In [1]:

```
# This Python 3 environment comes with many helpful analytics libraries ins
talled
# It is defined by the kaggle/python docker image:
https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load in

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will
list the files in the input directory

#from subprocess import check_output
#print(check_output(["ls", "D:\Kaggle\Amazon-basin"]).decode("utf8"))
import os
print(os.listdir('D:/Kaggle/Amazon-basin'))
```

```
['.ipynb_checkpoints', 'Code.ipynb', 'sample_submission_v2.csv', 'test-jpg'
, 'train-jpg', 'train_v2.csv', 'weights_kfold_1.h5']
```

In [2]:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import os
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D, BatchNormalization
from keras.callbacks import EarlyStopping, ModelCheckpoint
import cv2
from tqdm import tqdm
from keras import optimizers
#from sklearn.model_selection import KFold
from sklearn.cross_validation import KFold
from sklearn.metrics import fbeta_score
import time

x_train = []
x_test = []
y_train = []

df_train = pd.read_csv('train_v2.csv')
df_test = pd.read_csv('sample_submission_v2.csv')

flatten = lambda l: [item for sublist in l for item in sublist]
labels = list(set(flatten([l.split(' ') for l in
df_train['tags'].values])))

labels = ['blow_down',
 'bare_ground',
 'conventional_mine',
 'blooming',
 'cultivation',
 'artisinal_mine',
 'haze',
```

```
 'haze',
 'primary',
 'slash_burn',
 'habitation',
 'clear',
 'road',
 'selective_logging',
 'partly_cloudy',
 'agriculture',
 'water',
 'cloudy']

label_map = {'agriculture': 14,
 'artisinal_mine': 5,
 'bare_ground': 1,
 'blooming': 3,
 'blow_down': 0,
 'clear': 10,
 'cloudy': 16,
 'conventional_mine': 2,
 'cultivation': 4,
 'habitation': 9,
 'haze': 6,
 'partly_cloudy': 13,
 'primary': 7,
 'road': 11,
 'selective_logging': 12,
 'slash_burn': 8,
 'water': 15}
```

```
Using TensorFlow backend.
D:\Anaconda3\lib\site-packages\sklearn\cross_validation.py:44: DeprecationW
arning: This module was deprecated in version 0.18 in favor of the model_se
lection module into which all the refactored classes and functions are move
d. Also note that the interface of the new CV iterators are different from
that of this module. This module will be removed in 0.20.
  "This module will be removed in 0.20.", DeprecationWarning)
```

In [3]:

```
for f, tags in tqdm(df_train.values, miniters=1000):
    img = cv2.imread('./train-jpg/{}.jpg'.format(f))
    targets = np.zeros(17)
    for t in tags.split(' '):
        targets[label_map[t]] = 1
    x_train.append(cv2.resize(img, (64, 64)))
    y_train.append(targets)



y_train = np.array(y_train, np.uint8)
x_train = np.array(x_train, np.float32)/255.

print(x_train.shape)
print(y_train.shape)
```

```
100%|
40479/40479 [01:30<00:00, 445.79it/s]
```

```
(40479, 64, 64, 3)
```

```
(40479, 17)
```

In [4]:

```
for f, tags in tqdm(df_test.values, miniters=1000):
    img = cv2.imread('./test-jpg/{}.jpg'.format(f))
    x_test.append(cv2.resize(img, (64, 64)))

x_test  = np.array(x_test, np.float32)/255.
```

```
100%|
61191/61191 [02:15<00:00, 453.15it/s]
```

In [5]:

```
from sklearn.cross_validation import KFold

nfolds = 5

num_fold = 0
sum_score = 0

yfull_test = []
yfull_train =[]

kf = KFold(len(y_train), n_folds=nfolds, shuffle=True, random_state=1)

for train_index, test_index in kf:
        start_time_model_fitting = time.time()

        X_train = x_train[train_index]
        Y_train = y_train[train_index]
        X_valid = x_train[test_index]
        Y_valid = y_train[test_index]

        num_fold += 1
        print('Start KFold number {} from {}'.format(num_fold, nfolds))
        print('Split train: ', len(X_train), len(Y_train))
        print('Split valid: ', len(X_valid), len(Y_valid))

        kfold_weights_path = os.path.join('', 'weights_kfold_' +
str(num_fold) + '.h5')

        model = Sequential()
        model.add(BatchNormalization(input_shape=(64, 64,3)))
        model.add(Conv2D(32, kernel_size=(3, 3),padding='same', activation='
relu'))
        model.add(Conv2D(32, (3, 3), activation='relu'))
        model.add(MaxPooling2D(pool_size=(2, 2)))
        model.add(Dropout(0.25))

        model.add(Conv2D(64, kernel_size=(3, 3),padding='same', activation='
relu'))
        model.add(Conv2D(64, (3, 3), activation='relu'))
        model.add(MaxPooling2D(pool_size=(2, 2)))
        model.add(Dropout(0.25))

        model.add(Conv2D(128, kernel_size=(3, 3),padding='same', activation=
'relu'))
```

