________________________________________________
activation_113 (Activation)     (None, 19, 19, 192)  0           batch_normalization_64[0][0]

______________________________________________________________________________________________
activation_118 (Activation)     (None, 19, 19, 192)  0           batch_normalization_69[0][0]

______________________________________________________________________________________________
activation_119 (Activation)     (None, 19, 19, 192)  0           batch_normalization_70[0][0]

______________________________________________________________________________________________
mixed7 (Concatenate)            (None, 19, 19, 768)  0           activation_110[0][0]
                                                                 activation_113[0][0]
                                                                 activation_118[0][0]
                                                                 activation_119[0][0]

______________________________________________________________________________________________
conv2d_73 (Conv2D)              (None, 19, 19, 192)  147456      mixed7[0][0]

______________________________________________________________________________________________
batch_normalization_73 (BatchNo (None, 19, 19, 192)  576         conv2d_73[0][0]

______________________________________________________________________________________________
activation_122 (Activation)     (None, 19, 19, 192)  0           batch_normalization_73[0][0]

______________________________________________________________________________________________
conv2d_74 (Conv2D)              (None, 19, 19, 192)  258048      activation_122[0][0]

______________________________________________________________________________________________
batch_normalization_74 (BatchNo (None, 19, 19, 192)  576         conv2d_74[0][0]

______________________________________________________________________________________________
activation_123 (Activation)     (None, 19, 19, 192)  0           batch_normalization_74[0][0]

______________________________________________________________________________________________
conv2d_71 (Conv2D)              (None, 19, 19, 192)  147456      mixed7[0][0]

______________________________________________________________________________________________
```

```
conv2d_75 (Conv2D)              (None, 19, 19, 192)  258048      activation_123[0][0]
-----------------------------------------------------------------------------------------------
batch_normalization_71 (BatchNo (None, 19, 19, 192)  576         conv2d_71[0][0]
-----------------------------------------------------------------------------------------------
batch_normalization_75 (BatchNo (None, 19, 19, 192)  576         conv2d_75[0][0]
-----------------------------------------------------------------------------------------------
activation_120 (Activation)     (None, 19, 19, 192)  0           batch_normalization_71[0][0]
-----------------------------------------------------------------------------------------------
activation_124 (Activation)     (None, 19, 19, 192)  0           batch_normalization_75[0][0]
-----------------------------------------------------------------------------------------------
conv2d_72 (Conv2D)              (None, 9, 9, 320)    552960      activation_120[0][0]
-----------------------------------------------------------------------------------------------
conv2d_76 (Conv2D)              (None, 9, 9, 192)    331776      activation_124[0][0]
-----------------------------------------------------------------------------------------------
batch_normalization_72 (BatchNo (None, 9, 9, 320)    960         conv2d_72[0][0]
-----------------------------------------------------------------------------------------------
batch_normalization_76 (BatchNo (None, 9, 9, 192)    576         conv2d_76[0][0]
-----------------------------------------------------------------------------------------------
activation_121 (Activation)     (None, 9, 9, 320)    0           batch_normalization_72[0][0]
-----------------------------------------------------------------------------------------------
activation_125 (Activation)     (None, 9, 9, 192)    0           batch_normalization_76[0][0]
-----------------------------------------------------------------------------------------------
max_pooling2d_5 (MaxPooling2D)  (None, 9, 9, 768)    0           mixed7[0][0]
-----------------------------------------------------------------------------------------------
mixed8 (Concatenate)            (None, 9, 9, 1280)   0           activation_121[0][0]
                                                                 activation_125[0][0]
                                                                 max_pooling2d_5[0][0]
-----------------------------------------------------------------------------------------------
conv2d_81 (Conv2D)              (None, 9, 9, 448)    573440      mixed8[0][0]
-----------------------------------------------------------------------------------------------
batch_normalization_81 (BatchNo (None, 9, 9, 448)    1344        conv2d_81[0][0]
-----------------------------------------------------------------------------------------------
activation_130 (Activation)     (None, 9, 9, 448)    0           batch_normalization_81[0][0]
-----------------------------------------------------------------------------------------------
conv2d_78 (Conv2D)              (None, 9, 9, 384)    491520      mixed8[0][0]
-----------------------------------------------------------------------------------------------
conv2d_82 (Conv2D)              (None, 9, 9, 384)    1548288     activation_130[0][0]
-----------------------------------------------------------------------------------------------
batch_normalization_78 (BatchNo (None, 9, 9, 384)    1152        conv2d_78[0][0]
-----------------------------------------------------------------------------------------------
batch_normalization_82 (BatchNo (None, 9, 9, 384)    1152        conv2d_82[0][0]
-----------------------------------------------------------------------------------------------
activation_127 (Activation)     (None, 9, 9, 384)    0           batch_normalization_78[0][0]
-----------------------------------------------------------------------------------------------
activation_131 (Activation)     (None, 9, 9, 384)    0           batch_normalization_82[0][0]
-----------------------------------------------------------------------------------------------
conv2d_79 (Conv2D)              (None, 9, 9, 384)    442368      activation_127[0][0]
-----------------------------------------------------------------------------------------------
```

```
conv2d_80 (Conv2D)              (None, 9, 9, 384)    442368      activation_127[0][0]
_______________________________________________________________________________________________
conv2d_83 (Conv2D)              (None, 9, 9, 384)    442368      activation_131[0][0]
_______________________________________________________________________________________________
conv2d_84 (Conv2D)              (None, 9, 9, 384)    442368      activation_131[0][0]
_______________________________________________________________________________________________
average_pooling2d_8 (AveragePoo (None, 9, 9, 1280)   0           mixed8[0][0]
_______________________________________________________________________________________________
conv2d_77 (Conv2D)              (None, 9, 9, 320)    409600      mixed8[0][0]
_______________________________________________________________________________________________
batch_normalization_79 (BatchNo (None, 9, 9, 384)    1152        conv2d_79[0][0]
_______________________________________________________________________________________________
batch_normalization_80 (BatchNo (None, 9, 9, 384)    1152        conv2d_80[0][0]
_______________________________________________________________________________________________
batch_normalization_83 (BatchNo (None, 9, 9, 384)    1152        conv2d_83[0][0]
_______________________________________________________________________________________________
batch_normalization_84 (BatchNo (None, 9, 9, 384)    1152        conv2d_84[0][0]
_______________________________________________________________________________________________
conv2d_85 (Conv2D)              (None, 9, 9, 192)    245760      average_pooling2d_8[0][0]
_______________________________________________________________________________________________
batch_normalization_77 (BatchNo (None, 9, 9, 320)    960         conv2d_77[0][0]
_______________________________________________________________________________________________
activation_128 (Activation)     (None, 9, 9, 384)    0           batch_normalization_79[0][0]
_______________________________________________________________________________________________
activation_129 (Activation)     (None, 9, 9, 384)    0           batch_normalization_80[0][0]
_______________________________________________________________________________________________
activation_132 (Activation)     (None, 9, 9, 384)    0           batch_normalization_83[0][0]
_______________________________________________________________________________________________
activation_133 (Activation)     (None, 9, 9, 384)    0           batch_normalization_84[0][0]
_______________________________________________________________________________________________
batch_normalization_85 (BatchNo (None, 9, 9, 192)    576         conv2d_85[0][0]
_______________________________________________________________________________________________
activation_126 (Activation)     (None, 9, 9, 320)    0           batch_normalization_77[0][0]
_______________________________________________________________________________________________
mixed9_0 (Concatenate)          (None, 9, 9, 768)    0           activation_128[0][0]
                                                                 activation_129[0][0]
_______________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, 9, 9, 768)    0           activation_132[0][0]
                                                                 activation_133[0][0]
_______________________________________________________________________________________________
activation_134 (Activation)     (None, 9, 9, 192)    0           batch_normalization_85[0][0]
_______________________________________________________________________________________________
mixed9 (Concatenate)            (None, 9, 9, 2048)   0           activation_126[0][0]
                                                                 mixed9_0[0][0]
                                                                 concatenate_1[0][0]
                                                                 activation_134[0][0]
_______________________________________________________________________________________________
conv2d_90 (Conv2D)              (None, 9, 9, 448)    917504      mixed9[0][0]
```

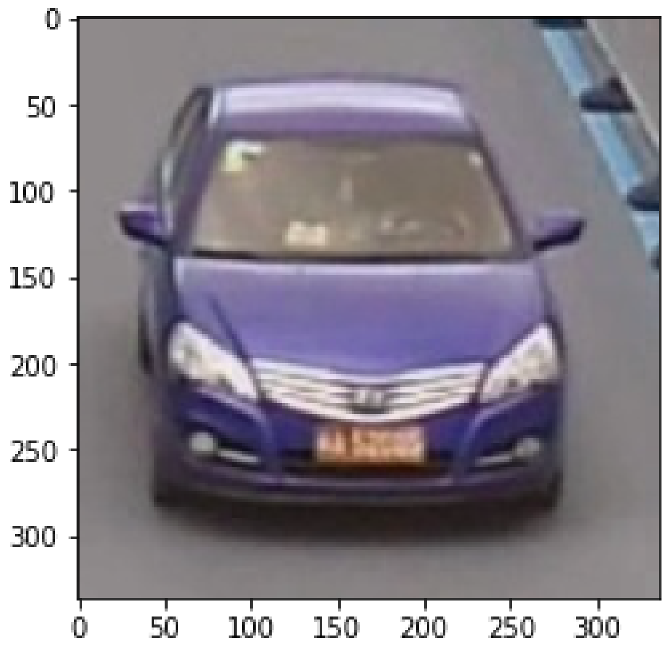
```
____________________________________________________________________________________________________
batch_normalization_90 (BatchNo (None, 9, 9, 448)    1344        conv2d_90[0][0]
____________________________________________________________________________________________________
activation_139 (Activation)     (None, 9, 9, 448)    0           batch_normalization_90[0][0]
____________________________________________________________________________________________________
conv2d_87 (Conv2D)              (None, 9, 9, 384)    786432      mixed9[0][0]
____________________________________________________________________________________________________
conv2d_91 (Conv2D)              (None, 9, 9, 384)    1548288     activation_139[0][0]
____________________________________________________________________________________________________
batch_normalization_87 (BatchNo (None, 9, 9, 384)    1152        conv2d_87[0][0]
____________________________________________________________________________________________________
batch_normalization_91 (BatchNo (None, 9, 9, 384)    1152        conv2d_91[0][0]
____________________________________________________________________________________________________
activation_136 (Activation)     (None, 9, 9, 384)    0           batch_normalization_87[0][0]
____________________________________________________________________________________________________
activation_140 (Activation)     (None, 9, 9, 384)    0           batch_normalization_91[0][0]
____________________________________________________________________________________________________
conv2d_88 (Conv2D)              (None, 9, 9, 384)    442368      activation_136[0][0]
____________________________________________________________________________________________________
conv2d_89 (Conv2D)              (None, 9, 9, 384)    442368      activation_136[0][0]
____________________________________________________________________________________________________
conv2d_92 (Conv2D)              (None, 9, 9, 384)    442368      activation_140[0][0]
____________________________________________________________________________________________________
conv2d_93 (Conv2D)              (None, 9, 9, 384)    442368      activation_140[0][0]
____________________________________________________________________________________________________
average_pooling2d_9 (AveragePoo (None, 9, 9, 2048)   0           mixed9[0][0]
____________________________________________________________________________________________________
conv2d_86 (Conv2D)              (None, 9, 9, 320)    655360      mixed9[0][0]
____________________________________________________________________________________________________
batch_normalization_88 (BatchNo (None, 9, 9, 384)    1152        conv2d_88[0][0]
____________________________________________________________________________________________________
batch_normalization_89 (BatchNo (None, 9, 9, 384)    1152        conv2d_89[0][0]
____________________________________________________________________________________________________
batch_normalization_92 (BatchNo (None, 9, 9, 384)    1152        conv2d_92[0][0]
____________________________________________________________________________________________________
batch_normalization_93 (BatchNo (None, 9, 9, 384)    1152        conv2d_93[0][0]
____________________________________________________________________________________________________
conv2d_94 (Conv2D)              (None, 9, 9, 192)    393216      average_pooling2d_9[0][0]
____________________________________________________________________________________________________
batch_normalization_86 (BatchNo (None, 9, 9, 320)    960         conv2d_86[0][0]
____________________________________________________________________________________________________
activation_137 (Activation)     (None, 9, 9, 384)    0           batch_normalization_88[0][0]
____________________________________________________________________________________________________
activation_138 (Activation)     (None, 9, 9, 384)    0           batch_normalization_89[0][0]
____________________________________________________________________________________________________
activation_141 (Activation)     (None, 9, 9, 384)    0           batch_normalization_92[0][0]
____________________________________________________________________________________________________
activation_142 (Activation)     (None, 9, 9, 384)    0           batch_normalization_93[0][0]
```

