**Project Report**

Group5

Chenxi Sun

**Abstract**

There are 5 steps for our general process dealing with ‘Understanding the Amazon form Space’ project.

First, download the data sets, browse images and analyze labels. We find that each image’s tag is composed of several single labels. There are 17 single labels in all. And the labels can be divided into different categories.

For the second step, we build two models — weather model and label13 model. Both of these models are built based on convolutional neutral network. we use keras library to built and train these two models. After getting two models, we make a combination model.

Third, after we get the trained model, we evaluate two models and one combination model by calculating F2-score. For label13 model, the F2-score value is 0,8344. For weather, the accuracy is 0.878. For combination model the F2-score value is 0.874. And we also analyze the reason why get these results.

Fourth, we summarize the advantages and disadvantages of two models and try to propose some improved methods.

Finally, we use our models to predict the tags for the images in test-jpg folder.

1. **Analyze Data**

We download the data from kaggle and get train-jpg folder (contain the images for training set), train\_v2.csv (labels for training set’s images), test-jpg folder (test data). Then use python on jupyter notebook to do data analysis.

First, show tags of images. We find that each tag is composed of several single labels. For example, train\_0.jpg’s tag is ‘haze primary’. That is, this image’s tag contains two single labels — haze and primary.

And we sum up all single labels and there are 17 kinds of single labels in all. see figure 1. The number of images for each single labels, see figure 2.

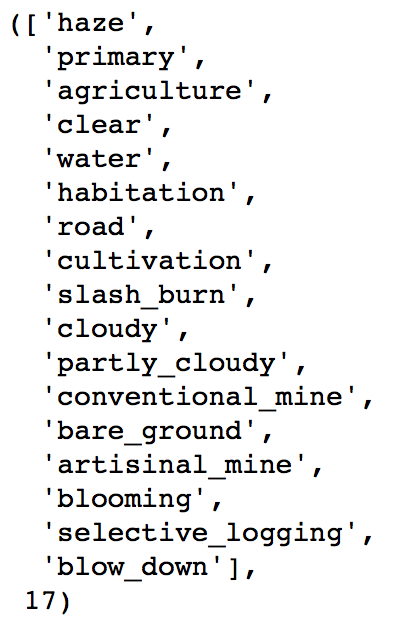


figure 1 all single labels

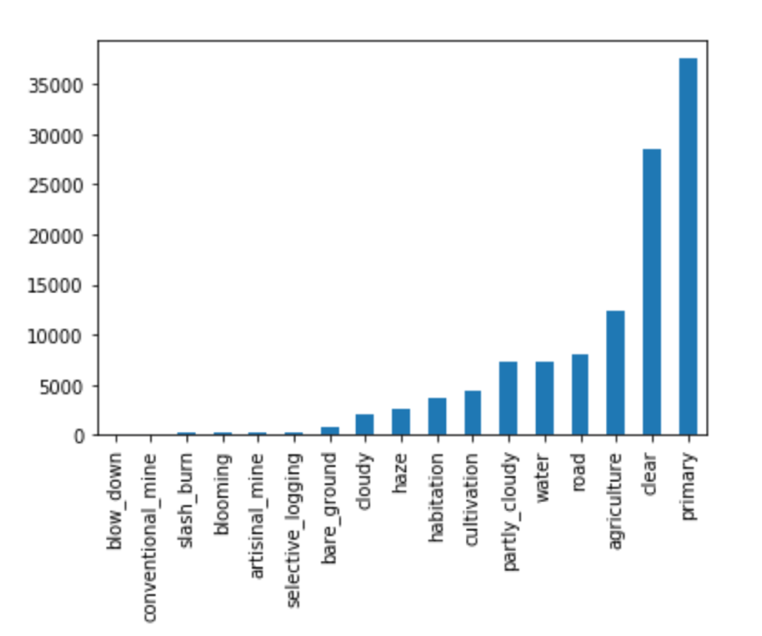


figure 2 number of images for each single labels

We also find these single labels can be divided into several categories. For example, weather labels have ‘haze’, ‘clear’, ‘cloudy’, ‘partly cloudy’. We decide to make two categories — weather class and label13 class.

Weather class have the labels which are used to describe the weather of images. Why we decide to make a weather class? There are two reasons:

* One image could have one weather label at most (analysis shown in figure 3). It is a four-classification problem.
* Every image must have a weather label. ( the number of showing times of four labels equal the number of all images )

Label13 class contain 13 kinds of labels, most of them are used to describe the landscape of images. But at first we made two categories for label13 class, one is land class and the other is rare class. Rare labels are that the labels which have rare images. All of their number of images are less than 1000. But we think all of them have the feature of landscape. So we decide to make one label13 class rather than two classes.

Then we do visualization for single labels. We make co-occurrence matrix for each category. Figure 3 is weather’ co-occurrence matrix. It’s clearly that the weather co-occurrence matrix is a diagonal matrix. That is, for each images, it should have exactly one weather label. It makes sense because it’s not possible to have two kinds of weather at the same time.

