CNN

April 16, 2023

0.1 Train and Deploy a CNN Model Using TensorFlow Serving

[1]:
```
import numpy as np
import pandas as pd
import keras
from tqdm import tqdm
import os
from sklearn.model_selection import train_test_split
import cv2
from PIL import Image
import tensorflow as tf
from matplotlib import pyplot as plt
from keras.layers import Dense, Dropout, Flatten, Input
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.utils import img_to_array, array_to_img, load_img
from keras.preprocessing import image
from keras.utils import plot_model
from keras.models import Model
from keras.layers.convolutional import Conv2D
from keras.layers.pooling import MaxPooling2D
from numpy import array
```

[2]:
```
train = pd.read_csv('trainLabels.csv')
```

[3]:
```
train.tail()
```

[3]:

	image	level
1422	9996_left	0
1423	9996_right	0
1424	9998_left	0
1425	9998_right	0
1426	9999_left	0

[4]:
```
targets_series = pd.Series(train['level'])
one_hot = pd.get_dummies(targets_series, sparse = True)
targets_series[:10]
```

[4]:
```
0    0
1    0
2    0
3    0
4    3
5    3
6    0
7    0
8    0
9    1
Name: level, dtype: int64
```

[5]:
```
one_hot[:10]
```

[5]:

	0	1	2	3	4
0	1	0	0	0	0
1	1	0	0	0	0
2	1	0	0	0	0
3	1	0	0	0	0
4	0	0	0	1	0
5	0	0	0	1	0
6	1	0	0	0	0
7	1	0	0	0	0
8	1	0	0	0	0
9	0	1	0	0	0

[6]:
```
one_hot_labels = np.asarray(one_hot)
one_hot_labelsY = np.asarray(targets_series)
one_hot_labelsY[:10]
```

[6]:
```
array([0, 0, 0, 0, 3, 3, 0, 0, 0, 1], dtype=int64)
```

[7]:
```
im_size1 = 224
im_size2 = 224
x_train = []
y_train = []
```

[8]:
```
i = 0
for f, breed in tqdm(train.values):
    print(f)
```

```
 41%|        | 578/1427 [00:00<00:00, 5308.52it/s]
84_left
84_right
95_left
95_right
99_left
99_right
```

842_left
842_right
844_left
844_right
845_left
845_right
851_left
851_right
857_left
857_right
862_left
862_right
863_left
863_right
865_left
865_right
866_left
866_right
869_left
869_right
876_left
876_right
877_left
877_right
878_left
878_right
879_left
879_right
880_left
880_right
885_left
885_right
886_left
886_right
897_left
897_right
899_left
899_right
904_left
904_right
905_left
905_right
906_left
906_right
909_left
909_right
910_left
910_right

912_left
912_right
914_left
914_right
916_left
916_right
917_left
917_right
921_left
921_right
925_left
925_right
928_left
928_right
929_left
929_right
930_left
930_right
931_left
931_right
932_left
932_right
934_left
934_right
935_left
935_right
936_left
936_right
939_left
939_right
940_left
940_right
941_left
941_right
943_left
943_right
945_left
945_right
947_left
947_right
953_left
953_right
958_left
958_right
959_left
959_right
960_left
960_right

963_left
963_right
966_left
966_right
967_left
967_right
968_left
968_right
969_left
969_right
970_left
970_right
971_left
971_right
977_left
977_right
978_left
978_right
979_left
979_right
980_left
980_right
983_left
983_right
985_left
985_right
986_left
986_right
987_left
987_right
990_left
990_right
992_left
992_right
994_left
994_right
997_left
997_right
998_left
998_right
8421_left
8421_right
8424_left
8424_right
8425_left
8425_right
8426_left
8426_right