```
__________________________________________________________________________________________________
batch_normalization_94 (BatchNo (None, 9, 9, 192)    576         conv2d_94[0][0]
__________________________________________________________________________________________________
activation_135 (Activation)     (None, 9, 9, 320)    0           batch_normalization_86[0][0]
__________________________________________________________________________________________________
mixed9_1 (Concatenate)          (None, 9, 9, 768)    0           activation_137[0][0]
                                                                 activation_138[0][0]
__________________________________________________________________________________________________
concatenate_2 (Concatenate)     (None, 9, 9, 768)    0           activation_141[0][0]
                                                                 activation_142[0][0]
__________________________________________________________________________________________________
activation_143 (Activation)     (None, 9, 9, 192)    0           batch_normalization_94[0][0]
__________________________________________________________________________________________________
mixed10 (Concatenate)           (None, 9, 9, 2048)   0           activation_135[0][0]
                                                                 mixed9_1[0][0]
                                                                 concatenate_2[0][0]
                                                                 activation_143[0][0]
__________________________________________________________________________________________________
global_average_pooling2d_1 (Glo (None, 2048)         0           mixed10[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 512)          1049088     global_average_pooling2d_1[0]
__________________________________________________________________________________________________
dense_2 (Dense)                 (None, 512)          262656      dense_1[0][0]
__________________________________________________________________________________________________
dense_3 (Dense)                 (None, 8)            4104        dense_2[0][0]
==================================================================================================
Total params: 23,118,632
Trainable params: 1,315,848
Non-trainable params: 21,802,784
__________________________________________________________________________________________________
```

```
In [34]: from keras.preprocessing.image import ImageDataGenerator

         BATCH_SIZE = 32
         train_datagen = ImageDataGenerator(
                                            rotation_range=0,
                                            width_shift_range=0.2,
                                            height_shift_range=0.2,
                                            horizontal_flip = 'false')
         train_generator = train_datagen.flow(X_train_new, y_train1, shuffle=False,
                                              batch_size=BATCH_SIZE, seed=1)
         batch_size = BATCH_SIZE
         train_steps_per_epoch = X_train_new.shape[0] // batch_size
         #val_steps_per_epoch = x_val.shape[0] // batch_size

         history = model.fit_generator(train_generator,
                                       steps_per_epoch=train_steps_per_epoch,
                                              epochs=15, verbose=1)

W1130 23:14:17.456909 140373928044352 deprecation.py:323] From /home/william/anaconda3/envs/tf
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where


Epoch 1/15
15/15 [==============================] - 66s 4s/step - loss: 1.9972 - acc: 0.2208
Epoch 2/15
15/15 [==============================] - 57s 4s/step - loss: 1.7412 - acc: 0.2945
Epoch 3/15
15/15 [==============================] - 59s 4s/step - loss: 1.6464 - acc: 0.3605
Epoch 4/15
15/15 [==============================] - 57s 4s/step - loss: 1.5584 - acc: 0.4535
Epoch 5/15
15/15 [==============================] - 57s 4s/step - loss: 1.4642 - acc: 0.5021
Epoch 6/15
15/15 [==============================] - 58s 4s/step - loss: 1.4143 - acc: 0.5501
Epoch 7/15
15/15 [==============================] - 56s 4s/step - loss: 1.3564 - acc: 0.5230
Epoch 8/15
15/15 [==============================] - 57s 4s/step - loss: 1.2873 - acc: 0.5515
Epoch 9/15
15/15 [==============================] - 57s 4s/step - loss: 1.2444 - acc: 0.5556
Epoch 10/15
15/15 [==============================] - 58s 4s/step - loss: 1.1553 - acc: 0.6224
Epoch 11/15
15/15 [==============================] - 57s 4s/step - loss: 1.1491 - acc: 0.5744
Epoch 12/15
15/15 [==============================] - 57s 4s/step - loss: 1.1275 - acc: 0.5780
Epoch 13/15
15/15 [==============================] - 56s 4s/step - loss: 1.0410 - acc: 0.6232
Epoch 14/15
15/15 [==============================] - 58s 4s/step - loss: 0.9893 - acc: 0.6733
Epoch 15/15
15/15 [==============================] - 57s 4s/step - loss: 1.0144 - acc: 0.6490
```

```
In [48]: im = string2image(fully_annotated[:][0][9])
         im = cv2.resize(im,(336,336))
         plt.imshow(im)
         im = np.expand_dims(im, axis =0)
         img = X_train_new[0:2]#.reshape(1, -1)
         #img1 = preprocess_input(X_train_new)
         #print(img1.shape)
         #prediction = model.predict(img1)
         predictions = model.predict(im)
         predictions
```

```
Out[48]: array([[4.8218969e-19, 5.7204437e-27, 0.0000000e+00, 9.9999201e-01,
                 7.6653373e-15, 2.9607233e-30, 9.5904443e-25, 8.0103518e-06]],
               dtype=float32)
```



In this case, the pipelining process is, how to say this, not very spectacular. Let's move on and train a Keras model! We will start with a simple model, as found in this blog post with a fully connected layer and 100 hidden units.

```
In [41]: from keras.models import Sequential
         from keras.layers import BatchNormalization, Conv2D, Activation, MaxPooling2D, Dense,

Using TensorFlow backend.
```

```
In [42]: model = Sequential()
         model.add(Dense(100, activation="relu", input_shape=(336*336,)))
         model.add(Activation('relu'))
         model.add(Dense(8))

WARNING: Logging before flag parsing goes to stderr.
W1129 20:54:58.370776 139950289872704 deprecation_wrapper.py:119] From /home/william/anaconda3/

W1129 20:54:58.394842 139950289872704 deprecation_wrapper.py:119] From /home/william/anaconda3/

W1129 20:54:58.398083 139950289872704 deprecation_wrapper.py:119] From /home/william/anaconda3/
```

Now let's compile the model and run the training.

