Neural network based image rotation

Nadav Halahmi, supervisor: Shlomo Rozenfeld

Introduction

Unwanted rotated photos are very common and fixing this rotation can be time consuming. The origin of the problem is based on two factors:

- The need to take pictures horizontally and vertically.
- Holding the camera upside down unintentionally

In this project, I created a neural network, based on 100 sample pictures I took. The neural network can fix a picture's rotation, with a high rate of success for landscape pictures.

Identification Method

To identify the right rotation of each picture I developed a system based on python, using the libraries PIL (python image library) for image processing and Pybrain, scikit-learn and pickle for building the optimal neural network.

Initially I picked 100 diverse pictures. Each photo was rotated to all of four possible rotations (0-3 rotations clockwise). Then the resulting 400 photos were shuffled to create a big and diverse enough database for training and testing the network. 75% of the photos were used for training several neural networks and the other 25% for testing them.

Image preparation and creation of the database

Each neural network required each image to be converted to a vector, based on its pixels. Each photo's resolution was reduced to different resolutions and was represented in one of two types (based on the structure of each network). Type 1 was the RGB data of each pixel, displayed in a vector. Type 2 was the average RGB of each pixel displayed in a smaller vector with the same resolution. Each database was used to create a unique type of network and the best network was chosen.

Neural network training

Each database was the base of several neural networks, and the difference between those networks was reflected by the number of neurons on the hidden layer. The input layer was based on the resolution and type of the pictures each database. The number of neurons on the output layer was 4 for each network, for each possible rotation of a picture. The best neural network for each database was chosen, and then I picked the best network of the ones chosen.

Results

The chosen network was for 4X3 or 3X4 RGB pictures. It had 90% chance of success on the test group, and 80% chance of success on random landscape pictures from google. Other networks successfully rotated between 60% and 85% of the test group, with a clear advantage for the RGB networks over the one-color networks. The program is very user-friendly: the user gives the program an import folder for the mixed pictures, and an export folder for the fixed one. He then simply runs the program and waits a few seconds for the fixed photos, thus saving him hours of sitting by the computer and rotating each photo.

BEFORE

AFTER
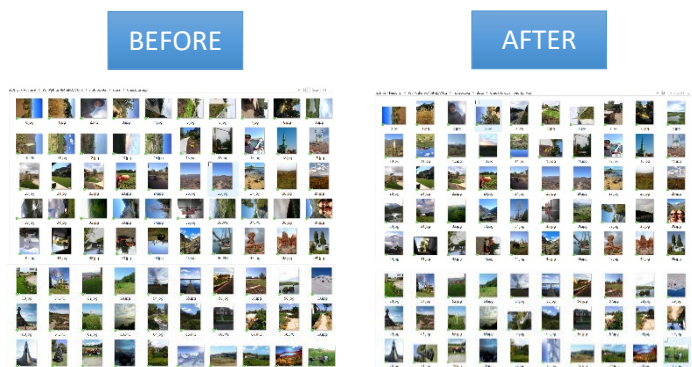
For Further Research

- Improve the user interface by using the context menu (right click menu) for picking the import and export folders, and running the program.
- Improve results by using much more photos for training the network and/or using already made networks for object identification. Likewise, more powerful computers will lead for better results.