8430_left
8430_right
8433_left
8433_right
8434_left
8434_right
8439_left
8439_right
8447_left
8447_right
8449_left
8449_right
8451_left
8451_right
8452_left
8452_right
8453_left
8453_right
8455_left
8455_right
8456_left
8456_right
8462_left
8462_right
8466_left
8466_right
8469_left
8469_right
8470_left
8470_right
8471_left
8471_right
8473_left
8473_right
8477_left
8477_right
8479_left
8479_right
8482_left
8482_right
8484_left
8484_right
8486_left
8486_right
8487_left
8487_right
8490_left
8490_right

8492_left
8492_right
8493_left
8493_right
8494_left
8494_right
8495_left
8495_right
8496_left
8496_right
8498_left
8498_right
8499_left
8499_right
8500_left
8500_right
8503_left
8503_right
8504_left
8504_right
8507_left
8507_right
8511_left
8511_right
8513_left
8513_right
8514_left
8514_right
8515_left
8515_right
8517_left
8517_right
8519_left
8519_right
8522_left
8522_right
8526_left
8526_right
8527_left
8527_right
8531_left
8531_right
8532_left
8532_right
8533_left
8533_right
8538_left
8538_right

8539_left
8539_right
8540_left
8540_right
8541_left
8541_right
8543_left
8543_right
8551_left
8551_right
8553_left
8553_right
8554_left
8554_right
8555_left
8555_right
8556_left
8556_right
8558_left
8558_right
8561_left
8561_right
8562_left
8562_right
8565_left
8565_right
8569_left
8569_right
8576_left
8576_right
8578_left
8578_right
8581_left
8581_right
8583_left
8583_right
8585_left
8585_right
8586_left
8586_right
8588_left
8588_right
8589_left
8589_right
8593_left
8593_right
8594_left
8594_right

8596_left
8596_right
8598_left
8598_right
8605_left
8605_right
8610_left
8610_right
8614_left
8614_right
8616_left
8616_right
8619_left
8619_right
8620_left
8620_right
8621_left
8621_right
8623_left
8623_right
8627_left
8627_right
8631_left
8631_right
8632_left
8632_right
8637_left
8637_right
8638_left
8638_right
8640_left
8640_right
8642_left
8642_right
8643_left
8643_right
8644_left
8644_right
8645_left
8645_right
8646_left
8646_right
8649_left
8649_right
8650_left
8650_right
8651_left
8651_right

8652_left
8652_right
8653_left
8653_right
8654_left
8654_right
8655_left
8655_right
8657_left
8657_right
8660_left
8660_right
8662_left
8662_right
8663_left
8663_right
8664_left
8664_right
8666_left
8666_right
8669_left
8669_right
8673_left
8673_right
8675_left
8675_right
8677_left
8677_right
8678_left
8678_right
8679_left
8679_right
8681_left
8681_right
8682_left
8682_right
8683_left
8683_right
8695_left
8695_right
8698_left
8698_right
8699_left
8699_right
8706_left
8706_right
8709_left
8709_right

8711_left
8711_right
8716_left
8716_right
8723_left
8723_right
8726_left
8726_right
8740_left
8740_right
8742_left
8742_right
8743_left
8743_right
8744_left
8744_right
8749_left
8749_right
8752_left
8752_right
8753_left
8753_right
8754_left
8754_right
8755_left
8755_right
8759_left
8759_right
8760_left
8760_right
8762_left
8762_right
8767_left
8767_right
8768_left
8768_right
8769_left
8769_right
8770_left
8770_right
8771_left
8771_right
8776_left
8776_right
8778_left
8778_right
8779_left
8779_right

8781_left
8781_right
8782_left
8782_right
8783_left
8783_right
8784_left
8784_right
8786_left
8786_right
8788_left
8788_right
8789_left
8789_right
8790_left
8790_right
8793_left
8793_right
8794_left
8794_right
8795_left
8795_right
8797_left
8797_right
8798_left
8798_right
8799_left
8799_right
8804_left
8804_right
8806_left
8806_right
8807_left
8807_right
8810_left
8810_right
8812_left
8812_right
8814_left
8814_right
8816_left
8816_right
8817_left
8817_right
8818_left
8818_right
8822_left
8822_right