```
In [43]: from keras import optimizers

         sgd = optimizers.SGD(lr=1e-5, decay=1e-4, momentum=0.9, nesterov=True)
         adam = optimizers.Adam(lr=10e-3, beta_1=0.9, beta_2=0.999, amsgrad=True)

         model.compile(optimizer=sgd, loss='mse', metrics=['accuracy'])
         epochs = 100
         history = model.fit(X_train.reshape(y_train.shape[0], -1), y_train,
                             validation_split=0.2, shuffle=True,
                             epochs=epochs, batch_size=20)
```

```
W1129 20:55:00.433584 139950289872704 deprecation_wrapper.py:119] From /home/william/anaconda3/

W1129 20:55:00.620710 139950289872704 deprecation_wrapper.py:119] From /home/william/anaconda3/

W1129 20:55:00.650210 139950289872704 deprecation_wrapper.py:119] From /home/william/anaconda3/



Train on 399 samples, validate on 100 samples
Epoch 1/100
399/399 [==============================] - 3s 7ms/step - loss: 0.6493 - acc: 0.0677 - val_loss
Epoch 2/100
399/399 [==============================] - 2s 5ms/step - loss: 0.5099 - acc: 0.0000e+00 - val_l
Epoch 3/100
399/399 [==============================] - 2s 5ms/step - loss: 0.4775 - acc: 0.0025 - val_loss
Epoch 4/100
399/399 [==============================] - 2s 6ms/step - loss: 0.4643 - acc: 0.0050 - val_loss
Epoch 5/100
399/399 [==============================] - 2s 5ms/step - loss: 0.4546 - acc: 0.0025 - val_loss
Epoch 6/100
399/399 [==============================] - 2s 5ms/step - loss: 0.4476 - acc: 0.0025 - val_loss
Epoch 7/100
399/399 [==============================] - 2s 5ms/step - loss: 0.4425 - acc: 0.0000e+00 - val_l
Epoch 8/100
399/399 [==============================] - 2s 5ms/step - loss: 0.4386 - acc: 0.0000e+00 - val_l
Epoch 9/100
399/399 [==============================] - 2s 5ms/step - loss: 0.4357 - acc: 0.0025 - val_loss
Epoch 10/100
399/399 [==============================] - 2s 5ms/step - loss: 0.4331 - acc: 0.0050 - val_loss
Epoch 11/100
399/399 [==============================] - 2s 5ms/step - loss: 0.4310 - acc: 0.0025 - val_loss
Epoch 12/100
399/399 [==============================] - 2s 5ms/step - loss: 0.4289 - acc: 0.0075 - val_loss
Epoch 13/100
399/399 [==============================] - 2s 5ms/step - loss: 0.4273 - acc: 0.0050 - val_loss
```

```
Epoch 14/100
399/399 [==============================] - 2s 5ms/step - loss: 0.4252 - acc: 0.0050 - val_loss
Epoch 15/100
399/399 [==============================] - 2s 5ms/step - loss: 0.4235 - acc: 0.0075 - val_loss
Epoch 16/100
399/399 [==============================] - 2s 5ms/step - loss: 0.4218 - acc: 0.0050 - val_loss
Epoch 17/100
399/399 [==============================] - 2s 5ms/step - loss: 0.4200 - acc: 0.0050 - val_loss
Epoch 18/100
399/399 [==============================] - 2s 5ms/step - loss: 0.4181 - acc: 0.0050 - val_loss
Epoch 19/100
399/399 [==============================] - 2s 5ms/step - loss: 0.4163 - acc: 0.0075 - val_loss
Epoch 20/100
399/399 [==============================] - 2s 5ms/step - loss: 0.4147 - acc: 0.0025 - val_loss
Epoch 21/100
399/399 [==============================] - 2s 5ms/step - loss: 0.4128 - acc: 0.0050 - val_loss
Epoch 22/100
399/399 [==============================] - 2s 5ms/step - loss: 0.4115 - acc: 0.0050 - val_loss
Epoch 23/100
399/399 [==============================] - 2s 5ms/step - loss: 0.4099 - acc: 0.0050 - val_loss
Epoch 24/100
399/399 [==============================] - 2s 5ms/step - loss: 0.4080 - acc: 0.0025 - val_loss
Epoch 25/100
399/399 [==============================] - 2s 6ms/step - loss: 0.4067 - acc: 0.0050 - val_loss
Epoch 26/100
399/399 [==============================] - 2s 6ms/step - loss: 0.4053 - acc: 0.0050 - val_loss
Epoch 27/100
399/399 [==============================] - 2s 6ms/step - loss: 0.4036 - acc: 0.0050 - val_loss
Epoch 28/100
399/399 [==============================] - 2s 5ms/step - loss: 0.4022 - acc: 0.0050 - val_loss
Epoch 29/100
399/399 [==============================] - 2s 6ms/step - loss: 0.4005 - acc: 0.0100 - val_loss
Epoch 30/100
399/399 [==============================] - 2s 6ms/step - loss: 0.3986 - acc: 0.0100 - val_loss
Epoch 31/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3969 - acc: 0.0125 - val_loss
Epoch 32/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3955 - acc: 0.0125 - val_loss
Epoch 33/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3941 - acc: 0.0125 - val_loss
Epoch 34/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3929 - acc: 0.0201 - val_loss
Epoch 35/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3915 - acc: 0.0175 - val_loss
Epoch 36/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3904 - acc: 0.0251 - val_loss
Epoch 37/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3892 - acc: 0.0301 - val_loss
```

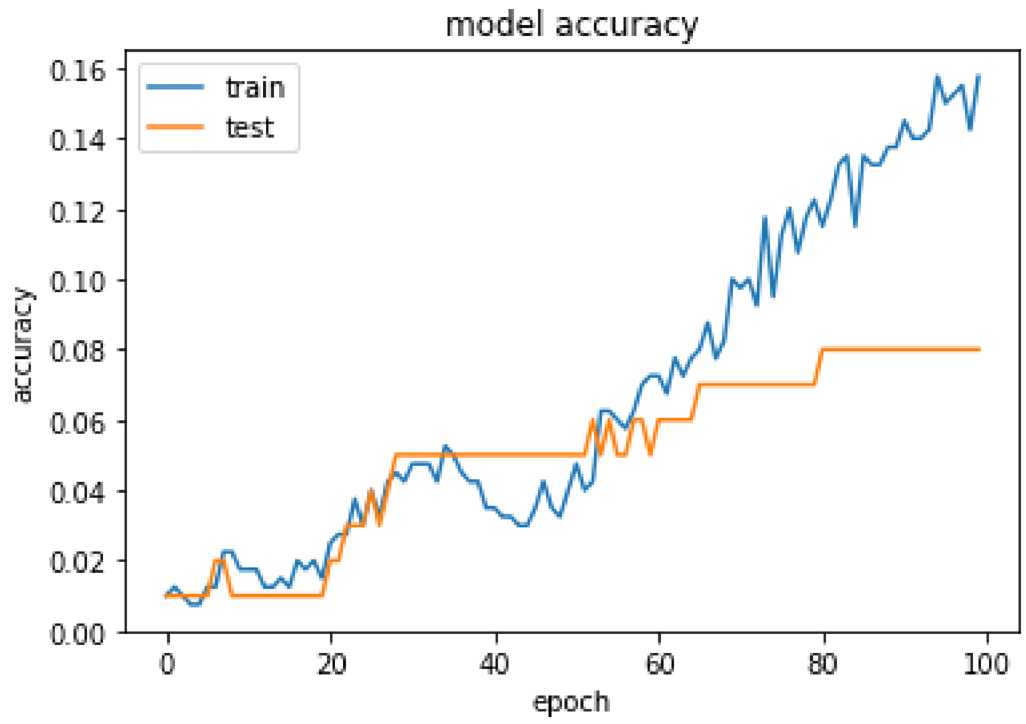
```
Epoch 38/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3883 - acc: 0.0351 - val_loss
Epoch 39/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3867 - acc: 0.0376 - val_loss
Epoch 40/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3855 - acc: 0.0301 - val_loss
Epoch 41/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3845 - acc: 0.0401 - val_loss
Epoch 42/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3834 - acc: 0.0501 - val_loss
Epoch 43/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3823 - acc: 0.0476 - val_loss
Epoch 44/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3811 - acc: 0.0426 - val_loss
Epoch 45/100
399/399 [==============================] - 2s 6ms/step - loss: 0.3802 - acc: 0.0526 - val_loss
Epoch 46/100
399/399 [==============================] - 2s 6ms/step - loss: 0.3794 - acc: 0.0526 - val_loss
Epoch 47/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3781 - acc: 0.0602 - val_loss
Epoch 48/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3773 - acc: 0.0627 - val_loss
Epoch 49/100
399/399 [==============================] - 2s 6ms/step - loss: 0.3762 - acc: 0.0576 - val_loss
Epoch 50/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3752 - acc: 0.0627 - val_loss
Epoch 51/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3745 - acc: 0.0702 - val_loss
Epoch 52/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3732 - acc: 0.0602 - val_loss
Epoch 53/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3726 - acc: 0.0702 - val_loss
Epoch 54/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3719 - acc: 0.0777 - val_loss
Epoch 55/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3710 - acc: 0.0752 - val_loss
Epoch 56/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3700 - acc: 0.0677 - val_loss
Epoch 57/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3692 - acc: 0.0752 - val_loss
Epoch 58/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3682 - acc: 0.0777 - val_loss
Epoch 59/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3675 - acc: 0.0727 - val_loss
Epoch 60/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3666 - acc: 0.0752 - val_loss
Epoch 61/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3658 - acc: 0.0727 - val_loss
```

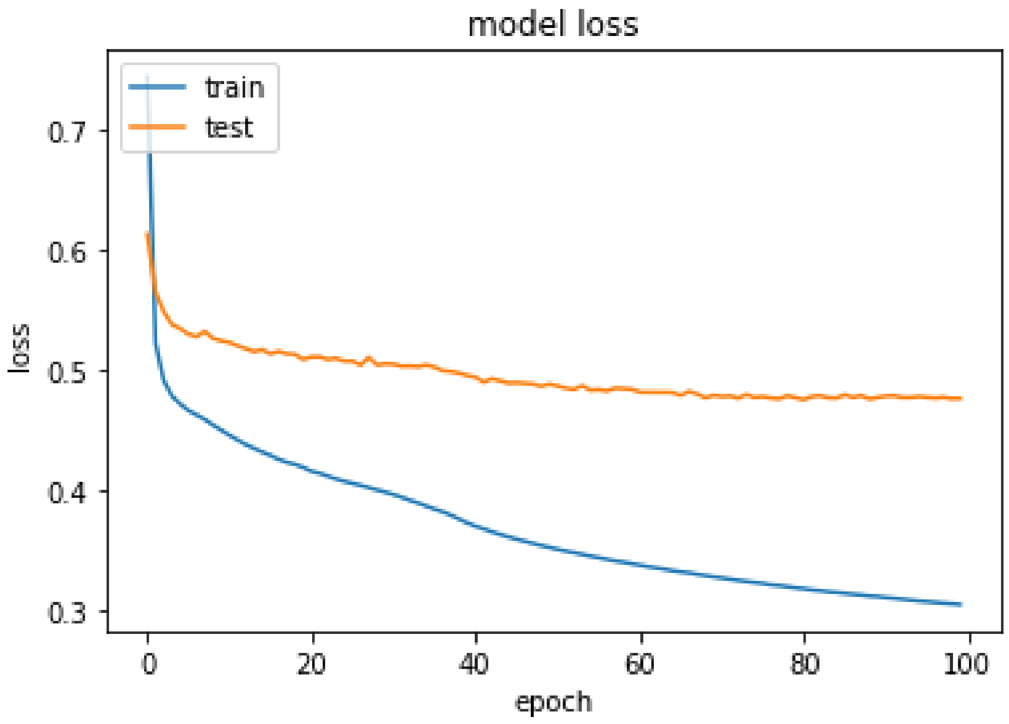
```
Epoch 62/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3651 - acc: 0.0802 - val_loss
Epoch 63/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3644 - acc: 0.0802 - val_loss
Epoch 64/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3635 - acc: 0.0752 - val_loss
Epoch 65/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3628 - acc: 0.0802 - val_loss
Epoch 66/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3621 - acc: 0.0752 - val_loss
Epoch 67/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3616 - acc: 0.0802 - val_loss
Epoch 68/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3607 - acc: 0.0702 - val_loss
Epoch 69/100
399/399 [==============================] - 2s 4ms/step - loss: 0.3600 - acc: 0.0827 - val_loss
Epoch 70/100
399/399 [==============================] - 2s 4ms/step - loss: 0.3592 - acc: 0.0777 - val_loss
Epoch 71/100
399/399 [==============================] - 2s 4ms/step - loss: 0.3587 - acc: 0.0752 - val_loss
Epoch 72/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3578 - acc: 0.0852 - val_loss
Epoch 73/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3572 - acc: 0.0877 - val_loss
Epoch 74/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3568 - acc: 0.0877 - val_loss
Epoch 75/100
399/399 [==============================] - 2s 4ms/step - loss: 0.3559 - acc: 0.0902 - val_loss
Epoch 76/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3551 - acc: 0.0927 - val_loss
Epoch 77/100
399/399 [==============================] - 2s 4ms/step - loss: 0.3546 - acc: 0.0902 - val_loss
Epoch 78/100
399/399 [==============================] - 2s 4ms/step - loss: 0.3539 - acc: 0.0802 - val_loss
Epoch 79/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3532 - acc: 0.0952 - val_loss
Epoch 80/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3526 - acc: 0.0952 - val_loss
Epoch 81/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3519 - acc: 0.0877 - val_loss
Epoch 82/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3513 - acc: 0.0852 - val_loss
Epoch 83/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3506 - acc: 0.0927 - val_loss
Epoch 84/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3500 - acc: 0.0977 - val_loss
Epoch 85/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3495 - acc: 0.1028 - val_loss
```

