

myfirstSVM

December 21, 2023

```
[1]: import os
import cv2 as cv
import numpy as np
from PIL import Image
from sklearn import svm
import pandas as pd
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV
```

```
[2]: def load_images_from_folder(folder):
    images = []
    for file in os.listdir(folder):
        img = cv.imread(os.path.join(folder,file), cv.IMREAD_GRAYSCALE)
        if(img is not None):
            images.append(img)
    return images
```

```
[3]: def resize_and_flatten(img, target_size=(64,64)):
    resized_img = cv.resize(img, target_size)
    flattened_img = resized_img.reshape(target_size[0] * target_size[1])
    return flattened_img
```

```
[4]: cat_data = []
for img in load_images_from_folder("D:/git/bachelor-thesis/datasets/cat_v_dog/
↪Cat"):
    cat_data.append(resize_and_flatten(img))
dog_data = []
for img in load_images_from_folder("D:/git/bachelor-thesis/datasets/cat_v_dog/
↪Dog"):
    dog_data.append(resize_and_flatten(img))
```

```
[5]: print(str(len(cat_data)) + " cat images")
print(str(len(dog_data)) + " dog images")
```

772 cat images

772 dog images

```
[6]: X = []
      y = []

      for entity in cat_data:
          X.append(entity)
          y.append(1)

      for entity in dog_data:
          X.append(entity)
          y.append(0)

[7]: classifier = svm.SVC()

[9]: gs = GridSearchCV(classifier, {}, cv=3)
      gs.fit(X,y)
      pd.DataFrame(gs.cv_results_)

[9]:  mean_fit_time  std_fit_time  mean_score_time  std_score_time  params  \
0      1.859278      0.018411      1.427994      0.021008      {}

      split0_test_score  split1_test_score  split2_test_score  mean_test_score  \
0      0.504854      0.549515      0.558366      0.537578

      std_test_score  rank_test_score
0      0.02342      1

[ ]:
```