8825_left
8825_right
8827_left
8827_right
8830_left
8830_right
8831_left
8831_right
8834_left
8834_right
8835_left
8835_right
8837_left
8837_right
8839_left
8839_right
8842_left
8842_right
8843_left
8843_right
8845_left
8845_right
8851_left
8851_right
8852_left
8852_right
8858_left
8858_right
8860_left
8860_right
8864_left
8864_right
8866_left
8866_right
8873_left
8873_right
8874_left
8874_right
8876_left
8876_right
8877_left
8877_right
8878_left
8878_right
8879_left
8879_right
8881_left
8881_right

8882_left
8882_right
8883_left
8883_right
8884_left
8884_right
8885_left
8885_right
8888_left
8888_right
8889_left
8889_right
8893_left
8893_right
8895_left
8895_right
8896_left
8896_right
8897_left
8897_right
8899_left
8899_right
8900_left
8900_right
8906_left
8906_right
8907_left
8907_right
8912_left
8912_right
8914_left
8914_right
8915_left
8915_right
8919_left
8919_right
8927_left
8927_right
8928_left
8928_right
8935_left
8935_right
8936_left
8936_right
8939_left
8939_right
8940_left
8940_right

8945_left
8945_right
8949_left
8949_right
8950_left
8950_right
8951_left
8951_right
8954_left
8954_right
8956_left
8956_right
8958_left
8958_right
8959_left
8959_right
8962_left
8962_right
8964_left
8964_right
8965_left
8965_right
8966_left
8966_right
8967_left
8967_right
8968_left
8968_right
8971_left
8971_right
8974_left
8974_right
8975_left
8975_right
8977_left
8977_right
8981_left
8981_right
8988_left
8988_right
8993_left
8993_right
8995_left
8995_right
8999_left
8999_right
9000_left
9000_right

9001_left
9001_right
9003_left
9003_right
9005_left
9005_right
9008_left
9008_right
9010_left
9010_right
9011_left
9011_right
9015_left
9015_right
9016_left
9016_right
9017_left
9017_right
9020_left
9020_right
9024_left
9024_right
9025_left
9025_right
9033_left
9033_right
9036_left
9036_right
9040_left
9040_right
9042_left
9042_right
9049_left
9049_right
9051_left
9051_right
9063_left
9063_right
9066_left
9066_right
9068_left
9068_right
9069_left
9069_right
9074_left
9074_right
9080_left
9080_right

9082_left
9082_right
9083_left
9083_right
9088_left
9088_right
9092_left
9092_right
9096_left
9096_right
9098_left
9098_right
9099_left
9099_right
9100_left
9100_right
9101_left
9101_right
9103_left
9103_right
9106_left
9106_right
9107_left
9107_right
9115_left
9115_right
9116_left
9116_right
9119_left
9119_right
9120_left
9120_right
9122_left
9122_right
9123_left
9123_right
9126_left
9126_right
9131_left
9131_right
9133_left
9133_right
9134_left
9134_right
9135_left
9135_right
9144_left
9144_right

9145_left
9145_right
9147_left
9147_right
9148_left
9148_right
9151_left
9151_right
9154_left
9154_right
9156_left
9156_right
9157_left
9157_right
9159_left
9159_right
9163_left
9163_right
9166_left
9166_right
9167_left
9167_right
9168_left
9168_right
9173_left
9173_right
9177_left
9177_right
9180_left
9180_right
9183_left
9183_right
9189_left
9189_right
9190_left
9190_right
9198_left
9198_right
9199_left
9199_right
9200_left
9200_right
9203_left
9203_right
9204_left
9204_right
9205_left
9205_right

9207_left
9207_right
9213_left
9213_right
9214_left
9214_right
9216_left
9216_right
9218_left
9218_right
9219_left
9219_right
9220_left
9220_right
9224_left
9224_right
9226_left
9226_right
9232_left
9232_right
9233_left
9233_right
9237_left
9237_right
9238_left
9238_right
9243_left
9243_right
9244_left
9244_right
9251_left
9251_right
9253_left
9253_right
9255_left
9255_right
9257_left
9257_right
9258_left
9258_right
9259_left
9259_right
9266_left
9266_right
9268_left
9268_right
9269_left
9269_right