```
Epoch 86/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3488 - acc: 0.0927 - val_loss
Epoch 87/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3483 - acc: 0.0927 - val_loss
Epoch 88/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3476 - acc: 0.0977 - val_loss
Epoch 89/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3472 - acc: 0.1003 - val_loss
Epoch 90/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3466 - acc: 0.1028 - val_loss
Epoch 91/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3459 - acc: 0.0952 - val_loss
Epoch 92/100
399/399 [==============================] - 2s 4ms/step - loss: 0.3455 - acc: 0.1053 - val_loss
Epoch 93/100
399/399 [==============================] - 2s 4ms/step - loss: 0.3450 - acc: 0.1078 - val_loss
Epoch 94/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3444 - acc: 0.1078 - val_loss
Epoch 95/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3438 - acc: 0.0902 - val_loss
Epoch 96/100
399/399 [==============================] - 2s 4ms/step - loss: 0.3432 - acc: 0.1128 - val_loss
Epoch 97/100
399/399 [==============================] - 2s 4ms/step - loss: 0.3426 - acc: 0.1003 - val_loss
Epoch 98/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3423 - acc: 0.1028 - val_loss
Epoch 99/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3414 - acc: 0.0977 - val_loss
Epoch 100/100
399/399 [==============================] - 2s 5ms/step - loss: 0.3410 - acc: 0.1003 - val_loss
```

Let's plot our training curves with this model.

```
In [30]: # 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()
```



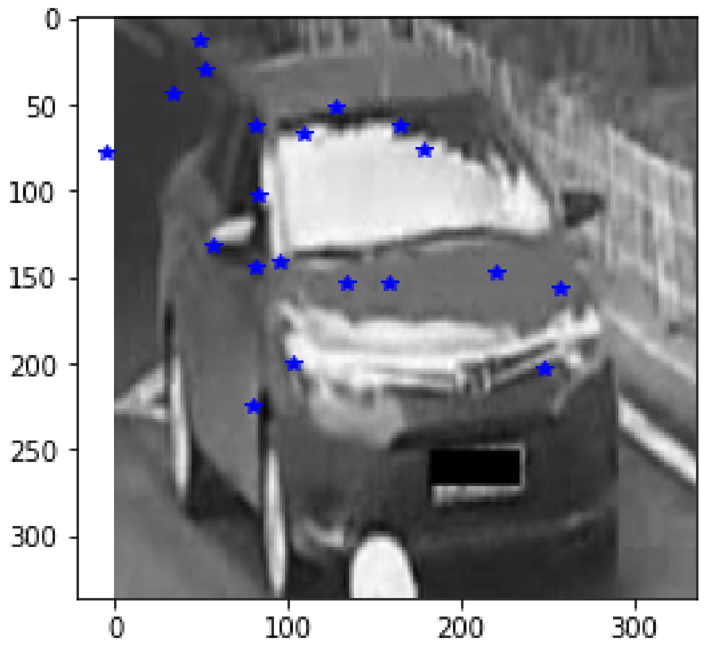


What we see here is that with this model, the learning quickly gets on a plateau. How can we improve this? There are a lot of options:

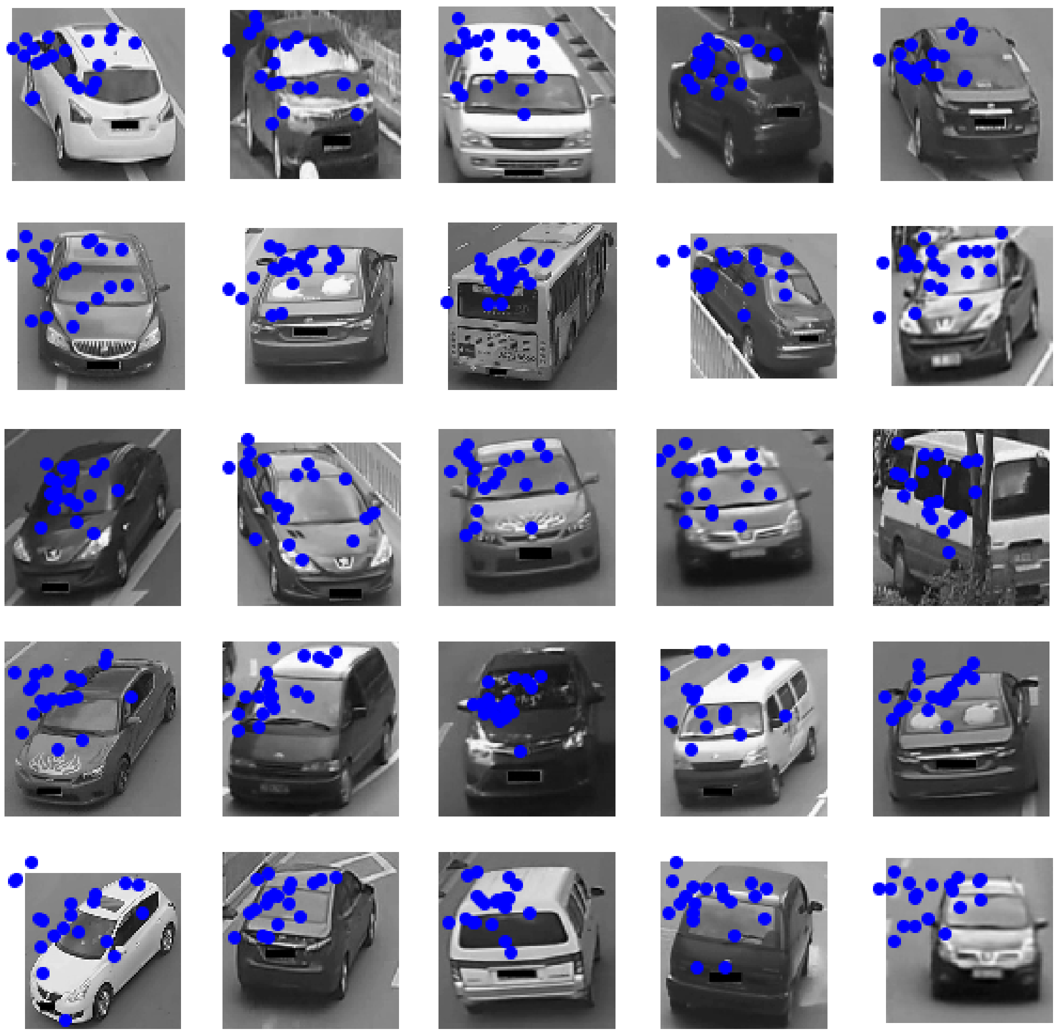
- adjust the optimizer settings
    - learning rate
    - batch size
    - momentum
- change the model

However, one things that is pretty clear from the above plot is that our model overfits: the train and test losses are not comparable (the test loss is 3 times higher). Let's see what the results of the net are on some samples from our data.