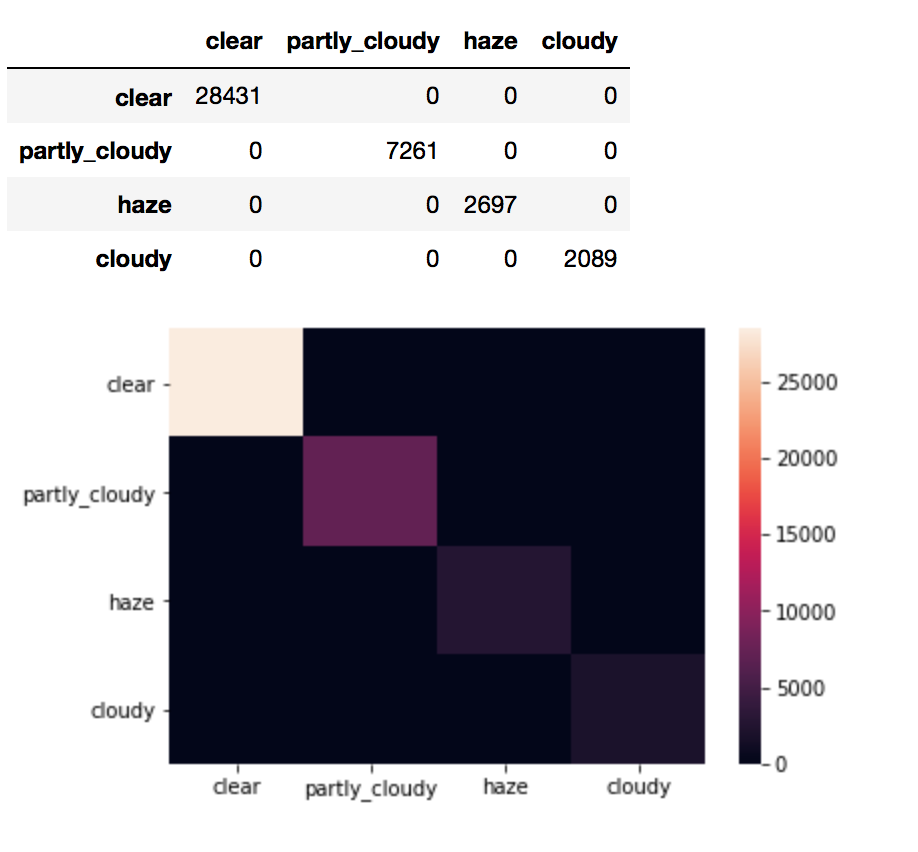


figure 3 weather’s co-occurrence matrix

Figure 4 is land’s co-occurrence matrix and figure 5 is rare labels’ co-occurrence matrix. We can find that both of them are not diagonal matrix. One image can have several land labels and rare labels at the same time. And that is also why we decide to make these two kinds of categories as one class.

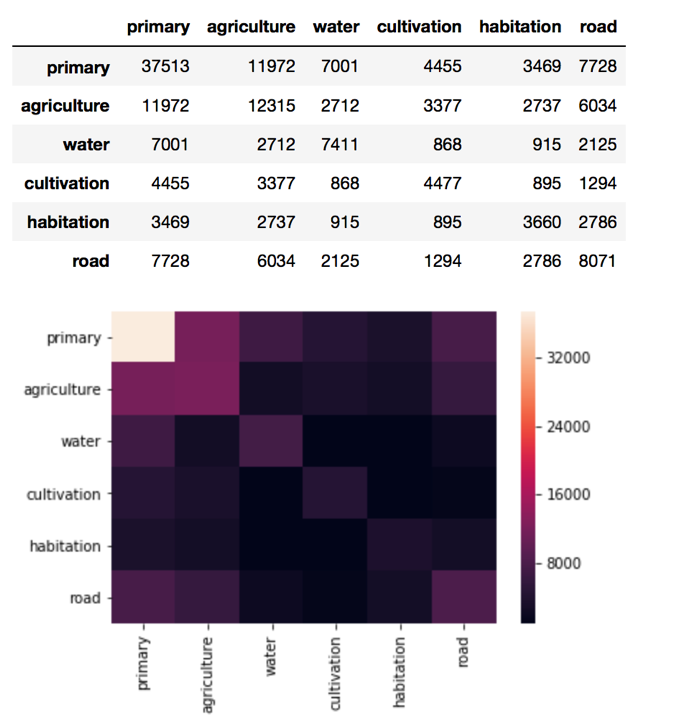


figure 4 land’s co-occurrence matrix

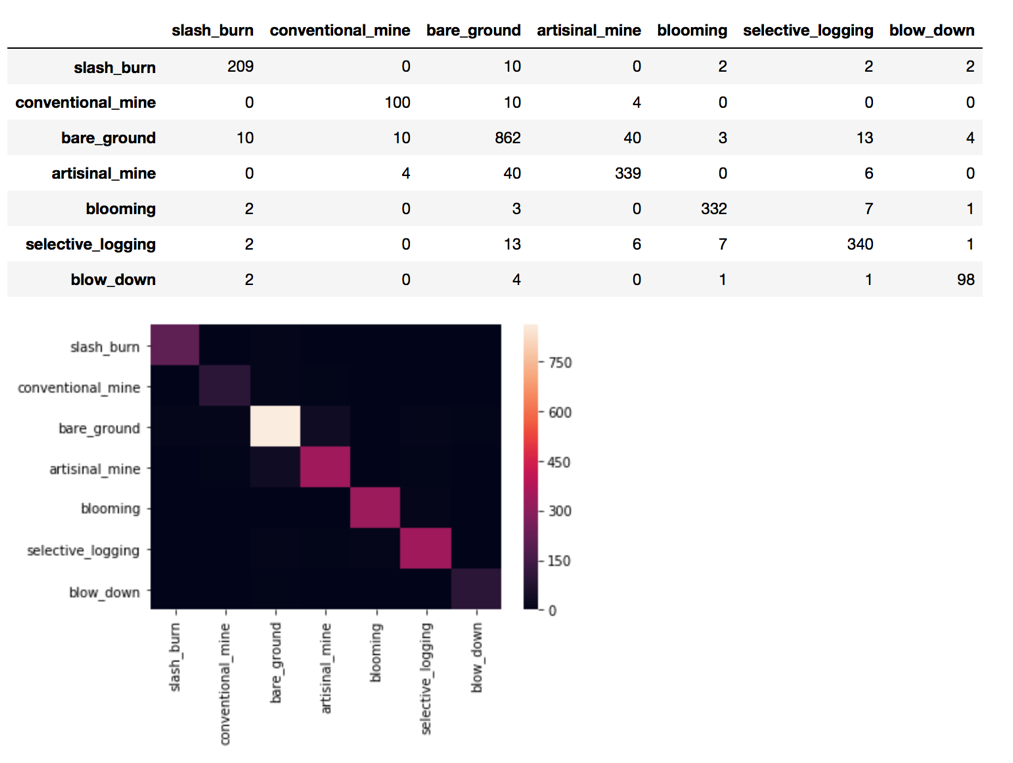


figure 5 rare co-occurrence matrix

Then make a big co-occurrence matrix for all of the labels, see figure 6. In this matrix, in ‘cloudy row’, only ‘cloudy column’ has the value, other column’s values are 0. Thus, we reckon that if one image has the ‘cloudy’ label, it will not have other labels.