```
'relu'))
        model.add(Conv2D(128, (3, 3), activation='relu'))
        model.add(MaxPooling2D(pool_size=(2, 2)))
        model.add(Dropout(0.25))

        model.add(Conv2D(256, kernel_size=(3, 3),padding='same', activation=
'relu'))
        model.add(Conv2D(256, (3, 3), activation='relu'))
        model.add(MaxPooling2D(pool_size=(2, 2)))
        model.add(Dropout(0.25))

        model.add(Flatten())
        model.add(Dense(512, activation='relu'))
        model.add(BatchNormalization())
        model.add(Dropout(0.5))
        model.add(Dense(17, activation='sigmoid'))

        epochs_arr = [20, 10, 5]
        learn_rates = [0.001, 0.0005, 0.00001]

        for learn_rate, epochs in zip(learn_rates, epochs_arr):
            opt  = optimizers.Adam(lr=learn_rate)
            model.compile(loss='binary_crossentropy', # We NEED binary
here, since categorical_crossentropy l1 norms the output before calculating
loss.
                          optimizer=opt,
                          metrics=['accuracy'])
            callbacks = [EarlyStopping(monitor='val_loss', patience=2, verbo
se=0),
            ModelCheckpoint(kfold_weights_path, monitor='val_loss',
save_best_only=True, verbose=0)]

            model.fit(x = X_train, y= Y_train, validation_data=(X_valid, Y_v
alid),
                  batch_size=128,verbose=2, epochs=epochs,callbacks=callback
s,shuffle=True)

        if os.path.isfile(kfold_weights_path):
            model.load_weights(kfold_weights_path)

        p_valid = model.predict(X_valid, batch_size = 128, verbose=2)
        print(fbeta_score(Y_valid, np.array(p_valid) > 0.2, beta=2, average=
'samples'))

        p_train = model.predict(x_train, batch_size =128, verbose=2)
        yfull_train.append(p_train)

        p_test = model.predict(x_test, batch_size = 128, verbose=2)
        yfull_test.append(p_test)

result = np.array(yfull_test[0])
for i in range(1, nfolds):
    result += np.array(yfull_test[i])
result /= nfolds
result = pd.DataFrame(result, columns = labels)
result
```

```
Start KFold number 1 from 5
Split train:  32383 32383
Split valid:  8096 8096
```