```
In [35]: img = X_train[2, :, :, :].reshape(1, -1)
         predictions = model.predict(img)

In [36]: img

Out[36]: array([[0.34117647, 0.34117647, 0.34509804, ..., 0.44313725, 0.44705882,
                 0.45098039]])

In [37]: xy_predictions = output_pipe.inverse_transform(predictions).reshape(20, 2)

In [38]: plt.imshow(X_train[2, :, :, 0], cmap='gray')
         plt.plot(xy_predictions[:, 0], xy_predictions[:, 1], 'b*')

Out[38]: [<matplotlib.lines.Line2D at 0x7f7bac22f940>]
```



```
In [39]: def plot_faces_with_keypoints_and_predictions(model, nrows=5, ncols=5, model_input='fl
             """Plots sampled faces with their truth and predictions."""
             selection = np.random.choice(np.arange(X.shape[0]), size=(nrows*ncols), replace=Fa
             fig, axes = plt.subplots(figsize=(10, 10), nrows=nrows, ncols=ncols)
             for ind, ax in zip(selection, axes.ravel()):
                 img = X_train[ind, :, :, 0]
                 if model_input == 'flat':
                     predictions = model.predict(img.reshape(1, -1))
                 else:
                     predictions = model.predict(img[np.newaxis, :, :, np.newaxis])
                 xy_predictions = output_pipe.inverse_transform(predictions).reshape(20, 2)
                 ax.imshow(img, cmap='gray')
                 ax.plot(xy_predictions[:, 0], xy_predictions[:, 1], 'bo')
                 ax.axis('off')

In [40]: plot_faces_with_keypoints_and_predictions(model)
```



Actually, this looks pretty good already. Let's try to train a more complicated model, this time following the initial model description found in Peter Skvarenina's article.

5 Towards more complicated models

```
In [41]: from keras.layers import Dropout, Flatten

In [43]: import math, json, os, sys

         import keras
         from keras.callbacks import EarlyStopping, ModelCheckpoint
         from keras.layers import Dense
         from keras.models import Model
         from keras.optimizers import Adam
         from keras.preprocessing import image


         DATA_DIR = 'VeRi'
         TRAIN_DIR = os.path.join(DATA_DIR, 'image_train')
         VALID_DIR = os.path.join(DATA_DIR, 'image_test')
         SIZE = (336, 336)
         BATCH_SIZE = 16



         num_train_samples = sum([500 for r, d, files in os.walk(TRAIN_DIR)])
         num_valid_samples = sum([50 for r, d, files in os.walk(VALID_DIR)])

         num_train_steps = math.floor(num_train_samples/BATCH_SIZE)
         num_valid_steps = math.floor(num_valid_samples/BATCH_SIZE)

         gen = keras.preprocessing.image.ImageDataGenerator()
         val_gen = keras.preprocessing.image.ImageDataGenerator(horizontal_flip=True, vertical_

         batches = gen.flow_from_directory(TRAIN_DIR, target_size=SIZE, class_mode='categorical
         val_batches = val_gen.flow_from_directory(VALID_DIR, target_size=SIZE, class_mode='ca

         model = keras.applications.resnet50.ResNet50()

         classes = list(iter(batches.class_indices))
         model.layers.pop()
         for layer in model.layers:
             layer.trainable=False
         last = model.layers[-1].output
         x = Dense(len(classes), activation="softmax")(last)
         finetuned_model = Model(model.input, x)
         finetuned_model.compile(optimizer=Adam(lr=0.0001), loss='categorical_crossentropy', me
         for c in batches.class_indices:
             classes[batches.class_indices[c]] = c
         finetuned_model.classes = classes

         early_stopping = EarlyStopping(patience=10)
         checkpointer = ModelCheckpoint('resnet50_best.h5', verbose=1, save_best_only=True)

         finetuned_model.fit_generator(batches, steps_per_epoch=num_train_steps, epochs=1000, c
         finetuned_model.save('resnet50_final.h5')
```

```
Found 0 images belonging to 0 classes.
Found 0 images belonging to 0 classes.
Downloading data from https://github.com/fchollet/deep-learning-models/releases/download/v0.2/r
102858752/102853048 [==============================] - 9s 0us/step
```

```
    ---------------------------------------------------------------------------

    AttributeError                            Traceback (most recent call last)

    <ipython-input-43-bf498f084cf9> in <module>
     46 checkpointer = ModelCheckpoint('resnet50_best.h5', verbose=1, save_best_only=True)
     47
---> 48 finetuned_model.fit_generator(batches, steps_per_epoch=num_train_steps, epochs=100
     49 finetuned_model.save('resnet50_final.h5')


    ~/anaconda3/envs/tfgpu/lib/python3.6/site-packages/keras/legacy/interfaces.py in wrapp
     89                 warnings.warn('Update your `' + object_name + '` call to the ' +
     90                               'Keras 2 API: ' + signature, stacklevel=2)
---> 91             return func(*args, **kwargs)
     92         wrapper._original_function = func
     93         return wrapper


    ~/anaconda3/envs/tfgpu/lib/python3.6/site-packages/keras/engine/training.py in fit_gen
   1416             use_multiprocessing=use_multiprocessing,
   1417             shuffle=shuffle,
-> 1418             initial_epoch=initial_epoch)
   1419
   1420     @interfaces.legacy_generator_methods_support


    ~/anaconda3/envs/tfgpu/lib/python3.6/site-packages/keras/engine/training_generator.py 
     38
     39     do_validation = bool(validation_data)
---> 40     model._make_train_function()
     41     if do_validation:
     42         model._make_test_function()


    ~/anaconda3/envs/tfgpu/lib/python3.6/site-packages/keras/engine/training.py in _make_t
    507                     training_updates = self.optimizer.get_updates(
    508                         params=self._collected_trainable_weights,
--> 509                         loss=self.total_loss)
    510                 updates = (self.updates +
    511                            training_updates +


    ~/anaconda3/envs/tfgpu/lib/python3.6/site-packages/keras/legacy/interfaces.py in wrapp
     89                 warnings.warn('Update your `' + object_name + '` call to the ' +
     90                               'Keras 2 API: ' + signature, stacklevel=2)
---> 91             return func(*args, **kwargs)
```

```
     92         wrapper._original_function = func
     93         return wrapper


    ~/anaconda3/envs/tfgpu/lib/python3.6/site-packages/keras/optimizers.py in get_updates(
    503                 p_t = p - lr_t * m_t / (K.sqrt(v_t) + self.epsilon)
    504 
--> 505             self.updates.append(K.update(m, m_t))
    506             self.updates.append(K.update(v, v_t))
    507             new_p = p_t


    ~/anaconda3/envs/tfgpu/lib/python3.6/site-packages/keras/backend/tensorflow_backend.py
    971         The variable `x` updated.
    972     """
--> 973     return tf.assign(x, new_x)
    974 
    975 


    ~/anaconda3/envs/tfgpu/lib/python3.6/site-packages/tensorflow/python/ops/state_ops.py
    220         ref, value, use_locking=use_locking, name=name,
    221         validate_shape=validate_shape)
--> 222   return ref.assign(value, name=name)
    223 
    224 


    AttributeError: 'Tensor' object has no attribute 'assign'
```

In [60]:
```
model = Sequential()
# input layer
model.add(BatchNormalization(input_shape=(336, 336, 1)))
model.add(Conv2D(24, (5, 5), kernel_initializer='he_normal'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Dropout(0.2))
# layer 2
model.add(Conv2D(36, (5, 5)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Dropout(0.2))
# layer 3
model.add(Conv2D(48, (5, 5)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Dropout(0.2))
```

```
        # layer 4
        model.add(Conv2D(64, (3, 3)))
        model.add(Activation('relu'))
        model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
        model.add(Dropout(0.2))
        # layer 5
        model.add(Conv2D(64, (3, 3)))
        model.add(Activation('relu'))
        model.add(Flatten())
        # layer 6
        model.add(Dense(500, activation="relu"))
        # layer 7
        model.add(Dense(120, activation="relu"))
        # layer 8
        model.add(Dense(40))
```

```
In [61]: sgd = optimizers.SGD(lr=1e-6, decay=1e-6, momentum=0.95, nesterov=True)
         model.compile(optimizer=sgd, loss='mse', metrics=['accuracy'])
         epochs = 110
         history = model.fit(X_train, y_train,
                             validation_split=0.2, shuffle=True,
                             epochs=epochs, batch_size=10)
```