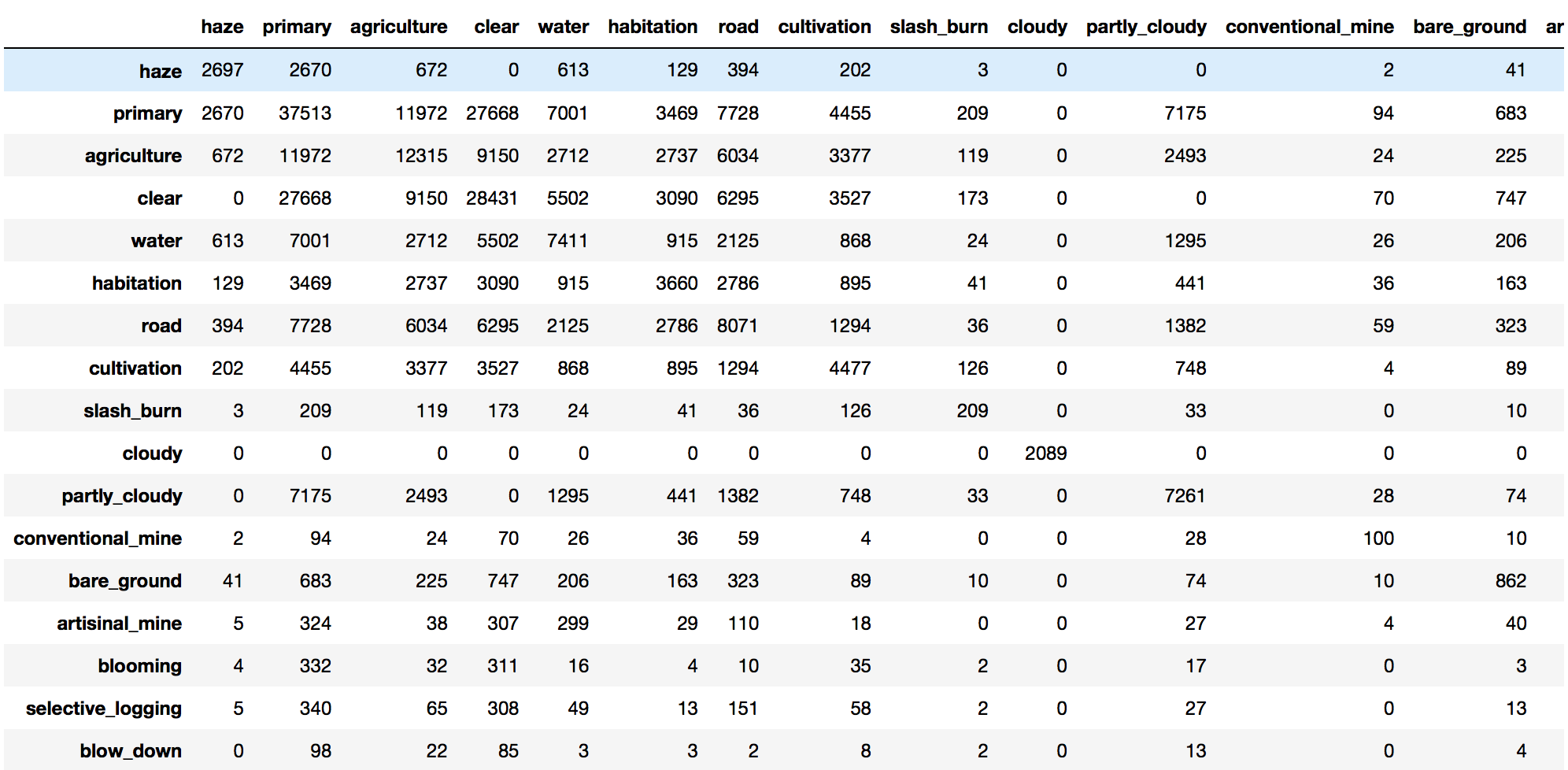


figure 6 co-

We show the sample image for each single label to see how them look like. Here, we just put 4 images for 4 labels (partly-cloudy, road, primary, agriculture) as figure 7.

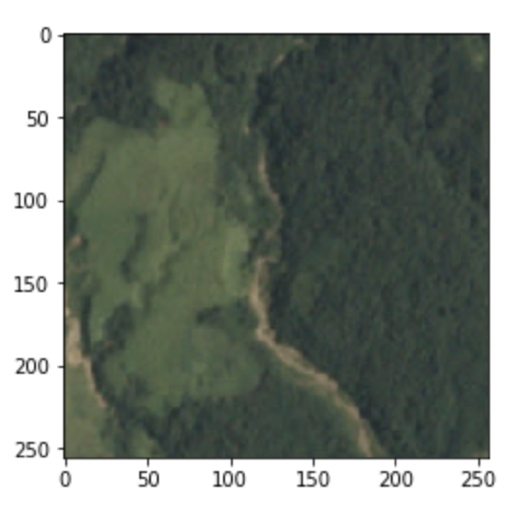
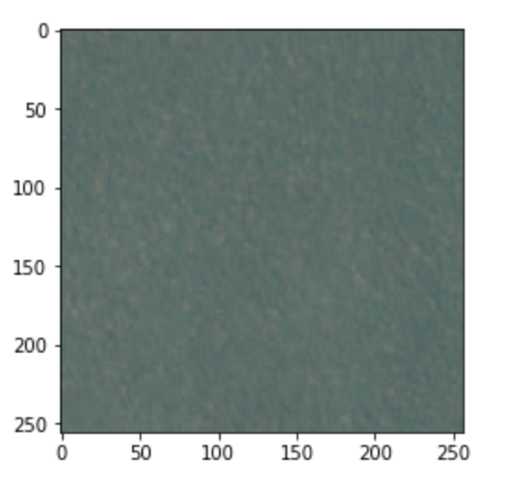
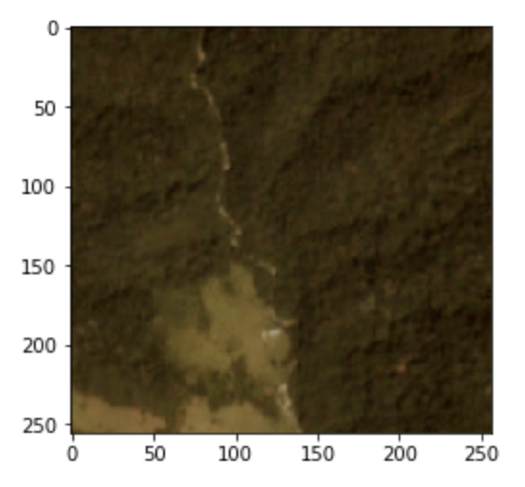
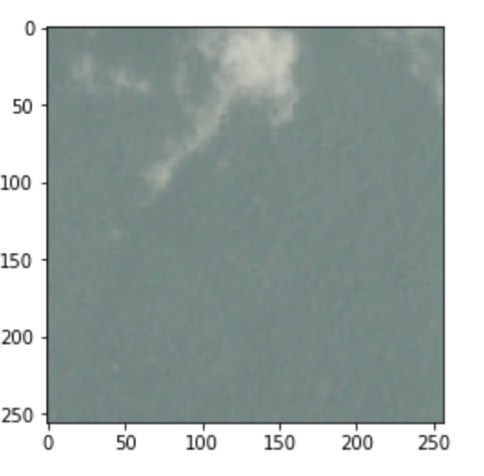


figure 7 partly-cloudy, road, primary, agriculture images

1. **Build Models**

**2.1 Label13 Model**

As one image can have several labels in label13 class. So we can build model for every separated labels. One original idea is that we create 13 models of every labels. And each model can solve the binary classification problem. For example, a road model’s mission is that telling whether the image has road feature or not. 13 models will produce 13 results, assembling these results will get the final results.