```
Train on 32383 samples, validate on 8096 samples
Epoch 1/20
215s - loss: 0.2716 - acc: 0.9009 - val_loss: 0.2535 - val_acc: 0.9067
Epoch 2/20
199s - loss: 0.1640 - acc: 0.9348 - val_loss: 0.1900 - val_acc: 0.9243
Epoch 3/20
199s - loss: 0.1508 - acc: 0.9397 - val_loss: 0.1434 - val_acc: 0.9424
Epoch 4/20
199s - loss: 0.1446 - acc: 0.9422 - val_loss: 0.1351 - val_acc: 0.9457
Epoch 5/20
199s - loss: 0.1391 - acc: 0.9448 - val_loss: 0.1286 - val_acc: 0.9485
Epoch 6/20
199s - loss: 0.1334 - acc: 0.9474 - val_loss: 0.1243 - val_acc: 0.9501
Epoch 7/20
199s - loss: 0.1292 - acc: 0.9492 - val_loss: 0.1217 - val_acc: 0.9528
Epoch 8/20
199s - loss: 0.1265 - acc: 0.9502 - val_loss: 0.1233 - val_acc: 0.9509
Epoch 9/20
199s - loss: 0.1237 - acc: 0.9513 - val_loss: 0.1276 - val_acc: 0.9489
Epoch 10/20
199s - loss: 0.1216 - acc: 0.9523 - val_loss: 0.1186 - val_acc: 0.9530
Epoch 11/20
199s - loss: 0.1199 - acc: 0.9532 - val_loss: 0.1182 - val_acc: 0.9542
Epoch 12/20
199s - loss: 0.1185 - acc: 0.9538 - val_loss: 0.1186 - val_acc: 0.9537
Epoch 13/20
199s - loss: 0.1161 - acc: 0.9547 - val_loss: 0.1153 - val_acc: 0.9553
Epoch 14/20
199s - loss: 0.1148 - acc: 0.9555 - val_loss: 0.1156 - val_acc: 0.9559
Epoch 15/20
199s - loss: 0.1133 - acc: 0.9561 - val_loss: 0.1150 - val_acc: 0.9568
Epoch 16/20
199s - loss: 0.1126 - acc: 0.9561 - val_loss: 0.1197 - val_acc: 0.9542
Epoch 17/20
199s - loss: 0.1104 - acc: 0.9573 - val_loss: 0.1093 - val_acc: 0.9580
Epoch 18/20
199s - loss: 0.1086 - acc: 0.9581 - val_loss: 0.1083 - val_acc: 0.9575
Epoch 19/20
199s - loss: 0.1063 - acc: 0.9588 - val_loss: 0.1106 - val_acc: 0.9575
Epoch 20/20
199s - loss: 0.1069 - acc: 0.9585 - val_loss: 0.1092 - val_acc: 0.9573
Train on 32383 samples, validate on 8096 samples
Epoch 1/10
200s - loss: 0.0998 - acc: 0.9612 - val_loss: 0.1071 - val_acc: 0.9593
Epoch 2/10
199s - loss: 0.0978 - acc: 0.9620 - val_loss: 0.1082 - val_acc: 0.9593
Epoch 3/10
199s - loss: 0.0965 - acc: 0.9627 - val_loss: 0.1067 - val_acc: 0.9596
Epoch 4/10
199s - loss: 0.0954 - acc: 0.9629 - val_loss: 0.1063 - val_acc: 0.9593
Epoch 5/10
199s - loss: 0.0948 - acc: 0.9630 - val_loss: 0.1079 - val_acc: 0.9597
Epoch 6/10
199s - loss: 0.0937 - acc: 0.9638 - val_loss: 0.1134 - val_acc: 0.9576
Epoch 7/10
199s - loss: 0.0925 - acc: 0.9638 - val_loss: 0.1080 - val_acc: 0.9593
Train on 32383 samples, validate on 8096 samples
Epoch 1/5
200s - loss: 0.0883 - acc: 0.9658 - val_loss: 0.1060 - val_acc: 0.9604
Epoch 2/5
```

```
199s - loss: 0.0868 - acc: 0.9662 - val_loss: 0.1059 - val_acc: 0.9605
Epoch 3/5
199s - loss: 0.0865 - acc: 0.9664 - val_loss: 0.1061 - val_acc: 0.9603
Epoch 4/5
199s - loss: 0.0866 - acc: 0.9663 - val_loss: 0.1060 - val_acc: 0.9603
Epoch 5/5
199s - loss: 0.0861 - acc: 0.9664 - val_loss: 0.1058 - val_acc: 0.9603
0.910609779138
Start KFold number 2 from 5
Split train:  32383 32383
Split valid:  8096 8096
Train on 32383 samples, validate on 8096 samples
Epoch 1/20
201s - loss: 0.2803 - acc: 0.9022 - val_loss: 0.2603 - val_acc: 0.9052
Epoch 2/20
196s - loss: 0.1627 - acc: 0.9355 - val_loss: 0.1969 - val_acc: 0.9179
Epoch 3/20
197s - loss: 0.1530 - acc: 0.9392 - val_loss: 0.1433 - val_acc: 0.9419
Epoch 4/20
197s - loss: 0.1452 - acc: 0.9422 - val_loss: 0.1389 - val_acc: 0.9446
Epoch 5/20
197s - loss: 0.1399 - acc: 0.9444 - val_loss: 0.1324 - val_acc: 0.9473
Epoch 6/20
197s - loss: 0.1360 - acc: 0.9463 - val_loss: 0.1346 - val_acc: 0.9460
Epoch 7/20
197s - loss: 0.1313 - acc: 0.9480 - val_loss: 0.1256 - val_acc: 0.9504
Epoch 8/20
197s - loss: 0.1285 - acc: 0.9496 - val_loss: 0.1230 - val_acc: 0.9519
Epoch 9/20
197s - loss: 0.1270 - acc: 0.9497 - val_loss: 0.1214 - val_acc: 0.9520
Epoch 10/20
197s - loss: 0.1238 - acc: 0.9515 - val_loss: 0.1253 - val_acc: 0.9494
Epoch 11/20
197s - loss: 0.1223 - acc: 0.9523 - val_loss: 0.1149 - val_acc: 0.9550
Epoch 12/20
197s - loss: 0.1200 - acc: 0.9529 - val_loss: 0.1134 - val_acc: 0.9560
Epoch 13/20
197s - loss: 0.1182 - acc: 0.9539 - val_loss: 0.1213 - val_acc: 0.9526
Epoch 14/20
197s - loss: 0.1168 - acc: 0.9545 - val_loss: 0.1134 - val_acc: 0.9557
Epoch 15/20
197s - loss: 0.1154 - acc: 0.9549 - val_loss: 0.1113 - val_acc: 0.9568
Epoch 16/20
197s - loss: 0.1145 - acc: 0.9554 - val_loss: 0.1126 - val_acc: 0.9567
Epoch 17/20
197s - loss: 0.1129 - acc: 0.9561 - val_loss: 0.1101 - val_acc: 0.9573
Epoch 18/20
197s - loss: 0.1107 - acc: 0.9568 - val_loss: 0.1159 - val_acc: 0.9549
Epoch 19/20
197s - loss: 0.1100 - acc: 0.9570 - val_loss: 0.1077 - val_acc: 0.9579
Epoch 20/20
197s - loss: 0.1087 - acc: 0.9577 - val_loss: 0.1096 - val_acc: 0.9576
Train on 32383 samples, validate on 8096 samples
Epoch 1/10
199s - loss: 0.1030 - acc: 0.9598 - val_loss: 0.1054 - val_acc: 0.9593
Epoch 2/10
197s - loss: 0.1014 - acc: 0.9606 - val_loss: 0.1060 - val_acc: 0.9592
Epoch 3/10
197s - loss: 0.0994 - acc: 0.9610 - val_loss: 0.1072 - val_acc: 0.9591
Epoch 4/10
```