```
Train on 400 samples, validate on 100 samples
Epoch 1/110
400/400 [==============================] - 49s 123ms/step - loss: 0.8674 - acc: 0.0150 - val_lo
Epoch 2/110
400/400 [==============================] - 56s 141ms/step - loss: 0.8432 - acc: 0.0200 - val_lo
Epoch 3/110
400/400 [==============================] - 51s 126ms/step - loss: 0.8003 - acc: 0.0100 - val_lo
Epoch 4/110
400/400 [==============================] - 51s 127ms/step - loss: 0.7700 - acc: 0.0100 - val_lo
Epoch 5/110
400/400 [==============================] - 51s 127ms/step - loss: 0.7482 - acc: 0.0225 - val_lo
Epoch 6/110
400/400 [==============================] - 50s 126ms/step - loss: 0.7278 - acc: 0.0175 - val_lo
Epoch 7/110
400/400 [==============================] - 50s 124ms/step - loss: 0.7063 - acc: 0.0175 - val_lo
Epoch 8/110
400/400 [==============================] - 49s 121ms/step - loss: 0.6998 - acc: 0.0175 - val_lo
Epoch 9/110
400/400 [==============================] - 51s 127ms/step - loss: 0.6803 - acc: 0.0075 - val_lo
Epoch 10/110
400/400 [==============================] - 49s 123ms/step - loss: 0.6744 - acc: 0.0150 - val_lo
Epoch 11/110
340/400 [========================>...] - ETA: 6s - loss: 0.6633 - acc: 0.0088
```

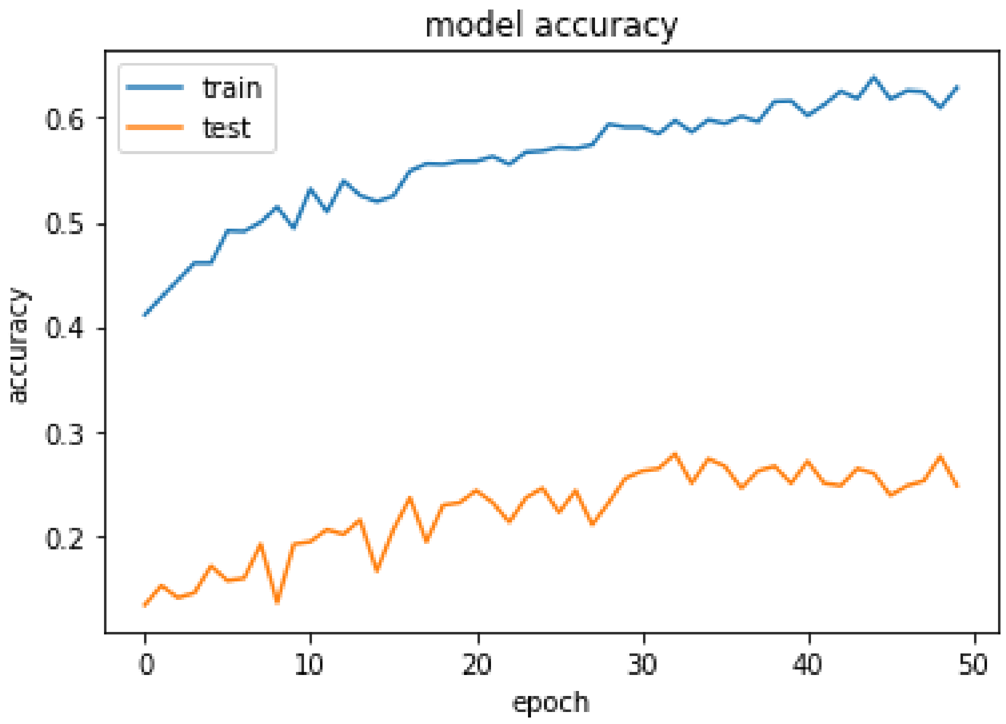
```
        ---------------------------------------------------------------------------
```

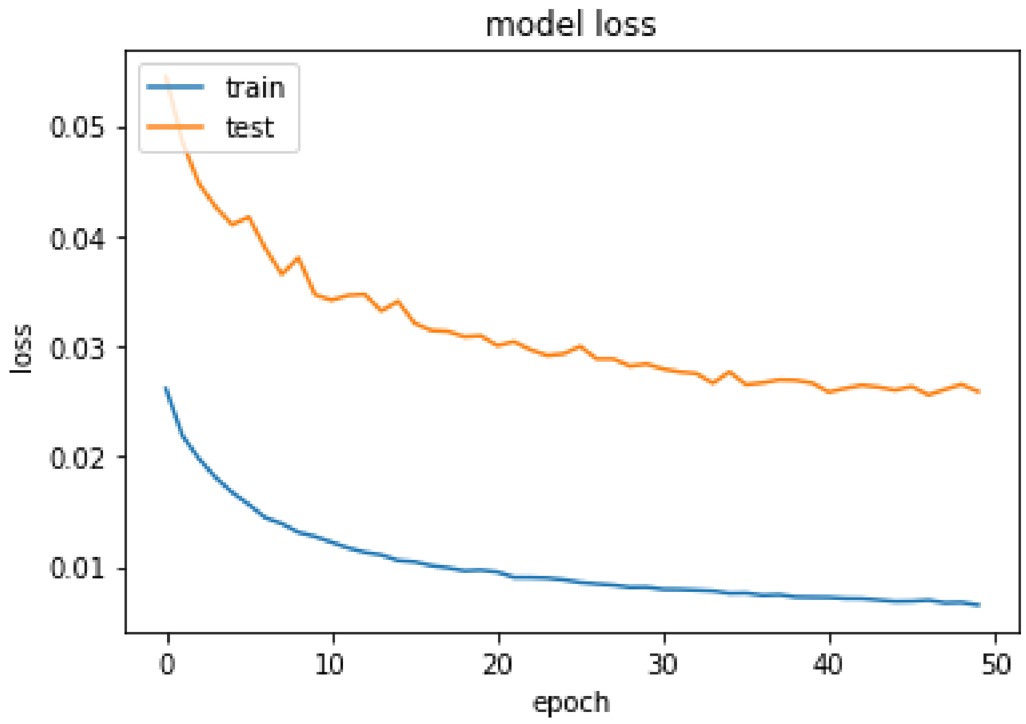
```
    KeyboardInterrupt                         Traceback (most recent call last)

    <ipython-input-61-63ad94e2956e> in <module>
      4 history = model.fit(X_train, y_train,
      5                   validation_split=0.2, shuffle=True,
----> 6                   epochs=epochs, batch_size=10)


    ~/anaconda3/envs/tfcpu/lib/python3.6/site-packages/keras/engine/training.py in fit(sel
   1037                                         initial_epoch=initial_epoch,
   1038                                         steps_per_epoch=steps_per_epoch,
-> 1039                                         validation_steps=validation_steps)
   1040
   1041     def evaluate(self, x=None, y=None,


    ~/anaconda3/envs/tfcpu/lib/python3.6/site-packages/keras/engine/training_arrays.py in
    197                         ins_batch[i] = ins_batch[i].toarray()
    198
--> 199                 outs = f(ins_batch)
    200                 outs = to_list(outs)
    201                 for l, o in zip(out_labels, outs):


    ~/anaconda3/envs/tfcpu/lib/python3.6/site-packages/keras/backend/tensorflow_backend.py
   2713                 return self._legacy_call(inputs)
   2714
-> 2715             return self._call(inputs)
   2716         else:
   2717             if py_any(is_tensor(x) for x in inputs):


    ~/anaconda3/envs/tfcpu/lib/python3.6/site-packages/keras/backend/tensorflow_backend.py
   2673             fetched = self._callable_fn(*array_vals, run_metadata=self.run_metadat
   2674         else:
-> 2675             fetched = self._callable_fn(*array_vals)
   2676         return fetched[:len(self.outputs)]
   2677


    ~/anaconda3/envs/tfcpu/lib/python3.6/site-packages/tensorflow/python/client/session.py
   1456         ret = tf_session.TF_SessionRunCallable(self._session._session,
   1457                                                self._handle, args,
-> 1458                                                run_metadata_ptr)
   1459         if run_metadata:
   1460           proto_data = tf_session.TF_GetBuffer(run_metadata_ptr)
```

```
KeyboardInterrupt:
```

Let's see that in curves:

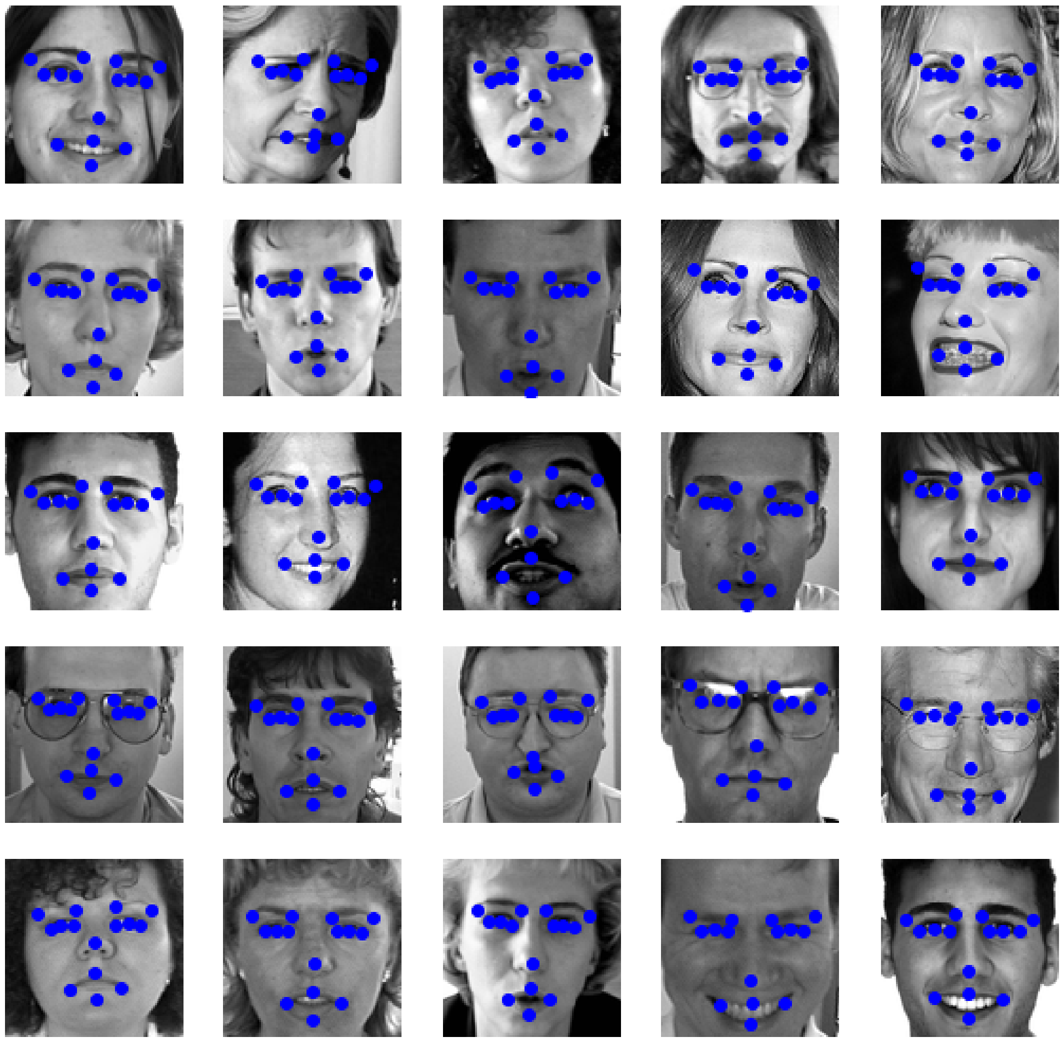
```
In [118]: # summarize history for accuracy
          plt.plot(history.history['acc'])
          plt.plot(history.history['val_acc'])
          plt.title('model accuracy')
          plt.ylabel('accuracy')
          plt.xlabel('epoch')
          plt.legend(['train', 'test'], loc='upper left')
          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()
```





How good is the result?