In practice, we just build one model to solve this problem. First we choose to use convolutional neural network to make the model. As all of 13 models are to solve the binary classification problem, they will have the same structure of cnn. Thus, making one general model and letting 13 models to share it’s layers is a better choice. Our label13 model using keras is shown in figure 8.



figure 8 label13 model using keras

**2.1.1 model details**

**Input**

Choose 10000 images in train-jpg folder. There are rare images which have labels of slash\_burn, conventional\_mine, bare\_ground, artisinal\_main, blooming, selective\_logging and blow\_down. We dare the network cannot learn the pattern of rare labels. so we write code to extract images which have the specific labels to pick out more images which have these rare labels and add them into training set. The number of training images is 10798. the size of training image is 64\*64\*3. But at first they are 256\*256\*4. We find the 4th channel (alpha channel) is always 255. it’s no use for the feature learning and the training speed will be faster without this channel. Input x is an array which are got from the resized input images.

One image has one corresponding tag. We define each tag is a vector contain 13 values. ‘1’ means that this image has this single label, ‘0’ means that the image does not have this single label. That is one-hot code. Of course one vector can have several ‘1’ which means one image can have several single labels.

**Convolutional Neural Network**

Use keras to build this model. First, two convolutional layer which have 32 kinds of 3\*3 kernels, of cause the active function are relu. And then use the 2\*2 max pooling layer. Then 2 convolutional layers which both have 64 kinds of 3\*3 kennels. And then one max pooling layer. Then similar layers. Then fully connected layer. Finally, output layer. In this model we put 13 output neurons in output layer. And each of them present one kind of label. For every neuron we use sigmoid to active it. Because each neuron’s job is to tell whether the input owns the corresponding label or not (binary classification problem). If the output is more than 0.5, it means the input image should have the corresponding single label.

Use RMSprop as the optimizer, the learning rate is 0.00001. Loss function use binary crossentropy. We add dropout to fix the over fitting problem.

**Train the Model**

Put the training set into this model, compile and fit it. we use early stopping to protect it to be over fitting. The patient is 16. 75% inputs are training set, 25% are validation set. Run it and get the result as figure 9. After early stopping, finally the accuracy on training set is 0.9434 and the validation accuracy is 0.9020.

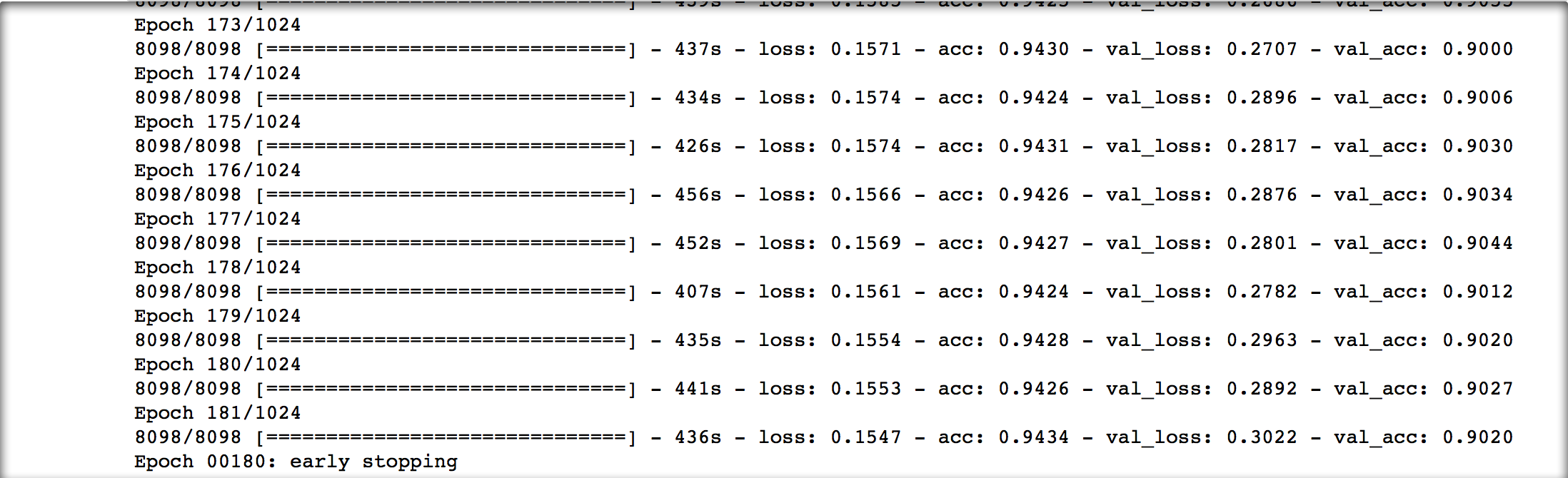


figure 9 training model

Draw the loss graph as figure 10. We find both the training loss and the validation loss become lower and lower until There's a tendency that the validation loss starts to increase.

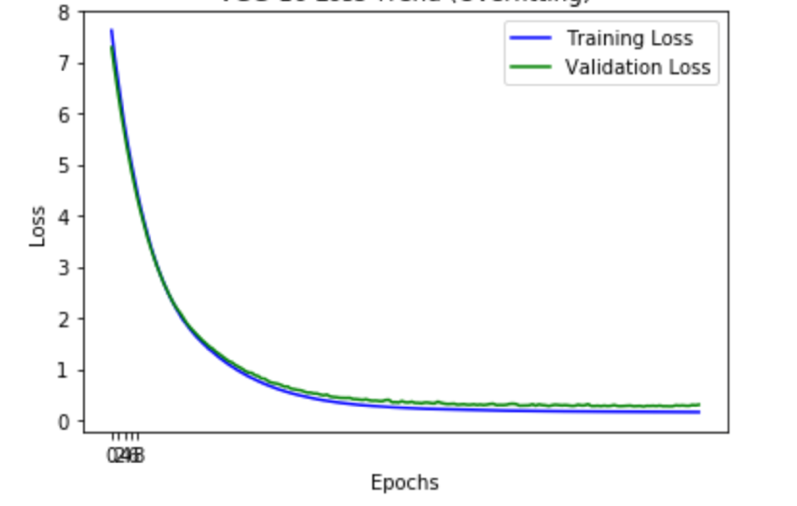


figure 10 loss graph

**Save Model**

Finally save this model and the weight in h5 file.