9271_left
9271_right
9277_left
9277_right
9278_left
9278_right
9279_left
9279_right
9280_left
9280_right
9281_left
9281_right
9282_left
9282_right
9284_left
9284_right
9285_left
9285_right
9287_left
9287_right
9288_left
9288_right
9294_left
9294_right
9297_left
9297_right
9298_left
9298_right
9300_left
9300_right
9307_left
9307_right
9308_left
9308_right
9310_left
9310_right
9311_left
9311_right
9312_left
9312_right
9313_left
9313_right
9317_left
9317_right
9322_left
9322_right
9324_left
9324_right

9325_left
9325_right
9333_left
9333_right
9339_left
9339_right
9340_left
9340_right
9342_left
9342_right
9346_left
9346_right
9348_left
9348_right
9351_left
9351_right
9353_left
9353_right
9358_left
9358_right
9359_left
9359_right
9366_left
9366_right
9368_left
9368_right
9370_left
9370_right
9376_left
9376_right
9377_left
9377_right
9379_left
9379_right
9381_left
9381_right
9382_left
9382_right
9383_left
9383_right
9384_left
9384_right
9390_left
9390_right
9391_left
9391_right
9392_left
9392_right

9396_left
9396_right
9398_left
9398_right
9402_left
9402_right
9404_left
9404_right
9411_left
9411_right
9412_left
9412_right
9419_left
9419_right
9423_left
9423_right
9428_left
9428_right
9448_left
9448_right
9450_left
9450_right
9452_left
9452_right
9454_left
9454_right
9456_left
9456_right
9462_left
9462_right
9463_left
9463_right
9464_left
9464_right
9465_left
9465_right
9467_left
9467_right
9469_left
9469_right
9478_left
9478_right
9485_left
9485_right
9487_left
9487_right
9488_left
9488_right

9490_left
9490_right
9497_left
9497_right
9499_left
9499_right
9502_left
9502_right
9503_left
9503_right
9505_left
9505_right
9507_left
9507_right
9509_left
9509_right
9511_left
9511_right
9513_left
9513_right
9516_left
9516_right
9517_left
9517_right
9518_left
9518_right
9521_left
9521_right
9525_left
9525_right
9526_left
9526_right
9528_left
9528_right
9529_left
9529_right
9535_left
9535_right
9536_left
9536_right
9538_left
9538_right
9539_left
9539_right
9540_left
9540_right
9542_left
9542_right

```
9544_left
9544_right
9545_left
9545_right
9546_left
9546_right
9547_left
9547_right
9549_left
9549_right
9553_left
9553_right
9555_left
9555_right
9560_left
9560_right
9564_left
9564_right
9566_left
9566_right
9569_left
9569_right
9570_left
9570_right
9573_left
9573_right
9578_left
9578_right
9584_left
9584_right
9586_left
9586_right
9587_left
9587_right
9589_left
9589_right
9590_left
9590_right
9592_left
100%|      | 1427/1427 [00:00<00:00, 5590.69it/s]
9592_right
9593_left
9593_right
9595_left
9595_right
9598_left
9598_right
```

9601_left
9601_right
9602_left
9602_right
9603_left
9603_right
9604_left
9604_right
9608_left
9608_right
9613_left
9613_right
9614_left
9614_right
9616_left
9616_right
9619_left
9619_right
9622_left
9622_right
9626_left
9626_right
9628_left
9628_right
9632_left
9632_right
9633_left
9633_right
9636_left
9636_right
9638_left
9638_right
9640_left
9640_right
9642_left
9642_right
9643_left
9643_right
9646_left
9646_right
9647_left
9647_right
9648_left
9648_right
9649_left
9649_right
9651_left
9651_right