```
In [120]: plot_faces_with_keypoints_and_predictions(model, model_input='2d')
```



If you ask me, that's already pretty good. Even though we didn't reach the performance advertised in Peter Skvarenina's blog post, with 80% validation accuracy. I wonder what he used to reach that level of performance: longer training? better settings?

Let's move on to the last section of this blog post: applications.

6 Applications

6.1 A face mask

A first thing we can do is to apply some sort of mask on top of the detected image. Let's draw a moustache over an image for example.

First, we need an image of a moustache.

```
In [175]: import skimage.color
          from skimage.filters import median
```

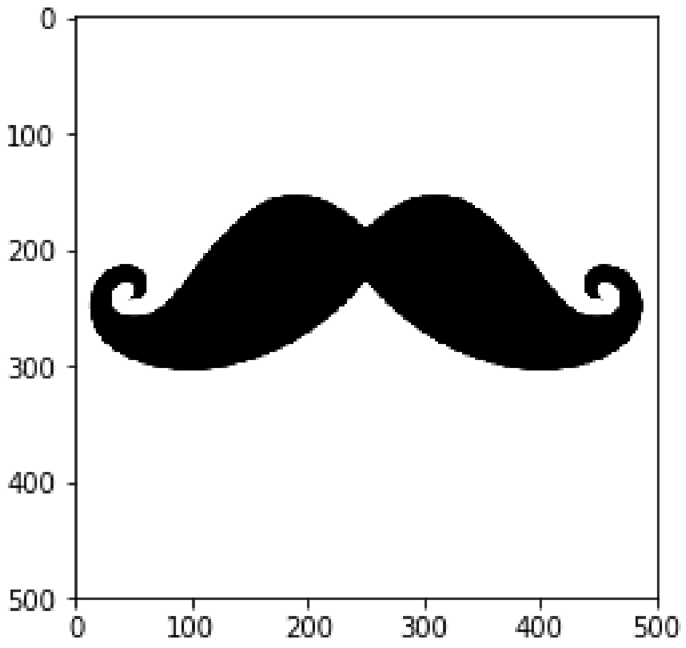
```
In [337]: moustache = plt.imread('http://www.freeiconspng.com/uploads/moustache-png-by-spoonswa
          moustache = skimage.color.rgb2gray(moustache)
```

```
In [338]: moustache = median(moustache, selem=np.ones((3, 3)))
```

```
/Users/kappamaki/anaconda/lib/python3.6/site-packages/skimage/util/dtype.py:122: UserWarning: P
  .format(dtypeobj_in, dtypeobj_out))
```

Let's display it.

```
In [339]: plt.imshow(moustache, cmap='gray')

Out[339]: <matplotlib.image.AxesImage at 0x1388b2198>
```



Now, let's extract the boundary of this moustache.

```
In [340]: from skimage import measure
          moustache_contour = measure.find_contours(moustache, 0.8)[0]
          moustache_contour -= np.array([250, 250])
```

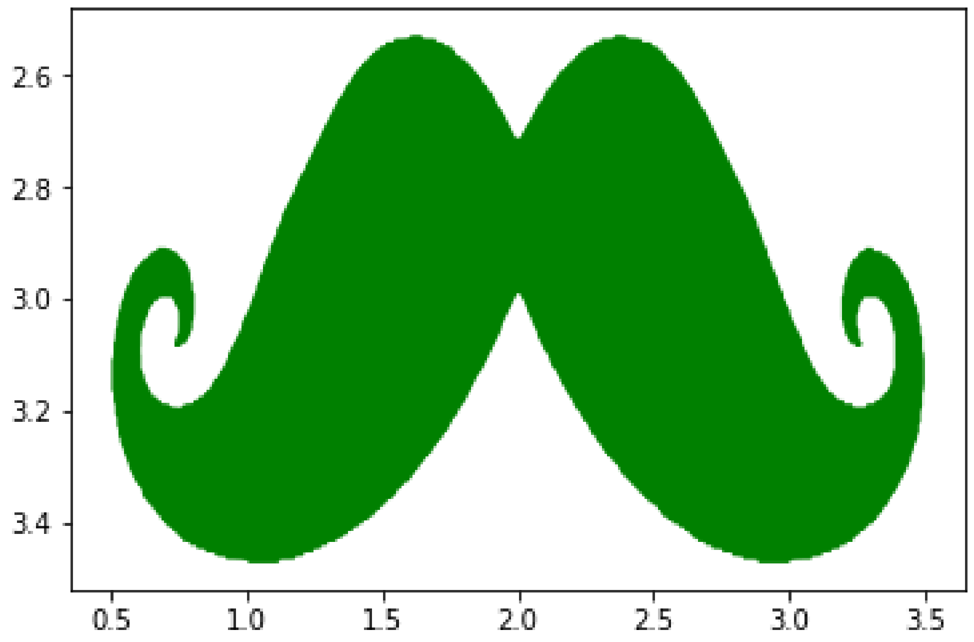
Now, let's write a function that plots a scaled moustache at a given position.

```
In [368]: def plot_scaled_moustache(ax, center_xy, dx):
              """Plots a moustache scaled by its width, dx, on current ax."""
              moustache_scaled = moustache_contour.copy()
              moustache_scaled -= moustache_contour.min(axis=0)
              moustache_scaled /= moustache_scaled.max(axis=0)[1]
```

```
        deltas = moustache_scaled.max(axis=0) - moustache_scaled.min(axis=0)
        moustache_scaled -= np.array([deltas[0]/2, deltas[1]/2])
        moustache_scaled *= dx
        moustache_scaled += center_xy[::-1]
        ax.fill(moustache_scaled[:, 1], moustache_scaled[:, 0], "g", linewidth=4)
```

Let's test this:

```
In [369]: ax = plt.gca()
          plot_scaled_moustache(ax, np.array([2, 3]), dx=3)
          ax.invert_yaxis()
```



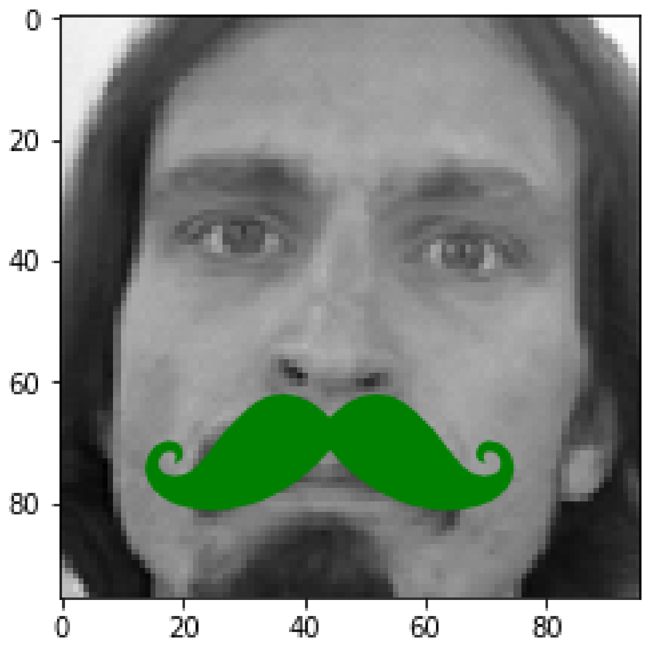
Finally, we can integrate this with a function of the predicted points. We will use the mouth location and space the moustache using the size of the mouth.