**Prediction**

When one image is put into this model, a vector will be given back. This vector has 13 values. Each value represents the possibility of the image having corresponding label. If a value is higher than 0.5, we will accept the corresponding label.

**2.2 Weather Model**

As one image can have at most one weather label, this is a four-classification problem. we also use cnn based on keras to build this model.

Four kinds of labels’ images and each single label have 1000 images. All of them is 4000 images. Input labels are vectors which have 4 values. There is only one ‘1’ in 4 values as one images can only have one weather label. The image size is same as label13 model input.

convolutional neural network is similar with label13 model but the output layer has 4 neurons and use softmax to figure out which one is the the best predicting label.

The model is shown in figure 12.



figure 12 weather model using keras

When training the weather model, use early stopping with validation loss and patient 100 to avoid over fitting. Shown as figure 13.

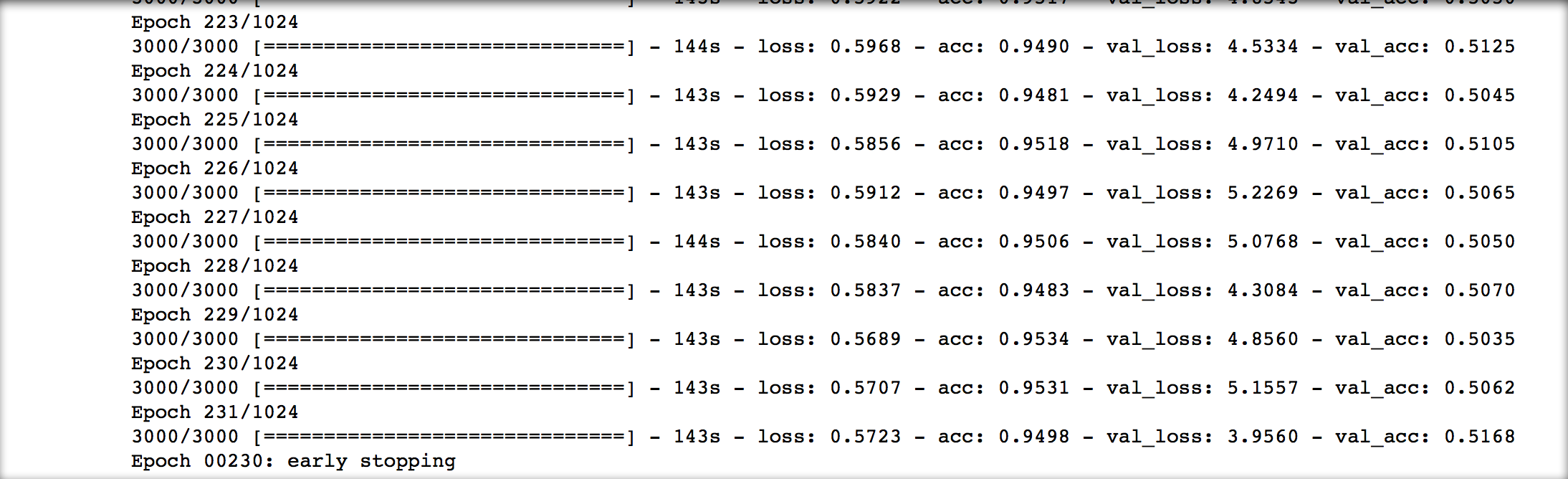


figure 13 result

The loss graph is shown in figure 14.

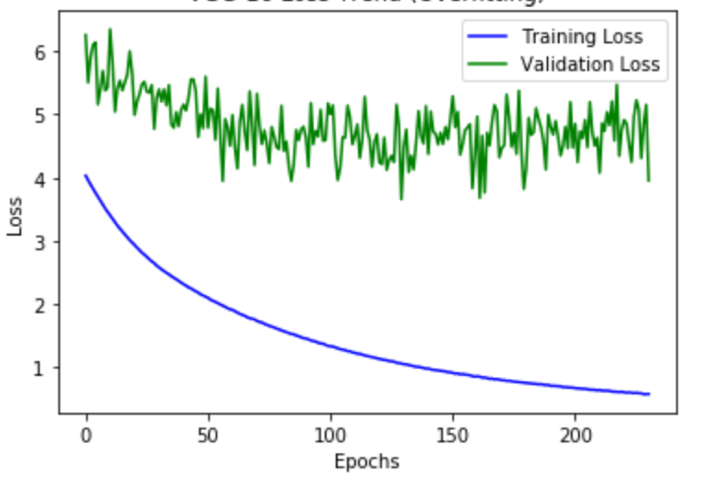


figure 14 loss graph

As we can see, the last validation accuracy is not good, only 0.5168. And the validation loss does not have the decrease tendency. But the training accuracy is high to 0.95. In this cease, there is no over fitting as we use the early stopping. So it dose get the best weight for this model. If the accuracy is bed, there is something wrong with the model itself, but not the epoch or over fitting. But when we make prediction of some sample using this model, the accuracy is not bad. Detailed testing and analysis are presented in part 3.

**Prediction**

When one image is put into this model and predict it labels. The out out is a vector which have 4 values. We will take the label which have the highest value as the image’s weather label.

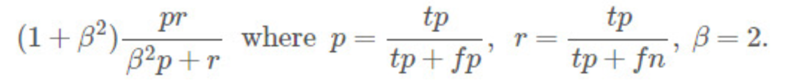
**2.3 Combination Model**

The combination model is a serialization of weather model and label13 model. The prediction label result link two parts of labels which are got from two models.

1. **Model Evaluation**

**3.1 Label13 Model Evaluation**

Select 1000 images in train-jpg folder and use this model to predict labels. Using F2-score to analyze. F2-score is:



Write a function to calculate F2-score, the function is shown as figure 15.



figure 15 F2-score function

Put the result of prediction into F2-score function, we can get the F2-score is 0.83442



The average F2-score for the predict is 0.83442. It’s not good.

