

Forum:[Hack Forums](#)

Thread: [\[sklearn\]Using OpenCV-SURF and Nerual Network for object detection.](#)

severos - December 1, 2017, 1:07 pm

I'm working on this project for university, and I'm having some problems here, would be nice if someone could lend me a hand with it.

The main idea is object recognition, detection algorithm used is SURF/SIFT, classifier is Neural Network, and this is the used dataset: [kaggle.com/c/cifar-10](https://www.kaggle.com/c/cifar-10).

Far as I read and understand:

- 1- Read all images and extract descriptors/keypoints from them.
- 2- ANN input is the descriptor, and output is label

Here's the current code, I've added comments to explain what I'm trying to do in each part:

Spoiler ([Click to View](#))

PHP Code

```
<?php
import cv2
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import classification_report

scaler = StandardScaler()
surf = cv2.xfeatures2d.SURF_create(300,upright=True, extended = True)

def read_train_image_by_id(id):
    return cv2.imread('./train/'+str(id)+'.png')

def read_test_image_by_id(id):
    return cv2.imread('/test/'+str(id)+'.png')

def read_train_data(df):
    #should append descriptors and keypoints to the dataframe
    descriptors = []
    key_points = []
    for x in df.iterrows():
        img = read_train_image_by_id(x[1]['id'])
        kp, des = surf.detectAndCompute(img,None)
        descriptors.append(des)
        key_points.append(kp)
    df['descriptors'] = pd.Series(descriptors)
    df['keyPoints'] = pd.Series(key_points)
    return
#initial values
neurons_per_layer = (128,)
hidden_layer_increment = 1
current_accuracy = 0.
```

```
min_accepted_accuracy = 0.9
learning_rate = 1e-5
```

```
print('neurons per layer: '+str(neurons_per_layer))
print('hidden layers increment: '+str(hidden_layer_increment))
print('learning rate:'+str(learning_rate))
```

```
#load training data, it contains the ID and correct label
data = pd.read_csv('./trainLabels.csv')
#convert label field from string to number so it works with neural network
data['label'], labelMap = pd.factorize(data['label'])
print('got: '+str(len(labelMap))+ ' object labels')
print('reading train data')
#fill dataframe with the actual image data
read_train_data(data)
data = data[(data['descriptors'].notnull())]
print('done reading data, splitting them to train/test sets')
```

```
#split data to train and test
train, test = train_test_split(data, test_size=0.2)
```

```
##following part isn't working, some why fit doesn't work, it throws an exception.
#scaler.fit(train['descriptors'])
#train['descriptors'] = scaler.transform(train['descriptors'])
#test['descriptors'] = scaler.transform(test['descriptors'])
#X = train['descriptors'] is the input for training network
#Y = train['label'] is the correct labels for training network.
```

```
iteration = 0
while current_accuracy < 0.9:
#keep increasing hidden layer count until reaching acceptable accuracy
    iteration+=1
    print('iteration: '+str(iteration))
    clf = MLPClassifier(
        hidden_layer_sizes=neurons_per_layer*(iteration*hidden_layer_increment),
        activation='tanh',
        solver='lbfgs',
        alpha=learning_rate,
        random_state=1337
    )
#an exception is being thrown here: ValueError: setting an array element with a sequence.
    clf.fit(train['descriptors'], train['label'])
    predictions = clf.predict(test['descriptors'])
    result = classification_report(test['label'],predictions)
    print(result)
    #check result and determine what is the accuracy, then updated it.
    input('click to start next iteration...')
```

```
tl;dr
1- StandardScaler isn't working with me, would it be a problem?
2- Is it correct to feed the ANN with descriptor from the SURF detection algorithm? if so, why is my
code throwing an exception?
3- Any extra info/tips would be awesome to learn.
```

