

A.I. Assignment 8

A simple face detection

Build a simple application that receives an image and test if there is the image of a human face or not.

For this assignment you have three tasks.

- I. Get familiarized with PIL/PILLOW (python image library) - for 20 points
- II. Preparing the database for training the network - for 50 points
- III. Create and train a network with the images from the database - for 130p

Task 1

Install Pillow is is not already installed and play with an image (f1.jpg for example)

```
pip install Pillow
```

A. Display an image with pillow:

```
# load and show an image with Pillow
from PIL import Image
# load the image
image = Image.open('opera_house.jpg')
# summarize some details about the image
print(image.format)
print(image.mode)
print(image.size)
# show the image
image.show()
```

B. Convert the image to a numpy array:

```
# load and display an image with Matplotlib
from matplotlib import image
from matplotlib import pyplot
# load image as pixel array
data = image.imread('opera_house.jpg')
# summarize shape of the pixel array
print(data.dtype)
print(data.shape)
```

```
# display the array of pixels as an image
pyplot.imshow(data)
pyplot.show()
```

C. Resize an image to a specific dimension:

```
# create a thumbnail of an image
from PIL import Image
# load the image
image = Image.open('opera_house.jpg')
# report the size of the image
print(image.size)
# create a thumbnail and preserve aspect ratio
image.thumbnail((100,100))
# report the size of the thumbnail
print(image.size)
```

D. Other operations can be found at the address [How to Load and Manipulate Images for Deep Learning in Python With PIL/Pillow](#)

Task 2

Build your database with images of human faces and others that do not contain faces. Download from the internet 50 faces of men, 50 of women, 50 of something else that have no faces on it. Because it will be a small database, take the images of faces that look like the passport photos (front face, no sides).

The dataset class should be something like:

```
import torch
import torch.nn as nn
import torch.optim as optim
import time

from torch.utils.data import Dataset, DataLoader
from torchvision import datasets, models, transforms

device = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')

class ImageClassifierDataset(Dataset):

    def __init__(self, image_list, image_classes):
        self.images = []
        self.labels = []
```

```

        self.classes = list(set(image_classes))
        self.class_to_label = {c: i for i, c in
                                enumerate(self.classes)}

        self.image_size = 224
        self.transforms = transforms.Compose([
            transforms.Resize(self.image_size),
            transforms.CenterCrop(self.image_size),
            transforms.ToTensor(),
            transforms.Normalize((0.5,0.5,0.5), (0.5,0.5,0.5))
        ])
        for image, image_class in zip(image_list, image_classes):

            transformed_image = self.transforms(image)
            self.images.append(transformed_image)

            label = self.class_to_label[image_class]
            self.labels.append(label)

    def __getitem__(self, index):
        return self.images[index], self.labels[index]

    def __len__(self):
        return len(self.images)

```

For your problem the `image_list` should be the images loaded with the PIL module from the images downloaded to the hard disk. Feel free to add methods and attributes but **DO NOT CHANGE** the `__getitem__` and the `__len__` methods (they are mandatory).

An instance of this class will hold all the images with their labels (1 - is a face, 0 - is not a face).

Task 3

Create the model and train it with the data from your dataset.

In the file `exampleCV.py` is an example that follows the tutorial from [Basics of Image Classification with PyTorch](#). (with some modifications in order to run)

In order to use this model, one has to **change** the dataset loaded in the example with the one created by us and **adjust all the parameters** in order to work.

To change the dataset, divide the images in 2 (train and test), make two objects of `ImageClassifierDataset` and replace with them the `train_set` and `test_set` in the `train_loader` and `test_loader`.

For this assignment one can get a maximum **200** points.

Due time:

Task 1 and Task 2 in one week.

Task 3 in the last week of the semester.

IF nothing is done in the first week you will have a penalty of 20 points.