```
In [370]: def draw_moustache(predicted_points, ax):
              """Draws a moustache using the predicted face points."""
              dx = 2 * np.linalg.norm(predicted_points[12, :] - predicted_points[11, :])
              center_xy = predicted_points[13, :]
              plot_scaled_moustache(ax, center_xy, dx)
```

Let's try this with the first image from the training set.

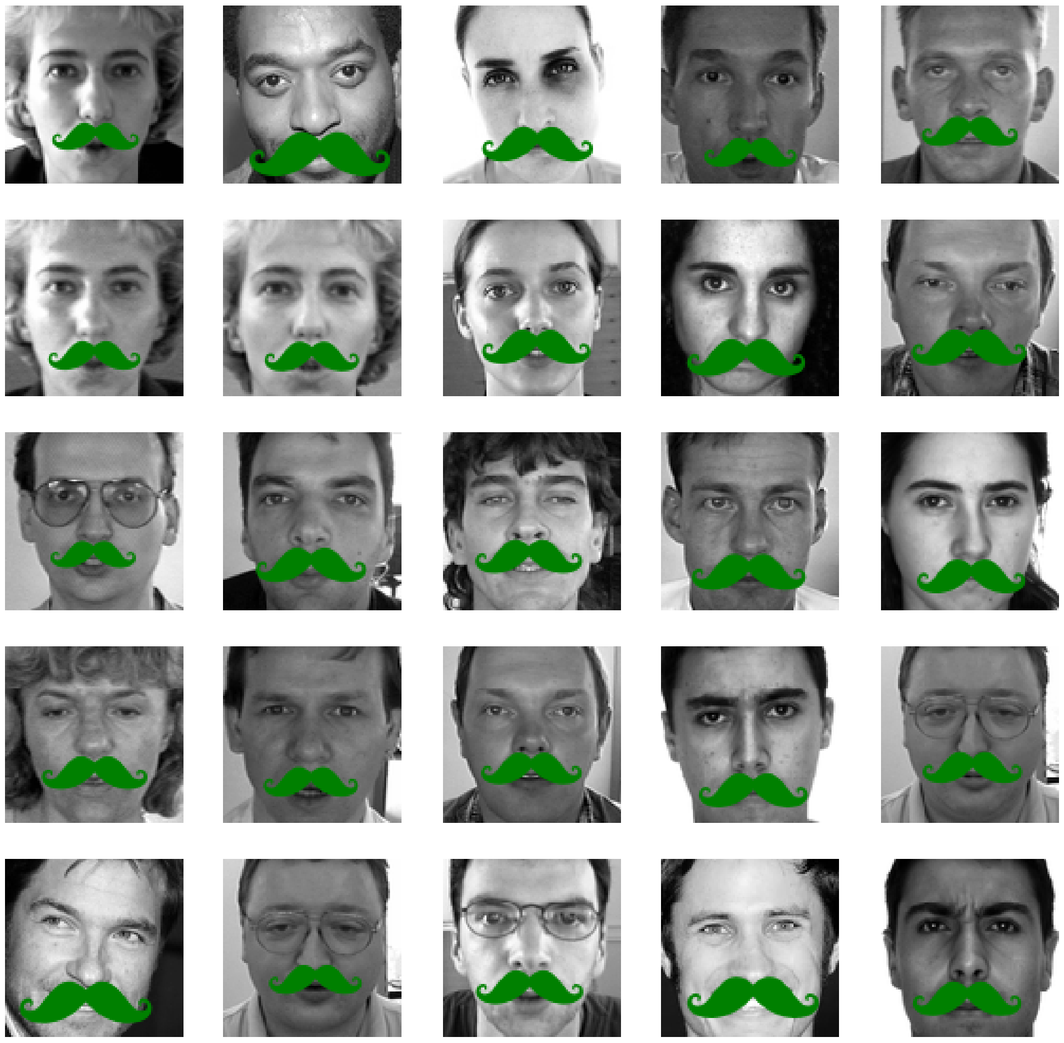
```
In [371]: img = X_train[0, :, :, :][np.newaxis, :, :, :]
          predictions = model.predict(img)
          xy_predictions = output_pipe.inverse_transform(predictions).reshape(15, 2)
```

```
In [372]: fig, ax = plt.subplots()
          ax.imshow(X_train[0, :, :, 0], cmap='gray')
          draw_moustache(xy_predictions, ax)
```



Ok, looks good. Let's apply this to a grid of images.

```
In [373]: def plot_faces_with_moustaches(model, nrows=5, ncols=5, model_input='flat'):
              """Plots sampled faces with their truth and predictions."""
              selection = np.random.choice(np.arange(X.shape[0]), size=(nrows*ncols), replace=
              fig, axes = plt.subplots(figsize=(10, 10), nrows=nrows, ncols=ncols)
              for ind, ax in zip(selection, axes.ravel()):
                  img = X_train[ind, :, :, 0]
                  if model_input == 'flat':
                      predictions = model.predict(img.reshape(1, -1))
                  else:
                      predictions = model.predict(img[np.newaxis, :, :, np.newaxis])
                  xy_predictions = output_pipe.inverse_transform(predictions).reshape(15, 2)
                  ax.imshow(img, cmap='gray')
                  draw_moustache(xy_predictions, ax)
                  ax.axis('off')

In [375]: plot_faces_with_moustaches(model, model_input='2d')
```



This is fun. There's a couple of ways we could better: adjust for face directions (the tilted faces in particular look strange). But that's already pretty nice. Let's make a gallery of famous faces with moustaches.

6.2 Famous faces with moustaches

Let's apply the skill of adding automated moustaches to some famous paintings.

```
In [501]: portrait_urls = ["https://upload.wikimedia.org/wikipedia/commons/thumb/e/ec/Mona_Lisa
                           "https://upload.wikimedia.org/wikipedia/commons/thumb/d/d2/Hans_Holbe
                           "https://upload.wikimedia.org/wikipedia/commons/b/b6/The_Blue_Boy.jpg
                           "https://upload.wikimedia.org/wikipedia/commons/thumb/2/2f/Thomas_Ker
                           "https://upload.wikimedia.org/wikipedia/en/d/d6/GertrudeStein.JPG",
                           "https://upload.wikimedia.org/wikipedia/commons/thumb/b/b0/Ambrogio_d
```

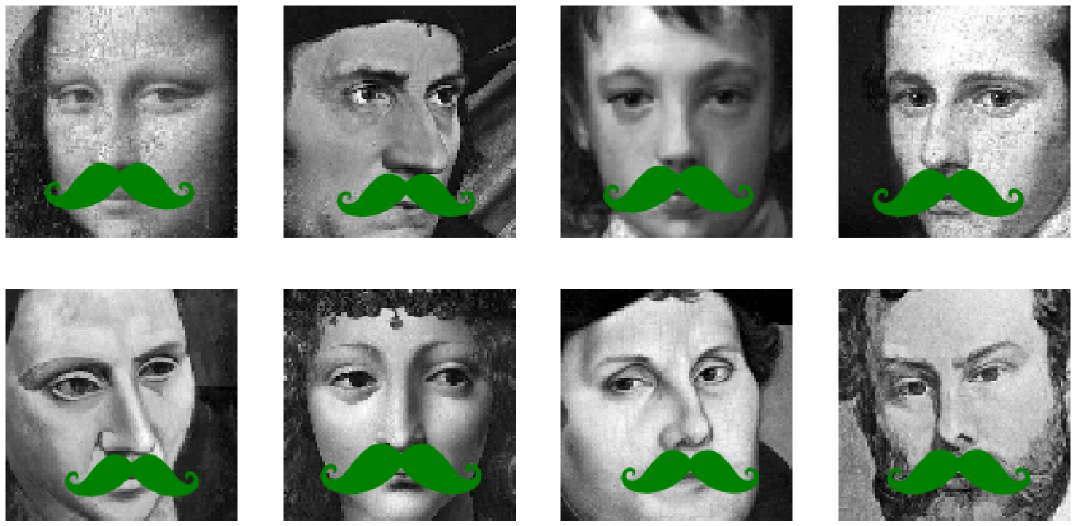
```
                         "https://upload.wikimedia.org/wikipedia/commons/f/f8/Martin_Luther%2
                         "https://upload.wikimedia.org/wikipedia/commons/thumb/6/60/Pierre-Au
```

```
In [502]: portraits = {}
          for url in portrait_urls:
              if url not in portraits:
                  portraits[url] = imread(url)
```

```
In [503]: from skimage.io import imread
          import cv2
```

```
In [505]: face_cascade = cv2.CascadeClassifier('data/haarcascade_frontalface_default.xml')
          fig, axes = plt.subplots(nrows=2, ncols=4, figsize=(12, 6))
          for img, ax in zip(portraits.values(), axes.ravel()):
              gray = (skimage.color.rgb2gray(img) * 255).astype(dtype='uint8')
              bounding_boxes = face_cascade.detectMultiScale(gray, 1.25, 6)
              for (x,y,w,h) in bounding_boxes:
                  roi_gray = gray[y:y+h, x:x+w]
                  roi_rescaled = skimage.transform.resize(roi_gray, (96, 96))
                  predictions = model.predict(roi_rescaled[np.newaxis, :, :, np.newaxis])
                  xy_predictions = output_pipe.inverse_transform(predictions).reshape(15, 2)
                  ax.imshow(roi_rescaled, cmap='gray')
                  draw_moustache(xy_predictions, ax)
              ax.axis('off')
```

```
/Users/kappamaki/anaconda/lib/python3.6/site-packages/skimage/transform/_warps.py:84: UserWarn
  warn("The default mode, 'constant', will be changed to 'reflect' in "
```



For comparison's sake, here are the original paintings:

```
In [506]: face_cascade = cv2.CascadeClassifier('data/haarcascade_frontalface_default.xml')
          fig, axes = plt.subplots(nrows=2, ncols=4, figsize=(12, 6))
          for img, ax in zip(portraits.values(), axes.ravel()):
              ax.imshow(img)
              ax.axis('off')
```

7 Conclusions

Okay, that's it for this blog post. So what steps did we go through? We trained a model using Kaggle data, Keras and a deep convolutional neural network. The model was good enough that we could apply it to images from the internet without major changes.

After doing all this, I still feel that we only scratched the edge of what we could do with this. In particular, the neural network part was not very satisfying since I feel the model we trained could have been better. The reason I did not delve deeper into this (no pun intended) is that I don't own any GPU and hence the training takes quite a long time, which I was not willing to wait for better results.

As a takeaway from this post, I think the claim that a high school genius could do things like these on his own is indeed true. If you have the data, it seems that the machine learning models are powerful and simple enough to allow you to do things that were much more complicated in the past.

If I have time for a next post, I'd love to extend the work we did here but do transfer learning, using features from famous already trained neural networks.