```
197s - loss: 0.0985 - acc: 0.9617 - val_loss: 0.1081 - val_acc: 0.9591
Train on 32383 samples, validate on 8096 samples
Epoch 1/5
199s - loss: 0.0949 - acc: 0.9632 - val_loss: 0.1039 - val_acc: 0.9605
Epoch 2/5
197s - loss: 0.0938 - acc: 0.9634 - val_loss: 0.1033 - val_acc: 0.9605
Epoch 3/5
197s - loss: 0.0934 - acc: 0.9633 - val_loss: 0.1029 - val_acc: 0.9608
Epoch 4/5
197s - loss: 0.0930 - acc: 0.9634 - val_loss: 0.1028 - val_acc: 0.9608
Epoch 5/5
197s - loss: 0.0925 - acc: 0.9638 - val_loss: 0.1027 - val_acc: 0.9609
0.911462956298
Start KFold number 3 from 5
Split train:  32383 32383
Split valid:  8096 8096
Train on 32383 samples, validate on 8096 samples
Epoch 1/20
202s - loss: 0.2760 - acc: 0.9031 - val_loss: 0.2632 - val_acc: 0.9038
Epoch 2/20
197s - loss: 0.1627 - acc: 0.9354 - val_loss: 0.1991 - val_acc: 0.9171
Epoch 3/20
197s - loss: 0.1513 - acc: 0.9396 - val_loss: 0.1475 - val_acc: 0.9406
Epoch 4/20
197s - loss: 0.1449 - acc: 0.9423 - val_loss: 0.1341 - val_acc: 0.9458
Epoch 5/20
197s - loss: 0.1403 - acc: 0.9441 - val_loss: 0.1319 - val_acc: 0.9482
Epoch 6/20
197s - loss: 0.1342 - acc: 0.9470 - val_loss: 0.1325 - val_acc: 0.9471
Epoch 7/20
197s - loss: 0.1311 - acc: 0.9483 - val_loss: 0.1296 - val_acc: 0.9478
Epoch 8/20
197s - loss: 0.1287 - acc: 0.9495 - val_loss: 0.1216 - val_acc: 0.9525
Epoch 9/20
197s - loss: 0.1253 - acc: 0.9512 - val_loss: 0.1199 - val_acc: 0.9528
Epoch 10/20
197s - loss: 0.1226 - acc: 0.9520 - val_loss: 0.1240 - val_acc: 0.9513
Epoch 11/20
197s - loss: 0.1216 - acc: 0.9523 - val_loss: 0.1215 - val_acc: 0.9529
Epoch 12/20
197s - loss: 0.1183 - acc: 0.9539 - val_loss: 0.1170 - val_acc: 0.9540
Epoch 13/20
197s - loss: 0.1172 - acc: 0.9542 - val_loss: 0.1160 - val_acc: 0.9544
Epoch 14/20
197s - loss: 0.1154 - acc: 0.9553 - val_loss: 0.1165 - val_acc: 0.9558
Epoch 15/20
197s - loss: 0.1148 - acc: 0.9555 - val_loss: 0.1159 - val_acc: 0.9541
Epoch 16/20
197s - loss: 0.1120 - acc: 0.9565 - val_loss: 0.1106 - val_acc: 0.9565
Epoch 17/20
197s - loss: 0.1103 - acc: 0.9569 - val_loss: 0.1185 - val_acc: 0.9545
Epoch 18/20
197s - loss: 0.1102 - acc: 0.9573 - val_loss: 0.1122 - val_acc: 0.9570
Epoch 19/20
197s - loss: 0.1092 - acc: 0.9575 - val_loss: 0.1100 - val_acc: 0.9576
Epoch 20/20
197s - loss: 0.1076 - acc: 0.9581 - val_loss: 0.1096 - val_acc: 0.9577
Train on 32383 samples, validate on 8096 samples
Epoch 1/10
199s - loss: 0.1019 - acc: 0.9604 - val_loss: 0.1056 - val_acc: 0.9595
```