We also make F2-score for each single label, and the result is as figure 16

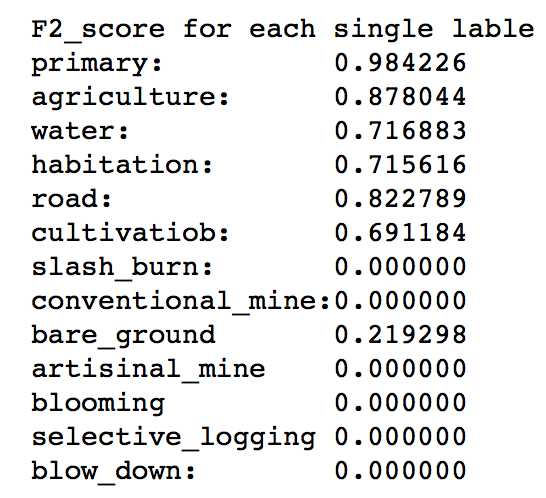
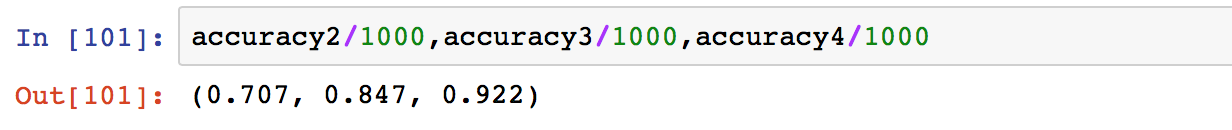


figure 16 F2-score

In the F2-score for each single label, we can find that many label’s score is zero. Print the true-positive, false-positive, false-negative values for last 7 labels. The value is small, even less than 10. That is, there are a few images have these 7 labels in this prediction set. At the same time, we find that both these 7 labels are rare labels. as the training set have a few images of these labels, so the feature might be not learned well.

And here are other accuracy -- accuracy of all labels right, accuracy of allowing one label wrong, accuracy of allowing two labels wrong. The values of them are as follow.



Have quick look at predicting labels and it’s corresponding images. See figure 16

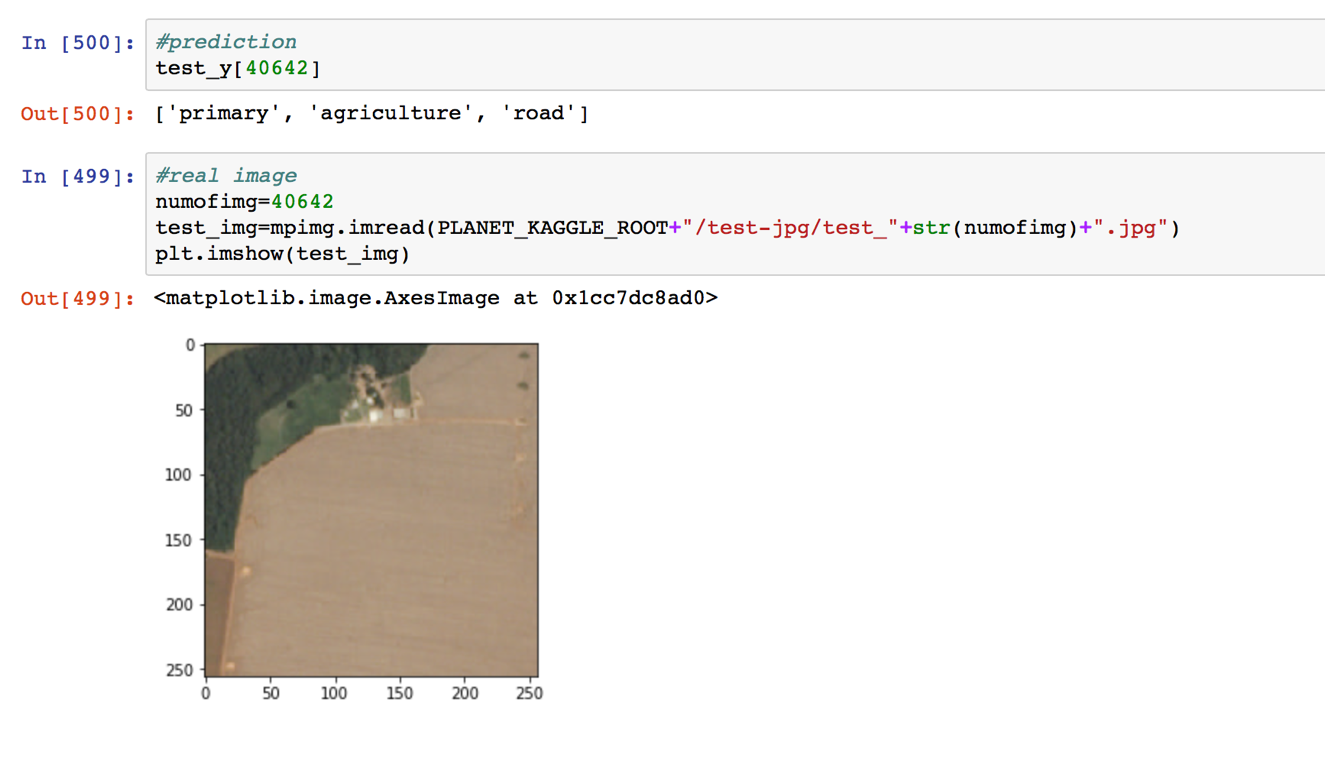




figure 16

* 1. **weather model evaluation**

As one image can have one weather label at most, we choose simple accuracy but not F2-score to evaluate the model. The accuracy is defined as follow:

We write the function to calculate this accuracy. Extract 1000 images from training set as test set. Put it into model and predict the labels. Use the result and the accuracy function to calculate the accuracy. Then we can get the accuracy which is 0.878. See figure 17.

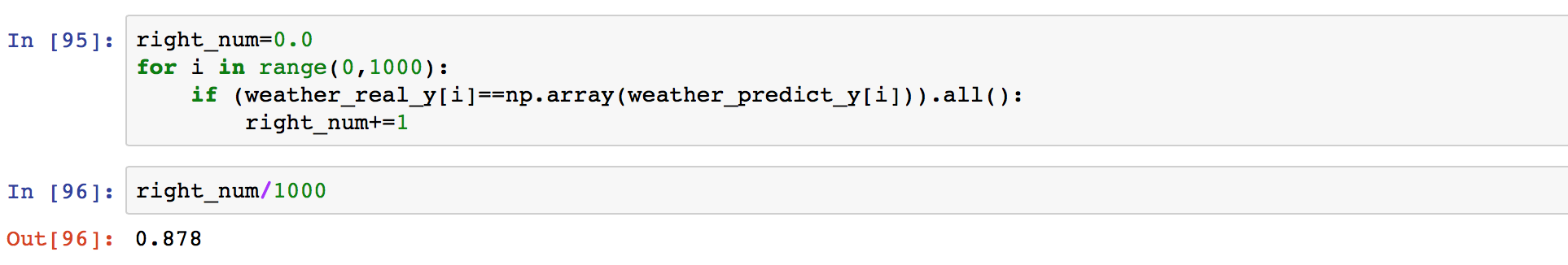


figure 17 weather model test accuracy

We have analyzed in part 2, when training the model, the validation accuracy is always 0.5. But here, the accuracy is more than 0.8. We have made sure that the test images are different from the training images. And we also make many tests, all tests’ results are more than 0.8. But why does this happen? We are not sure yet, but we have some speculation:

The validation accuracy 0.5 is not bad for four-classification problem.