9652_left
9652_right
9653_left
9653_right
9657_left
9657_right
9658_left
9658_right
9659_left
9659_right
9661_left
9661_right
9663_left
9663_right
9665_left
9665_right
9667_left
9667_right
9668_left
9668_right
9670_left
9670_right
9672_left
9672_right
9673_left
9673_right
9680_left
9680_right
9682_left
9682_right
9683_left
9683_right
9684_left
9684_right
9688_left
9688_right
9689_left
9689_right
9690_left
9690_right
9691_left
9691_right
9693_left
9693_right
9694_left
9694_right
9696_left
9696_right

9697_left
9697_right
9698_left
9698_right
9699_left
9699_right
9702_left
9702_right
9705_left
9705_right
9708_left
9708_right
9709_left
9709_right
9713_left
9713_right
9714_left
9714_right
9716_left
9716_right
9717_left
9717_right
9721_left
9721_right
9722_left
9722_right
9724_left
9724_right
9728_left
9728_right
9736_left
9736_right
9737_left
9737_right
9739_left
9739_right
9740_left
9740_right
9741_left
9741_right
9742_left
9742_right
9744_left
9744_right
9747_left
9747_right
9750_left
9750_right

9752_left
9752_right
9755_left
9755_right
9759_left
9759_right
9761_left
9761_right
9764_left
9764_right
9767_left
9767_right
9768_left
9768_right
9772_left
9772_right
9775_left
9775_right
9777_left
9777_right
9779_left
9779_right
9781_left
9781_right
9783_left
9783_right
9785_left
9785_right
9788_left
9788_right
9789_left
9789_right
9793_left
9793_right
9794_left
9794_right
9795_left
9795_right
9796_left
9796_right
9797_left
9797_right
9798_left
9798_right
9799_left
9799_right
9800_left
9800_right

9801_left
9801_right
9804_left
9804_right
9807_left
9807_right
9809_left
9809_right
9810_left
9810_right
9811_left
9811_right
9812_left
9812_right
9813_left
9813_right
9817_left
9817_right
9819_left
9819_right
9821_left
9821_right
9826_left
9826_right
9828_left
9828_right
9830_left
9830_right
9831_left
9831_right
9832_left
9832_right
9840_left
9840_right
9844_left
9844_right
9845_left
9845_right
9846_left
9846_right
9848_left
9848_right
9849_left
9849_right
9853_left
9853_right
9854_left
9854_right

9855_left
9855_right
9857_left
9857_right
9863_left
9863_right
9868_left
9868_right
9869_left
9869_right
9871_left
9871_right
9873_left
9873_right
9874_left
9874_right
9875_left
9875_right
9876_left
9876_right
9878_left
9878_right
9879_left
9879_right
9884_left
9884_right
9889_left
9889_right
9893_left
9893_right
9894_left
9894_right
9895_left
9895_right
9896_left
9896_right
9897_left
9897_right
9899_left
9899_right
9900_left
9900_right
9906_left
9906_right
9907_left
9907_right
9913_left
9913_right

9914_left
9914_right
9919_left
9919_right
9925_left
9925_right
9927_left
9927_right
9929_left
9929_right
9930_left
9930_right
9933_left
9933_right
9934_left
9934_right
9937_left
9937_right
9939_left
9939_right
9940_left
9940_right
9942_left
9942_right
9945_left
9945_right
9947_left
9947_right
9948_left
9948_right
9949_left
9949_right
9951_left
9951_right
9952_left
9952_right
9957_left
9957_right
9958_left
9958_right
9959_left
9959_right
9961_left
9961_right
9963_left
9963_right
9964_left
9964_right

```
9965_left
9965_right
9966_left
9966_right
9967_left
9967_right
9971_left
9971_right
9972_left
9972_right
9975_left
9975_right
9976_left
9976_right
9977_left
9977_right
9979_left
9979_right
9980_left
9980_right
9984_left
9984_right
9992_left
9992_right
9993_left
9993_right
9996_left
9996_right
9998_left
9998_right
9999_left
```