```
Epoch 2/10
197s - loss: 0.0994 - acc: 0.9614 - val_loss: 0.1070 - val_acc: 0.9589
Epoch 3/10
197s - loss: 0.0985 - acc: 0.9618 - val_loss: 0.1066 - val_acc: 0.9589
Epoch 4/10
197s - loss: 0.0966 - acc: 0.9625 - val_loss: 0.1068 - val_acc: 0.9590
Train on 32383 samples, validate on 8096 samples
Epoch 1/5
200s - loss: 0.0930 - acc: 0.9637 - val_loss: 0.1042 - val_acc: 0.9600
Epoch 2/5
198s - loss: 0.0923 - acc: 0.9642 - val_loss: 0.1037 - val_acc: 0.9604
Epoch 3/5
197s - loss: 0.0919 - acc: 0.9642 - val_loss: 0.1035 - val_acc: 0.9605
Epoch 4/5
197s - loss: 0.0912 - acc: 0.9644 - val_loss: 0.1035 - val_acc: 0.9606
Epoch 5/5
198s - loss: 0.0915 - acc: 0.9644 - val_loss: 0.1034 - val_acc: 0.9607
0.911560763742
Start KFold number 4 from 5
Split train:  32383 32383
Split valid:  8096 8096
Train on 32383 samples, validate on 8096 samples
Epoch 1/20
204s - loss: 0.2665 - acc: 0.9061 - val_loss: 0.2599 - val_acc: 0.9044
Epoch 2/20
196s - loss: 0.1616 - acc: 0.9359 - val_loss: 0.1995 - val_acc: 0.9188
Epoch 3/20
196s - loss: 0.1520 - acc: 0.9397 - val_loss: 0.1524 - val_acc: 0.9391
Epoch 4/20
197s - loss: 0.1455 - acc: 0.9425 - val_loss: 0.1348 - val_acc: 0.9463
Epoch 5/20
197s - loss: 0.1390 - acc: 0.9450 - val_loss: 0.1341 - val_acc: 0.9474
Epoch 6/20
197s - loss: 0.1353 - acc: 0.9467 - val_loss: 0.1304 - val_acc: 0.9492
Epoch 7/20
197s - loss: 0.1310 - acc: 0.9484 - val_loss: 0.1210 - val_acc: 0.9526
Epoch 8/20
197s - loss: 0.1276 - acc: 0.9497 - val_loss: 0.1259 - val_acc: 0.9501
Epoch 9/20
197s - loss: 0.1247 - acc: 0.9512 - val_loss: 0.1212 - val_acc: 0.9528
Epoch 10/20
197s - loss: 0.1227 - acc: 0.9519 - val_loss: 0.1164 - val_acc: 0.9544
Epoch 11/20
197s - loss: 0.1206 - acc: 0.9529 - val_loss: 0.1198 - val_acc: 0.9539
Epoch 12/20
197s - loss: 0.1183 - acc: 0.9540 - val_loss: 0.1140 - val_acc: 0.9560
Epoch 13/20
197s - loss: 0.1165 - acc: 0.9543 - val_loss: 0.1118 - val_acc: 0.9568
Epoch 14/20
197s - loss: 0.1157 - acc: 0.9547 - val_loss: 0.1118 - val_acc: 0.9569
Epoch 15/20
197s - loss: 0.1139 - acc: 0.9557 - val_loss: 0.1128 - val_acc: 0.9562
Epoch 16/20
197s - loss: 0.1120 - acc: 0.9563 - val_loss: 0.1130 - val_acc: 0.9563
Epoch 17/20
197s - loss: 0.1114 - acc: 0.9567 - val_loss: 0.1128 - val_acc: 0.9571
Train on 32383 samples, validate on 8096 samples
Epoch 1/10
201s - loss: 0.1055 - acc: 0.9590 - val_loss: 0.1061 - val_acc: 0.9590
Epoch 2/10
197s - loss: 0.1030 - acc: 0.9598 - val_loss: 0.1068 - val_acc: 0.9586
```

