Machine Learning Engineer Nanodegree

Plant Seedling Classification

Abin Saju
April 8th, 2019

Domain Background

Classification of seedlings is an important process in agriculture. It is important to be able to identify a weed in a batch of cultivated plants to ensure the seedling does not have to compete for the limited resources. The process is important as it ensures high crop yields, however, the current method involves visual inspection which is not a scalable and reliable method.

Problem Statement

The project aims to use publicly available data to automatically classify seedlings. The training dataset contains RGB images with species labels identifying each image as one of twelve species. During training, the performance of the model will be evaluated using a micro-averaged f1-score. Once the model is trained on the dataset, it would be used to predict images in the testing dataset. The predictions will then be verified by submitting the generated file to Kaggle.

Using machine learning to solve the problem means the problem can be easily scaled with a high degree of accuracy. The final objective of the project would be to successfully classify all data points in the testing set.

Datasets and Inputs

Data for the project is downloaded from https://www.kaggle.com/c/plant-seedlings-classification/data. The dataset contains coloured images of 960 unique pants belonging to 12 species at several growth stages [1]. In total, there are 4750 training data points and 794 testing points. The data is provided publicly by the authors of the paper on plant seedling classification [2].

The authors of the paper have collected the data scientifically, in the hope that the data can be used in a machine learning framework.

Solution Statement

The project will aim to classify the testing dataset, the results of which will be submitted to Kaggle for verification. The testing dataset contains PNG images, the model will predict if the image is one of 12 species.

Benchmark Model

The performance of the model will be compared to the ResNet top-1 error from the ResNet paper [3]. The model was trained on a classification task, similar to this project and it would be interesting to compare the performance of the two models.

Evaluation Metrics

I am using the evaluation metric provided on the Kaggle website. Which is a micro-averaged F1-score.

$$Precision_{micro} = \frac{\sum_{k \in C} TP_k}{\sum_{k \in C} TP_k + FP_k}$$

$$Recall_{micro} = \frac{\sum_{k \in C} TP_k}{\sum_{k \in C} TP_k + FN_k}$$

F1-score is the harmonic mean of precision and recall

$$MeanFScore = F1_{micro} = \frac{2Precision_{micro}Recall_{micro}}{Precision_{micro} + Recall_{micro}}$$

Figure 1: micro F1 score [4]

Project Design

The project can be split into three different sections data loading, model creation and predictions. In the data loading part, the images will be loaded and converted to tensors. The data can be visualised at this stage and enhancements and augmentation can be applied to improve the quality and quantity of the provided data. Looking briefly at the data set it can be seen the lighting in the images is poor, which would be my first to rectify after running through a simple model.

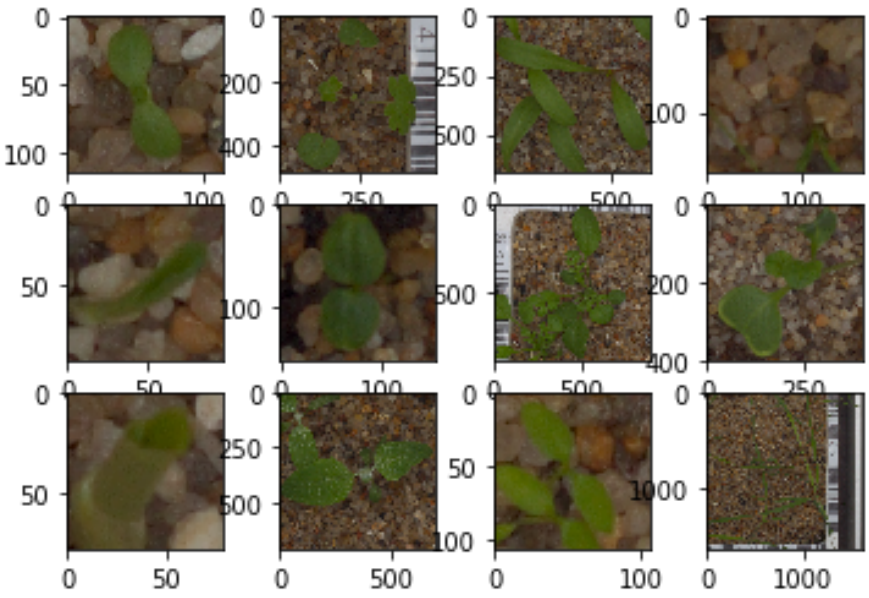

Figure 2: Examples from training dataset

For model creation, I would aim to create a basic model to ensure the data can be run through and iron out any minor issues. Once a simple model is found to work with the data, I would attempt to work with more complex models such as ResNet. I have provided code below of a simple model.

```
from keras.layers import Conv2D, MaxPooling2D, GlobalAveragePooling2D
from keras.layers import Dropout, Flatten, Dense
from keras.models import Sequential

model = Sequential()

model.add(Conv2D(16,kernel_size=2,input_shape=(224,224,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(32,(2,2), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(64,(2,2), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(GlobalAveragePooling2D())
model.add(Dense(12,activation='softmax'))

model.summary()

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

For the training of the model, the data would be split 80/20 between the training and the validation set. Furthermore, to report the micro f1 score after each epoch a call-back will be added, as shown below.

```
from keras.callbacks import Callback
from sklearn.metrics import confusion_matrix, f1_score, precision_score, recall_score
from sklearn.model_selection import train_test_split

class Metrics(Callback):
    def on_train_begin(self, logs={}):
        self.val_f1s = []
        self.val_recalls = []
        self.val_precisions = []

    def on_epoch_end(self, epoch, logs={}):
        val_predict = (np.asarray(self.model.predict(self.validation_data[0]))).round()
        val_targ = self.validation_data[1]
```

```
        _val_f1 = f1_score(val_targ, val_predict, average =
'micro')
        _val_recall = recall_score(val_targ, val_predict, average
= 'micro')
        _val_precision = precision_score(val_targ, val_predict,
average = 'micro')
        self.val_f1s.append(_val_f1)
        self.val_recalls.append(_val_recall)
        self.val_precisions.append(_val_precision)
        print '— val_f1: %f — val_precision: %f — val_recall %f'
%(_val_f1, _val_precision, _val_recall)
        return

metrics = Metrics()
```

The predictions part would predict the images in the testing dataset and create a submission csv file which can be uploaded into Kaggle.

Works Cited

[1] M. Dyrmann and P. Christiansen, "Plant Seedlings Dataset," 2014. [Online]. Available: https://vision.eng.au.dk/plant-seedlings-dataset/.

[2] T. M. Giselsson, R. N. Jørgensen, P. K. Jensen, M. . Dyrmann and H. S. Midtiby, "A Public Image Database for Benchmark of Plant Seedling Classification Algorithms," *Biosystems Engineering,* vol. , no. , p. , 2016.

[3] K. . He, X. . Zhang, S. . Ren and J. . Sun, "Deep Residual Learning for Image Recognition," *arXiv: Computer Vision and Pattern Recognition,* vol. , no. , pp. 770-778, 2016.

[4] Kaggle Inc, "Plant Seedlings Classification," [Online]. Available: https://www.kaggle.com/c/plant-seedlings-classification/overview/evaluation.