[9]:
```
test = train[:1000]
len(train)
```

[9]:
```
1427
```

[10]:
```
# Set the target size to a smaller value
target_size = (224, 224)

# Iterate over each image in the test set
for f in tqdm(test.values):
    try:
        # Load the image using the correct path
        img = load_img('9804_left.jpeg'.format(f), target_size=target_size)
        arr = img_to_array(img)
```

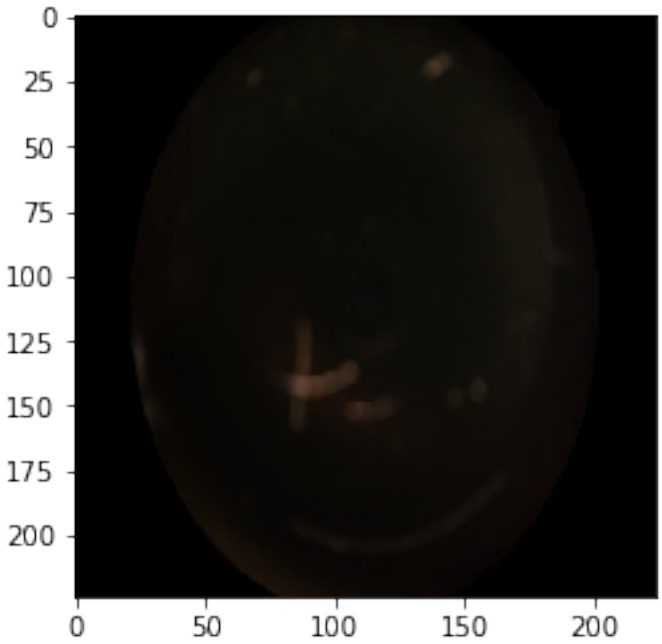
```
        label = one_hot_labelsY[i]
        x_train.append(arr)
        y_train.append(label)
        i += 1
    except:
        pass
```

```
100%|      | 1000/1000 [00:02<00:00, 377.29it/s]
```

[11]:
```
len(x_train)
```

[11]:
```
1000
```

[12]:
```
plt.imshow(x_train[681]/255) #681 > Try some other number too
plt.show()
```



[13]:
```
x_valid = []
y_valid = []
X_train, X_valid, Y_train, Y_valid = train_test_split(x_train, y_train,␣
 ↪test_size=0.1, random_state=1)
```

[14]:
```
visible = Input(shape=(224,224,3))
conv1 = Conv2D(32, kernel_size=4, activation='relu')(visible)
pool1 = MaxPooling2D(pool_size=(2, 2))(conv1)
conv2 = Conv2D(16, kernel_size=4, activation='relu')(pool1)
pool2 = MaxPooling2D(pool_size=(2, 2))(conv2)
```

```
flat = Flatten()(pool2)
hidden1 = Dense(10, activation='relu')(flat)
output = Dense(1, activation='sigmoid')(hidden1)
model = Model(inputs=visible, outputs=output)
```

[15]:
```
y_train_raw = np.array(Y_train)
x_train_raw = np.array(X_train)
```

[21]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_2 (Conv2D)           (None, 783, 783, 32)      1568

 max_pooling2d_2 (MaxPooling  (None, 391, 391, 32)     0
 2D)

 conv2d_3 (Conv2D)           (None, 388, 388, 64)      32832

 max_pooling2d_3 (MaxPooling  (None, 194, 194, 64)     0
 2D)

 conv2d_4 (Conv2D)           (None, 191, 191, 128)     131200

 max_pooling2d_4 (MaxPooling  (None, 95, 95, 128)      0
 2D)

 flatten_1 (Flatten)         (None, 1155200)           0

 dense_2 (Dense)             (None, 256)               295731456

 dropout (Dropout)           (None, 256)               0

 dense_3 (Dense)             (None, 5)                 1285

=================================================================
Total params: 295,898,341
Trainable params: 295,898,341
Non-trainable params: 0
_________________________________________________________________
```

[22]:
```
# Convert labels to integers and adjust the label range
y_train_raw = y_train_raw.astype(np.int32)
y_train_raw -= y_train_raw.min()
```