```
197s - loss: 0.1030 - acc: 0.9598 - val_loss: 0.1068 - val_acc: 0.9586
Epoch 3/10
197s - loss: 0.1023 - acc: 0.9601 - val_loss: 0.1065 - val_acc: 0.9589
Epoch 4/10
197s - loss: 0.1008 - acc: 0.9608 - val_loss: 0.1065 - val_acc: 0.9593
Train on 32383 samples, validate on 8096 samples
Epoch 1/5
199s - loss: 0.0959 - acc: 0.9625 - val_loss: 0.1040 - val_acc: 0.9602
Epoch 2/5
198s - loss: 0.0953 - acc: 0.9628 - val_loss: 0.1038 - val_acc: 0.9604
Epoch 3/5
197s - loss: 0.0948 - acc: 0.9630 - val_loss: 0.1037 - val_acc: 0.9605
Epoch 4/5
197s - loss: 0.0951 - acc: 0.9631 - val_loss: 0.1036 - val_acc: 0.9604
Epoch 5/5
197s - loss: 0.0947 - acc: 0.9632 - val_loss: 0.1033 - val_acc: 0.9606
0.911169446439
Start KFold number 5 from 5
Split train:  32384 32384
Split valid:  8095 8095
Train on 32384 samples, validate on 8095 samples
Epoch 1/20
206s - loss: 0.2679 - acc: 0.9046 - val_loss: 0.2616 - val_acc: 0.9057
Epoch 2/20
197s - loss: 0.1596 - acc: 0.9374 - val_loss: 0.2012 - val_acc: 0.9148
Epoch 3/20
197s - loss: 0.1516 - acc: 0.9405 - val_loss: 0.1475 - val_acc: 0.9407
Epoch 4/20
197s - loss: 0.1422 - acc: 0.9433 - val_loss: 0.1362 - val_acc: 0.9461
Epoch 5/20
197s - loss: 0.1361 - acc: 0.9466 - val_loss: 0.1281 - val_acc: 0.9503
Epoch 6/20
197s - loss: 0.1320 - acc: 0.9482 - val_loss: 0.1271 - val_acc: 0.9499
Epoch 7/20
197s - loss: 0.1295 - acc: 0.9493 - val_loss: 0.1228 - val_acc: 0.9512
Epoch 8/20
197s - loss: 0.1257 - acc: 0.9509 - val_loss: 0.1180 - val_acc: 0.9541
Epoch 9/20
197s - loss: 0.1224 - acc: 0.9520 - val_loss: 0.1179 - val_acc: 0.9540
Epoch 10/20
197s - loss: 0.1201 - acc: 0.9532 - val_loss: 0.1199 - val_acc: 0.9534
Epoch 11/20
197s - loss: 0.1180 - acc: 0.9541 - val_loss: 0.1153 - val_acc: 0.9555
Epoch 12/20
197s - loss: 0.1161 - acc: 0.9548 - val_loss: 0.1101 - val_acc: 0.9574
Epoch 13/20
197s - loss: 0.1153 - acc: 0.9553 - val_loss: 0.1175 - val_acc: 0.9543
Epoch 14/20
197s - loss: 0.1142 - acc: 0.9556 - val_loss: 0.1121 - val_acc: 0.9564
Epoch 15/20
197s - loss: 0.1119 - acc: 0.9568 - val_loss: 0.1082 - val_acc: 0.9577
Epoch 16/20
197s - loss: 0.1107 - acc: 0.9569 - val_loss: 0.1113 - val_acc: 0.9567
Epoch 17/20
197s - loss: 0.1089 - acc: 0.9578 - val_loss: 0.1109 - val_acc: 0.9577
Epoch 18/20
197s - loss: 0.1092 - acc: 0.9575 - val_loss: 0.1060 - val_acc: 0.9594
Epoch 19/20
197s - loss: 0.1074 - acc: 0.9585 - val_loss: 0.1105 - val_acc: 0.9584
Epoch 20/20
197s - loss: 0.1055 - acc: 0.9590 - val_loss: 0.1096 - val_acc: 0.9583
```

