CS364/AM792: Homework #6

Due on 11 June, 2020 at 23:59pm

Unathi K. Skosana

Problem 1

I have chosen the Keras deep learning API. Keras is built on top of popular machine learning library by Google, TensorFlow. According to Keras' about page, Keras is an infrastructure layer for differentiable programming, with the ability to efficiently compute the gradient of arbitrary differentiable expressions and low-level tensor operations on CPU, GPU or TPU. The reason for choosing `Keras` was because of it's popularity in the scientific python community, making it easier to find learning resources on the internet. The `TensorFlow` and `Keras` packages can be easily installed to a python virtual environment via `pip`.

Problem 2

(a) Here the images from the previous assignment were resized to 224×224, and loaded onto a numpy array with the following code snippet.

```
## Building labels and training data
train_data = []
train_labels = []

classes = ["Coast", "Forest", "Highway", "Kitchen",\
                "Mountain", "Office", "Store", "Street", "Suburb"]

for cl in classes:
  folder_path = os.path.join("../assets/assignment-6/dataset/train", cl)
  for fname in os.listdir(folder_path):
    fpath = os.path.join(folder_path, fname)
    im = io.imread(fpath)
    im_resized = transform.resize(im, (224, 224) , anti_aliasing=True)
    im_arr = img_to_array(im_resized)
    train_data.append(im_arr)
    train_labels.append(classes.index(cl))

train_data = np.array(train_data)
train_labels = np.array(train_labels)
```

(b)

The three convolutional stages consisted of a convolutional layer of kernel size 2×2 with the layer sizes progressively increasing from 16 to 64. Similarly, the max pooling layer was of size 2×2. These parameters were taken from one of the code examples of Keras (Keras 2020) titled 'Image classification from scratch'. The model was trained with `SGD` for a mini batch size of 16 and a learning rate of 0.005 for 20 epochs. This yielded an accuracy of 66% on the testing set.

(c)

Using Keras' `ImageDataGenerator` we can augment some of the training set images to randomly flip left-right so as to increase the sample size. Furthermore a dropout of 50% was added just before the final stage of the model. In the same values for the learning rate, batch size and number of epochs, which yielded an accuracy of roughly 73%.

Problem 2 continued on next page...

(d)

The top-1 accuracy is just same as the test accuracy and the top-3 accuracy for this model was 93.6%. The confusion matrix is given below

$$M_{\mathcal{C}} = \begin{bmatrix} 187 & 2 & 44 & 9 & 8 & 8 & 0 & 0 & 2 \\ 4 & 187 & 0 & 4 & 15 & 3 & 10 & 5 & 0 \\ 10 & 4 & 114 & 15 & 2 & 4 & 2 & 7 & 2 \\ 0 & 0 & 0 & 82 & 0 & 16 & 8 & 1 & 3 \\ 33 & 27 & 28 & 28 & 125 & 16 & 3 & 12 & 2 \\ 0 & 0 & 0 & 12 & 0 & 103 & 0 & 0 & 0 \\ 0 & 3 & 0 & 21 & 0 & 10 & 162 & 11 & 8 \\ 0 & 2 & 2 & 10 & 1 & 7 & 12 & 153 & 5 \\ 0 & 0 & 2 & 10 & 0 & 5 & 5 & 4 & 115 \end{bmatrix}$$

Here are a few samples of correctly classified test images.

(a) Coast scene correctly identified

(b) Kitchen scene correctly identified

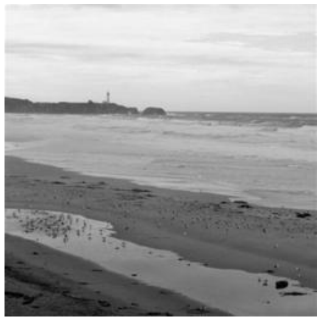
(c) Forest scene correctly identified

(d) Mountain scene correctly identified

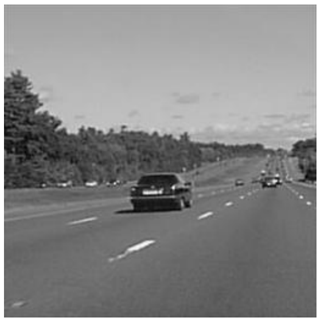
(e) Forest scene identified

Here are a few samples of incorrectly classified test images.

