Machine Learning Engineer Nanodegree
Capstone Proposal

Egor Ermilov
2017-06-21

Proposal

Domain Background

Image recognition is a classical problem in computer vision. It solves many different problems such as human-computer interaction, detection, control, navigation, etc.

Deep Neural Networks are very efficient at finding difficult patterns in the data (including image recognition). Adding a convolutional layer makes a huge impact on the neural networks and their abilities to identify objects on an image.

As we already saw from the previous Udacity Machine Learning Nanodegree project, a neural network with two convolutional layers could identify objects with hight accuracy.

Problem Statement

However, the goal of this project is to train a deep neural network to recognize human emotions. The model should be able to recognize emotions by a given human face photo with high accuracy. It should also be able to use not only the test photos from the original dataset, but also any custom photo uploaded.

Datasets and Inputs

This project will use the dataset from the Emotion and identity detection from face images Kaggle competition (https://inclass.kaggle.com/c/facial-keypoints-detector/data).

The dataset consists of 3,761 gray-scaled images of 48x48 pixels in size and a 3,761 label set of seven elements each.

Each element encodes an emotional stretch:
0 = anger,
1 = disgust,
2 = fear,
3 = happy,
4 = sad,
5 = surprise,
6 = neutral.

Solution Statement

For this project a convolution neural network will be built and trained on the Kaggle dataset. The application will be written in Python and use Tensorflow for building the model.

The approximate architecture of the network would be the following:

– Input layer
– Convolutional layer 1
– Pooling layer 1
– Convolutional layer 2
– Pooling layer 2
– Fully connected layer 1
– Fully connected layer 2
– Output layer

Benchmark Model

In the field of emotion recognition it is hard to be 100% accurate. People themselves don't always understand their emotions, let alone computers.

However, most of the reported models have a pretty high accuracy of 70-80%
For example:
http://www.paulvangent.com/2016/04/01/emotion-recognition-with-python-opencv-and-a-face-dataset/

The original Kaggle competition winner had an accuracy of 50%:
https://inclass.kaggle.com/c/facial-keypoints-detector/leaderboard

Evaluation Metrics

The first evaluation metric would be accuracy, which is the percentage of correctly identified photos.

The second would be manual – upload a custom facial photo, run the application and evaluate the result.

Project Design

Firstly, the dataset will be downloaded and preprocessed. It would be transformed into numpy arrays, normalized and split into training, testing and validation datasets.

Then a neural network model will be built. The architecture is described above. The model will be trained on the training dataset. The performance will be evaluated every epoch on the validation dataset, and finally on the testing dataset.

Once the model is trained and evaluated, it would be saved.

The saved model would be used to upload a custom photo and evaluate the performance.