```
Train on 32384 samples, validate on 8095 samples
Epoch 1/10
199s - loss: 0.1006 - acc: 0.9609 - val_loss: 0.1029 - val_acc: 0.9603
Epoch 2/10
197s - loss: 0.0980 - acc: 0.9620 - val_loss: 0.1053 - val_acc: 0.9599
Epoch 3/10
197s - loss: 0.0968 - acc: 0.9623 - val_loss: 0.1044 - val_acc: 0.9606
Epoch 4/10
197s - loss: 0.0956 - acc: 0.9631 - val_loss: 0.1049 - val_acc: 0.9596
Train on 32384 samples, validate on 8095 samples
Epoch 1/5
199s - loss: 0.0925 - acc: 0.9640 - val_loss: 0.1021 - val_acc: 0.9608
Epoch 2/5
197s - loss: 0.0905 - acc: 0.9648 - val_loss: 0.1017 - val_acc: 0.9610
Epoch 3/5
197s - loss: 0.0902 - acc: 0.9650 - val_loss: 0.1013 - val_acc: 0.9611
Epoch 4/5
197s - loss: 0.0897 - acc: 0.9650 - val_loss: 0.1011 - val_acc: 0.9611
Epoch 5/5
197s - loss: 0.0896 - acc: 0.9650 - val_loss: 0.1010 - val_acc: 0.9613
0.914830954708
```

Out[5]:

	blow_down	bare_ground	conventional_mine	blooming	cultivation	artisinal_mine	
0	0.000412	0.000431	4.006761e-05	0.007126	0.002437	5.980067e-05	0.0
1	0.004837	0.000821	5.298111e-05	0.032775	0.010383	8.335907e-05	0.0
2	0.000223	0.000359	2.351456e-05	0.000252	0.007065	2.014326e-05	0.0
3	0.011162	0.010321	3.841159e-04	0.017751	0.239849	5.601675e-04	0.0
4	0.000309	0.002546	1.565049e-04	0.000248	0.008930	4.684106e-04	0.0
5	0.000212	0.000275	2.246654e-05	0.002734	0.002268	3.124239e-05	0.0
6	0.001810	0.025815	8.963640e-04	0.003969	0.527947	1.332393e-03	0.3
7	0.000002	0.028016	2.869982e-04	0.000014	0.003440	2.798877e-04	0.0
8	0.000820	0.000653	4.006185e-05	0.004969	0.005208	5.331761e-05	0.0
9	0.000473	0.010207	1.111569e-04	0.000915	0.319717	3.572095e-05	0.8
10	0.003308	0.016970	5.077684e-04	0.000731	0.070325	6.386618e-04	0.0
11	0.001540	0.021285	1.576447e-05	0.005150	0.945986	2.188746e-04	0.0
12	0.000006	0.000115	7.319985e-06	0.000015	0.001432	5.729397e-06	0.0
13	0.002904	0.026400	3.995568e-04	0.003803	0.370646	2.865250e-04	0.0
14	0.003191	0.017730	1.881389e-04	0.006595	0.485887	2.131831e-04	0.0
15	0.000391	0.000509	4.087621e-05	0.004733	0.002941	6.201131e-05	0.0
16	0.000383	0.022844	1.199553e-03	0.000469	0.387344	3.658846e-04	0.0
17	0.000482	0.000945	4.282291e-05	0.002652	0.012538	7.857783e-05	0.0
18	0.000062	0.000537	2.615232e-05	0.000761	0.005911	1.852325e-05	0.3
19	0.000548	0.001658	2.622981e-04	0.008388	0.020334	2.529515e-04	0.0

	blow_down	bare_ground	conventional_mine	blooming	cultivation	artisinal_mine	
20	0.000943	0.025229	1.028834e-03	0.002203	0.136361	2.263383e-03	0.(
21	0.000168	0.020353	5.232099e-04	0.000597	0.066675	2.167202e-03	0.(
22	0.000326	0.018148	1.281021e-03	0.000511	0.217176	4.841637e-04	0.(
23	0.000005	0.000096	5.732509e-06	0.000013	0.001449	4.131970e-06	0.1
24	0.000480	0.024394	2.588494e-03	0.000110	0.051427	6.046075e-04	0.(
25	0.000397	0.000327	3.112931e-05	0.006371	0.002277	5.150873e-05	0.(
26	0.000009	0.001493	7.370555e-04	0.000029	0.120794	2.239215e-06	0.(
27	0.003164	0.133461	9.065690e-04	0.001051	0.101599	1.736401e-04	0.(
28	0.000229	0.049573	7.901724e-03	0.000285	0.080243	3.655500e-04	0.(
29	0.000522	0.000573	3.351616e-05	0.003474	0.003545	4.760017e-05	0.(
...	...	...	...	...	...	...	...