```
# Define the model
model = keras.models.Sequential([
    keras.layers.Conv2D(32, kernel_size=4, activation='relu', input_shape=(224,␣
 ↪224, 3)),
    keras.layers.MaxPooling2D(pool_size=(2, 2)),
    keras.layers.Conv2D(64, kernel_size=4, activation='relu'),
    keras.layers.MaxPooling2D(pool_size=(2, 2)),
    keras.layers.Conv2D(128, kernel_size=4, activation='relu'),
    keras.layers.MaxPooling2D(pool_size=(2, 2)),
    keras.layers.Flatten(),
    keras.layers.Dense(256, activation='relu'),
    keras.layers.Dropout(0.5),
    keras.layers.Dense(5, activation='softmax')
])

# Compile the model
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',␣
 ↪metrics=['accuracy'])
```

[23]:
```
x_valid_raw = np.array(X_valid)
y_valid_raw = np.array(Y_valid)
```

[24]:
```
# Define the callback to stop training once the desired accuracy is reached
class StopTrainingAtAccuracy(keras.callbacks.Callback):
    def __init__(self, target_accuracy):
        super(StopTrainingAtAccuracy, self).__init__()
        self.target_accuracy = target_accuracy

    def on_epoch_end(self, epoch, logs=None):
        if logs is None:
            logs = {}
        if logs.get('val_accuracy') >= self.target_accuracy:
            self.model.stop_training = True
            print(f"\nReached {self.target_accuracy*100:.2f}% validation␣
 ↪accuracy. Stopping training...\n")
```

[25]:
```
history = model.fit(x_train_raw, y_train_raw, epochs=10,␣
 ↪validation_data=(x_valid_raw, y_valid_raw),
                    callbacks=[StopTrainingAtAccuracy(0.80)])
```

```
Epoch 1/10
29/29 [==============================] - 85s 3s/step - loss: 7.1256 - accuracy:
0.6467 - val_loss: 0.8418 - val_accuracy: 0.7500
Epoch 2/10
29/29 [==============================] - 99s 3s/step - loss: 0.9515 - accuracy:
0.7244 - val_loss: 0.9085 - val_accuracy: 0.7500
Epoch 3/10
29/29 [==============================] - 131s 5s/step - loss: 0.9635 - accuracy:
```

```
0.7244 - val_loss: 0.8464 - val_accuracy: 0.7500
Epoch 4/10
29/29 [==============================] - 127s 4s/step - loss: 0.9344 - accuracy:
0.7244 - val_loss: 0.8282 - val_accuracy: 0.7500
Epoch 5/10
29/29 [==============================] - 104s 4s/step - loss: 0.9403 - accuracy:
0.7244 - val_loss: 0.8335 - val_accuracy: 0.7500
Epoch 6/10
29/29 [==============================] - 94s 3s/step - loss: 0.9160 - accuracy:
0.7244 - val_loss: 0.8301 - val_accuracy: 0.7500
Epoch 7/10
29/29 [==============================] - 93s 3s/step - loss: 0.9425 - accuracy:
0.7244 - val_loss: 0.8349 - val_accuracy: 0.7500
Epoch 8/10
29/29 [==============================] - 103s 4s/step - loss: 0.9315 - accuracy:
0.7244 - val_loss: 0.8551 - val_accuracy: 0.7500
Epoch 9/10
29/29 [==============================] - 107s 4s/step - loss: 0.9304 - accuracy:
0.7244 - val_loss: 0.8299 - val_accuracy: 0.7500
Epoch 10/10
29/29 [==============================] - 125s 4s/step - loss: 0.9235 - accuracy:
0.7244 - val_loss: 0.8310 - val_accuracy: 0.7500
```

[ ]: