

Neural network for image recognition

SEBASTIAN ROCHACKI

1 INTRODUCTION

The program is written in Python and is based on a neural network. The main goal of the program is a prediction for loaded photo. The categories (classes) in which the program is able to classify the object in the picture:

- T-shirt
- Trouser
- Pullover
- Dress
- Coat
- Sandal
- Shirt
- Sneakre
- Bag
- Ankle boot

For this purpose, I use free and open-source software library Tensor Flow. Image.py works with an accuracy of about 85% and has be trained on the basis 6,000 photos (fashion, kreas model). In addition, image.py learn to recognize more, if the program returns prediction for photo.png smaller than 50% for one of categories, then user can indicate what is in the picture.

Photos are loaded in .png format, the selection rule for photos is that their pixel number should be $N \times N$, where $N \geq 28$. In the case of, for example, photos with a higher quality, the program will reduce it to 28 by 28 pixels. For simplicity, when loaded photo is close to square, image.py has function which checks it and for 10% difference of rectangle sides return square. So, the loaded photo has not to be ideal square.

2 RUN

Image.py is in the main folder. Additionally, in the main folder, there is a few another path

- /picture, includes examples of photos, here should be photos which we want to use
- /results, folder contains one results.dat with the maximum prediction for each loaded photo and full predictions
- /28x28, it is a folder with uploaded photos prepared by image.py for classification

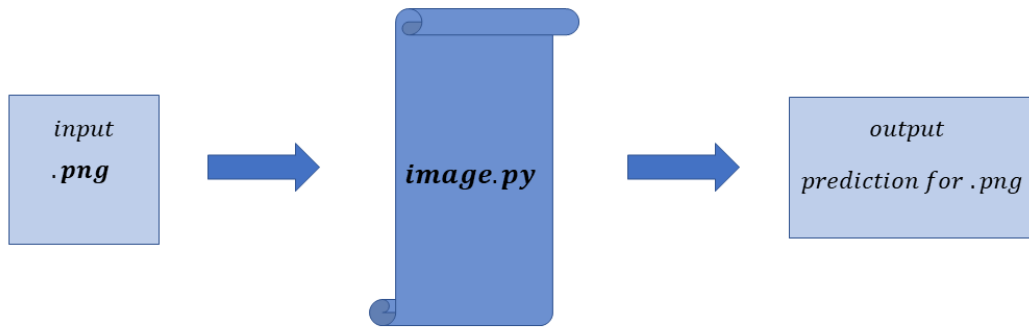


Figure 2.1: scheme

There is a couple option to use program, by:

- **name**, then user writes name of photo with its type, e.g. 'picture_1.png'
- **series**, then user writes number of photos in folder /pictures, which are named according to template 'picture_i.png'
- **name + re-train**, option if the prediction is wrong

Please, notice that neural-network has 85% accuracy so sometimes prediction can be not perfect. In that case, user can use option **name + re-train**.

Last program option is 'reset'. After use, image.py saves output model for basic test dataset.

EASY TEST FOR THE PROGRAM IS BY CHOOSE OPTION '2' = 'SERIES' AND E.G. RUN PROGRAM FOR $i = 8$.