When training the model, using 0.25 splitting images as training set and validation set has chance. Maybe the validation set have many images which are difficult to classify by weather model.

When training weather model, the training accuracy is more than 0.9. It is also a principle to define the presentation of model. Because the model is not over fitting.

But we have no idea about the most reasonable explanation. We have to continue learning.

* 1. **Combination model evaluation**

Link the label13 model and weather model, we get the combination model. Put 1000 images as samples into the combination model and do prediction. Then we can get the prediction labels which can be seen as the linking of model13 model prediction labels and weather model prediction labels.

The calculate the F2-score. The function is created as figure 18.

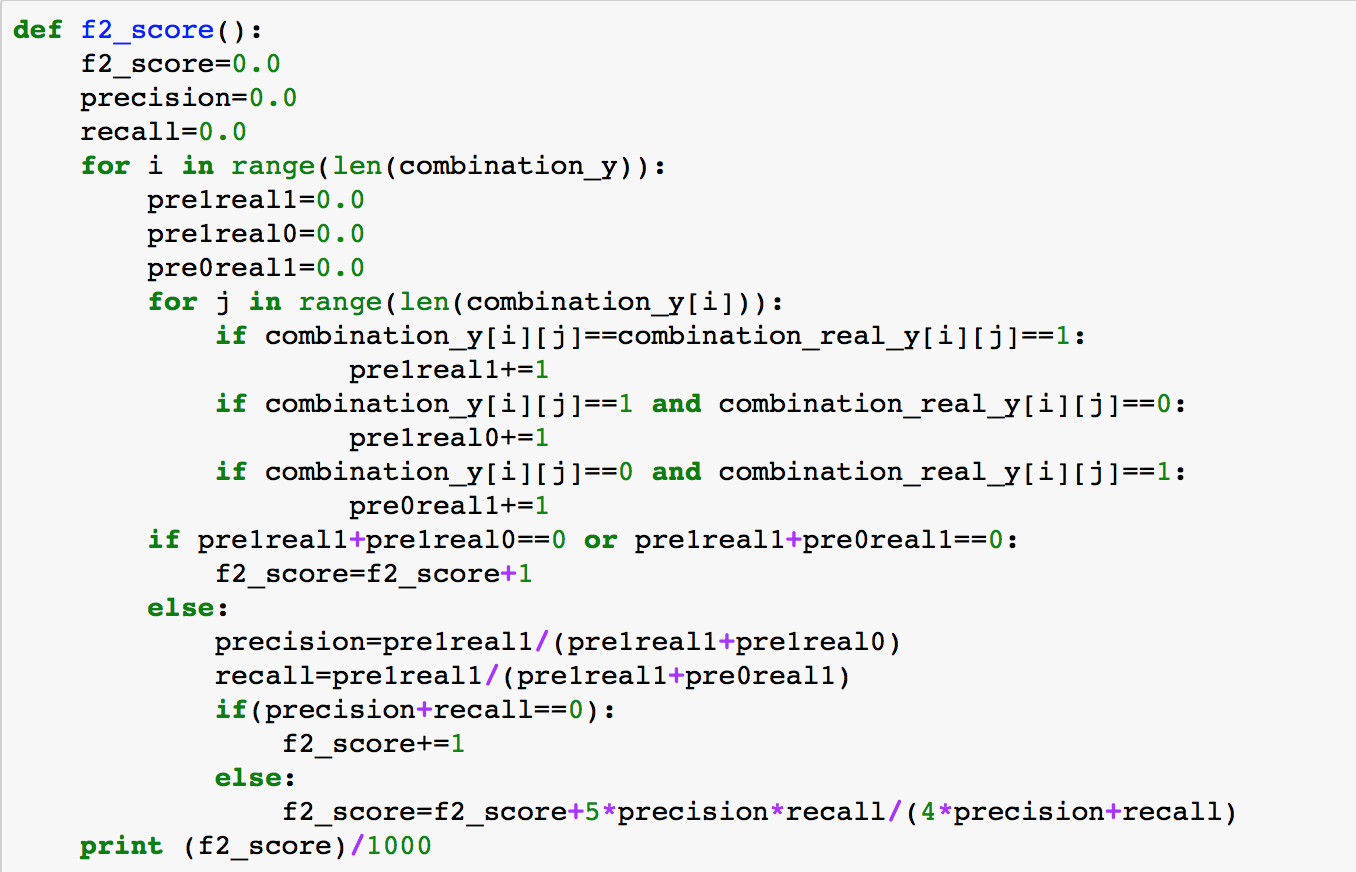
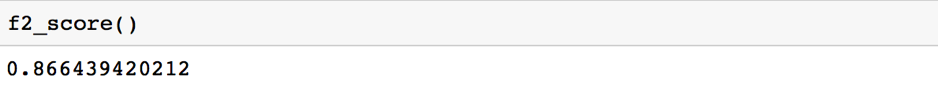


figure 18 F2-score function for combination model

Run this function, we get the value of F2-score, the value is 0.866



Then we change the test set, using other 5000 images to get the F2-score, the value is 0.874. It is a not bad accuracy.

1. **Advantages and Disadvantages**

**4.1** **Advantages**

* We separate the labels into two reasonable categories. Let several single labels share the layers of one model.
* We make some changes when we decide whether a label belongs to an image or not. In label13 model, one original idea is that as we use sigmoid active function, one label belongs to an image if the predict accuracy is higher than 0.5. But it performs badly when make a decision about the rare label. Then we change the threshold to 0.1 for rare labels. The result becomes better.
* Our model’s F2-score is not bad. Higher than 0.85 and average is 0.87.
  1. **Disadvantages**
* Although we do some changes when we make a decision about the rare labels, the result is not good yet.
* There is still a big space for improvement in accuracy
* Use cnn as the model and use library’s function, do not have much innovation.
  1. **Improvement method**
* For rare labels’ images, we can make horizontal/vertical flip and rotation to get more images to rich the data set. It’s one reason that the rare labels prediction is bad.
* To do some images’ data enhancement. As for weather, the feature of weather is global but not detailed. We can do something to enhance images’ weather feature.
* Choice other neural network to solve this problem. Because our knowledge is limited, we still have to learn more knowledge of machine learning.

