

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, it has been used free and open-source software Tensor Flow. The program predictions work with an accuracy of about 85% and have be trained on the basis 6,000 photos (fashion, kreas model). In addition, the program learns 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 photos with a higher quality, the program will reduce it to 28 by 28 pixels. For simplicity, when loaded photo is close to square, the program 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 CONTENTS AND PATHS

The main program (image.py) is in the main folder. Additionally, there are a few another path:

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

3 ENVIRONMENT AND REQUIRED LIBRARIES

Project has been created in Python 3.7.5 so it's a recommended version of Python. Before the use, it should be installed the following libraries:

- Tensor Flow 2.0.0
- Numpy 1.17.4
- Matplotlib
- PIL 6.2.1
- Argparse

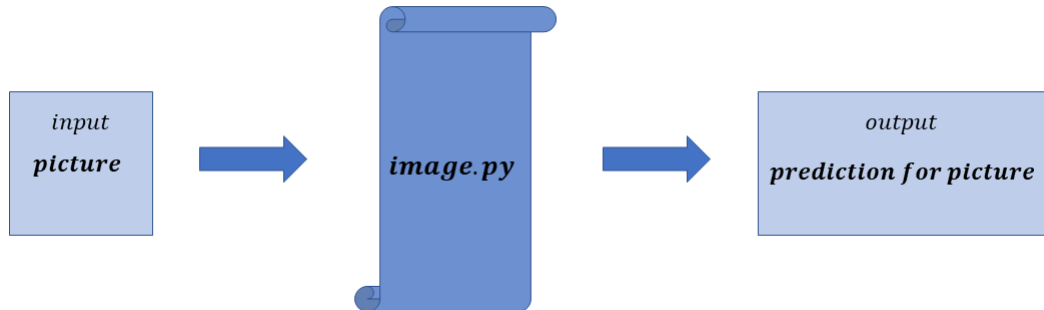


Figure 4.1: scheme

Options of use the program:

- **name**, next, user writes name of photo with its type, e.g. 'picture_1.png'
- **name+**, option if the prediction is wrong, then user can do a re-train by add image to datasets
- **series**, then user writes number of photos in folder /pictures, which are named according to template 'picture_i.png'
- **reset**, then program resets neural network to initial settings (by the use standard datasets)

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

Basic commands:

- **python image.py -help** or **python image.py -h**
help option
- **python image.py -m name -ns**
'-ns' if user don't want to save images, e.g a lot of pictures
- **python image.py -m series -n 7**
a complete easy test for the program