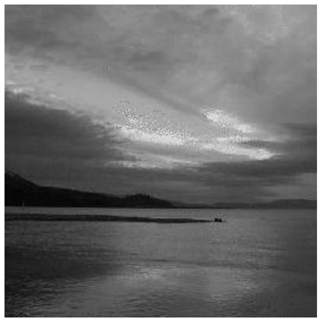
(a) Mountain scene incorrectly identified as a coast scene



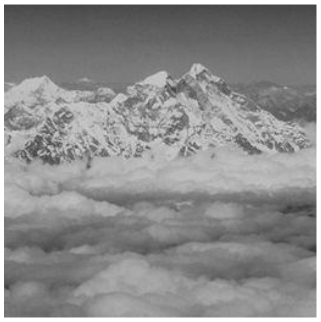
(b) Coast scene incorrectly identified as a mountain scene



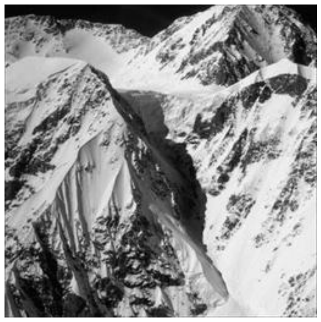
(c) Office scene incorrectly identified as a office scene



(d) Kitchen scene incorrectly identified as mountain scene



(e) Highway scene incorrectly identified as mountain scene



Problem 3

(a)

The model I choose for transfer learning was InceptionV3, primarily because of it's high top-1 accuracy, only second to Microsoft's ResNet, this plausibly due to the network being less deep than the latter. Google's InceptionV3 is mainly focused on computational cost (Google Cloud 2020), with ResNet focused on accuracy. However, the error rate of ResNet is not marginally better than InceptionV3's. So with the resources (inc. time) available to me at the time of writing prompt me to use InceptionV3.

(b)

The input tensor was of shape $(224, 224, 3)$, this was done by repeating the grayscale value of a pixel across the three channels, then appropriately converting these values to integer values. This data was then pre-processed by a helper function from Keras so that it lies in the range $[-1, 1]$. Using the same values for the learning rate, batch size and number of epochs, after fitting the model, the model yielded 95% accuracy on the test set.

(c)

The top-1 accuracy is the same the test accuracy here as well. The top-3 accuracy is an outstanding 99% ! The confusion matrix is given below

$$M_{\mathcal{C}} = \begin{bmatrix} 241 & 3 & 5 & 0 & 8 & 0 & 3 & 0 & 0 \\ 1 & 210 & 2 & 0 & 8 & 0 & 5 & 0 & 2 \\ 2 & 0 & 144 & 1 & 1 & 0 & 1 & 11 & 0 \\ 0 & 0 & 0 & 100 & 0 & 8 & 2 & 0 & 0 \\ 1 & 5 & 0 & 0 & 268 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 113 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 2 & 208 & 0 & 1 \\ 0 & 0 & 4 & 0 & 0 & 0 & 4 & 181 & 3 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 1 & 138 \end{bmatrix} \tag{1}$$

Problem 3 continued on next page...

(a) Forest scene incorrectly identified as a mountain scene

(b) Mountain scene incorrectly identified as a coast scene

(c) Suburb scene incorrectly identified as a forest scene

(d) Street scene incorrectly identified as a highway scene

(e) Forest scene incorrectly identified as coast scene

Problem 3 continued on next page...

Here are a few samples of correctly identified images.

(a) Coast scene correctly identified

(b) Store scene correctly identified

(c) Store scene correctly identified

(d) Mountain scene correctly identified

(e) Street scene correctly identified

References

Keras.io, *Code examples*, viewed 11 June 2020, https://Keras.io/examples/
Google Cloud, *Advanced Guide to Inception v3 on Cloud TPU*, viewed 11 June 2020, https://cloud.google.com/tpu/docs/incep
v3-advanced