1. **Test-jpg prediction**

We use our model to predict the images’ tags in test-jpg folder. Then save as csv file. We try to put it on kaggle and get the score, but it failed. Because kaggle’s submission form must have more data (mapping work) which we don’t do. So we just get the prediction tags in csv file but we have no way to get its accuracy. The result is shown in figure 19.

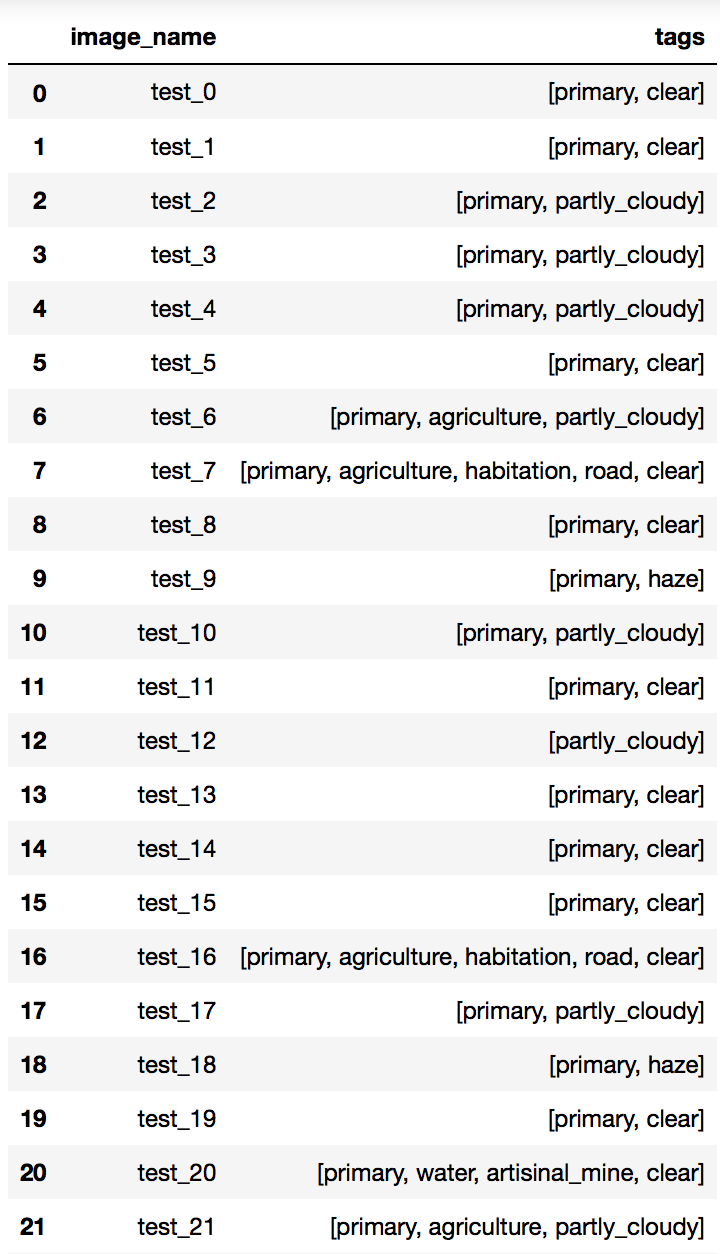
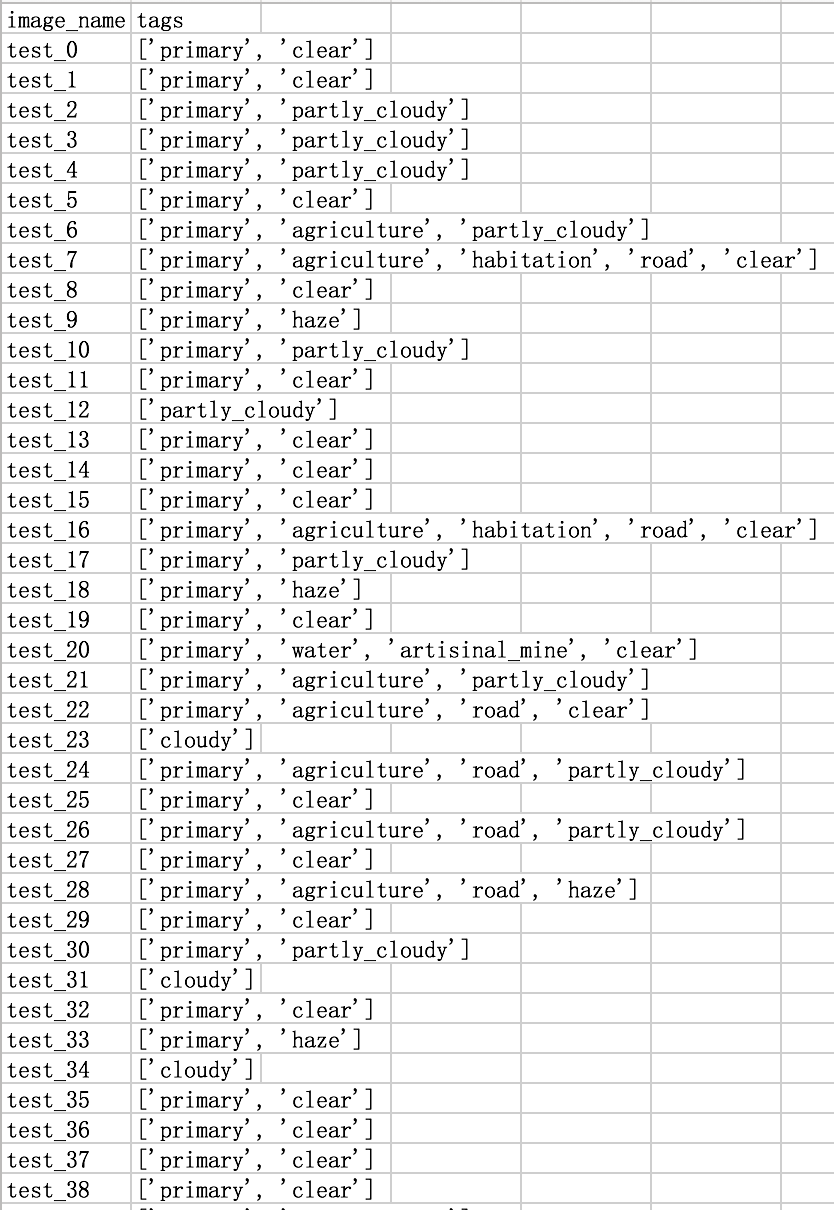


figure 19 two csv open forms