61161	0.010137	0.021169	9.853866e-05	0.007608	0.571631	2.043044e-04	0.(
61162	0.000769	0.031433	9.654845e-04	0.000850	0.292621	7.841937e-04	0.(
61163	0.006027	0.002663	8.275602e-05	0.017912	0.019470	1.489638e-04	0.(
61164	0.000604	0.000522	6.034400e-05	0.008177	0.005836	7.977312e-05	0.(
61165	0.002575	0.019589	2.920084e-05	0.003758	0.821756	7.359782e-05	0.(
61166	0.000191	0.000281	2.249257e-05	0.002414	0.002210	2.997483e-05	0.(
61167	0.000039	0.000011	4.171416e-07	0.000139	0.001177	3.973999e-07	0.(
61168	0.000381	0.000396	3.850194e-05	0.005892	0.002582	5.400659e-05	0.(
61169	0.002261	0.000911	9.060156e-05	0.027668	0.004929	1.491399e-04	0.(
61170	0.000186	0.000278	2.378689e-05	0.002463	0.002111	3.245121e-05	0.(
61171	0.001302	0.026751	1.648872e-04	0.000965	0.252351	5.158713e-05	0.(
61172	0.001141	0.014738	1.354928e-03	0.002021	0.244238	2.683896e-04	0.(
61173	0.003550	0.000784	5.237614e-05	0.055194	0.004683	9.762375e-05	0.(
61174	0.000181	0.000268	2.204912e-05	0.002409	0.002120	2.922291e-05	0.(
61175	0.000203	0.000281	2.380411e-05	0.002827	0.002118	3.160100e-05	0.(
61176	0.000189	0.000287	2.424404e-05	0.002499	0.002200	3.281612e-05	0.(
61177	0.000010	0.003157	1.594522e-04	0.000034	0.123500	1.026254e-06	0.(
61178	0.001579	0.000594	5.024246e-05	0.018384	0.003149	9.323742e-05	0.(
61179	0.000448	0.026523	3.773913e-03	0.000723	0.148589	3.633686e-04	0.(
61180	0.001608	0.131366	7.577287e-03	0.000867	0.085494	8.370895e-04	0.(
61181	0.001317	0.000633	7.688433e-05	0.026337	0.003747	9.714971e-05	0.(
61182	0.000463	0.008893	7.125403e-04	0.001307	0.375590	4.195791e-04	0.(
61183	0.000186	0.000269	2.326754e-05	0.002599	0.002044	3.175418e-05	0.(
61184	0.000062	0.007190	2.374286e-05	0.000232	0.026971	6.936490e-05	0.(

	blow_down	bare_ground	conventional_mine	blooming	cultivation	artisinal_mine	
61185	0.000113	0.004319	1.018951e-04	0.000122	0.005692	3.860476e-04	0.(
61186	0.000112	0.000912	5.367146e-05	0.000176	0.004339	1.174276e-04	0.(
61187	0.000139	0.008706	6.182826e-05	0.001066	0.041362	4.527542e-04	0.(
61188	0.001842	0.005619	3.325673e-04	0.008020	0.016727	4.828009e-04	0.(
61189	0.000004	0.000053	3.983095e-06	0.000016	0.000591	3.258744e-06	0.(
61190	0.000010	0.028870	3.497869e-04	0.000052	0.005841	2.512229e-04	0.(

61191 rows × 17 columns

In [6]:

```
from tqdm import tqdm
thres = [0.07, 0.17, 0.2, 0.04, 0.23, 0.33, 0.24, 0.22, 0.1, 0.19, 0.23,
0.24, 0.12, 0.14, 0.25, 0.26, 0.16]
preds = []
for i in tqdm(range(result.shape[0]), miniters=1000):
    a = result.ix[[i]]
    a = a.apply(lambda x: x > 0.2, axis=1)
    a = a.transpose()
    a = a.loc[a[i] == True]
    ' '.join(list(a.index))
    preds.append(' '.join(list(a.index)))

df_test['tags'] = preds
df_test.to_csv('submission_keras_5_fold_CV_0.9136_LB_0.913.csv', index=Fals
e)
## 0.913
```

```
  0%|
| 0/61191 [00:00<?, ?it/s]D:\Anaconda3\lib\site-
packages\ipykernel\__main__.py:5: DeprecationWarning:
.ix is deprecated. Please use
.loc for label based indexing or
.iloc for positional indexing

See the documentation here:
http://pandas.pydata.org/pandas-docs/stable/indexing.html#deprecate_ix
100%|
61191/61191 [02:07<00:00, 479.33it/s]
```