Illinois Institute of Technology

Project Report

Gender and Age Classification using Facial Features

Submitted by

Harsh Parikh
hparik11@hawk.iit.edu

Abstract

The principal objective of this project is to develop methods for the estimation of the gender and the age of a person based on a facial image. The extracted information can be useful in, for example, security or commercial applications. This is a difficult estimation problem, since the only information we have is the image, that is, the looks of the person. The training process needs to be optimized in terms of pre-processing, feature selection, choice of classifier and parameters. Automatic age and gender classification has become relevant to an increasing amount of applications, particularly since the rise of social platforms and social media. Nevertheless, performance of existing methods on real-world images is still significantly lacking, especially when compared to the tremendous leaps in performance recently reported for the related task of face recognition. In this paper we show that by learning representations through the use of deep-convolutional neural networks (CNN), a significant increase in performance can be obtained on these tasks.

1. Introduction

Over the last decade, the rate of image uploads to the Internet has grown at a nearly exponential rate. Some of the most basic classifications regarding humans are gender and age. They are among the very first things a person decides on sight of someone. These decisions are based on many different features strictly coming from the person, but also from the environment. The marketing or sales departments of companies are usually interested in their products' targeted customers. Thus, it is important in many fields to have statistics of the target audience of their products. In the same way, some services, permissions or products are only allowed to an audience of certain gender or age, and it has to be somehow controlled. Bringing together the two ideas mentioned above, the need of estimating the gender and age in some automatic way appears. A human can easily make these estimates from faces. Yet, it is still a challenging task for a computer. This project is focused on gender and age estimation based on face images using computer vision techniques Applications for these systems include everything from suggesting who to "tag" in Facebook photos to pedestrian detection in self-driving cars. However, the next major step to take building off of this work is to ask not only how many faces are in a picture and where they are, but also what characteristics do those faces have. The goal of this project do exactly that by attempting to classify the age and gender of the faces in an image. Social media websites like Facebook could use the information about the age and gender of the people to better infer the context of the image. For example, if a picture contains many people studying together, Facebook might be able to caption the scene with "study session." However, if it can also detect that the people are all men in their early 20s and that some are wearing shirts with the same letters, it may predict "College students in a fraternity studying." Age and gender classification is an inherently challenging problem though, more so than many other tasks in computer vision. The main reason for this discrepancy in difficulty lies in the nature of the data that is needed to train these types of systems.

2. Related Work

The areas of age and gender classification have been studied for decades. Various different approaches have been taken over the years to tackle this problem, with varying levels of success.

Early methods for age estimation are based on calculating ratios between different measurements of facial features. Once facial features (e.g. eyes, nose, mouth, chin, etc.) are localized and their sizes and distances measured, ratios between them are calculated and used for classifying the face into different age categories according to hand-crafted rules.

In this study, similar approaches are developed for gender and age estimation which could be exploited to develop a more general system that can perform both tasks. First, face and eye detection are performed on the input image. After detection, alignment based on eye coordinates of the detected face image is applied to scale and translate face and reduces in feature space. Aligned face image is divided into local blocks and discrete cosine transform is performed on these local blocks. Concatenating features of each block, an overall feature vector is obtained. In gender estimation, SVM classifier is used for binary classification between female and male. In age estimation, a two-step classifier is used. In the first step, SVM classifier is used to discriminate between youth and adult and in the second step, support vector regression (SVR) is used for youth, adult and global age estimation to determine the specific age.

In recent years, with the dawn of never-before seen fast and cheap compute, revived the interest in CNNs showing that deep architectures are now both feasible and effective, and continued to increase the depth of such networks to show even better performance. They advocate for a relatively shallow network, however, in order to prevent over-fitting the relatively small dataset they were operating on. Deeper networks, although generally more expressive, also have a greater tendency to fit noise in the data. So while improved performance with deeper architectures training on millions of images, shows improvements for shallower architectures for their use case.

3. Data

We collected data from LinkedIn profiles. Initially took 4 different LinkedIn IDs and recursively get other profiles based on LinkedIn recommended id list. So one by one, we scraped profiles and predict age from his/her school year and find gender based on first name using data.gov website. Sometime, it may possible that we get same Male and Female name so we ignore those profiles. We have put some threshold value to discriminate two genders. For age, we hardly find people put age/birth date on LinkedIn. So to calculate age, we find profiles' school/bachelor year and if we get it then we can add +18 and that we consider it as an age of a person. Sometimes we don't get bachelor degree information then we search for Masters or higher degree and calculate age based on that.

So we have collected around 8000 LinkedIn profiles and out of which 3000 have all useful information i.e. age and gender. In this profiles, we have around 1448 Female and 1652 Male. Most of the profiles' age fall between 20-35.

-

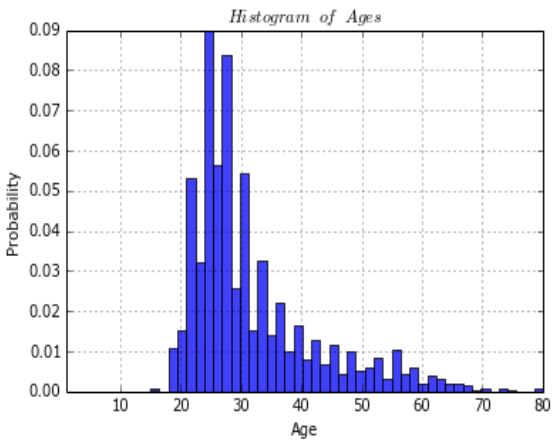

Fig 1. Histogram of profiles' age range

Fig 2. image dataset examples. Top row: 5 males of various ages. Bottom row: 5 females of various ages.

4. Method

The first thing to do while facing a big problem is to analyse it and divide it into different specific parts. A generic problem can be split into many steps with specific functions, so a tedious work is divided into small and well-defined parts.

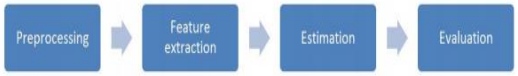

Fig 3. Steps for Classification

An RGB image being input to the network is first scaled to 3x400x400 and then resized to 3x100x100. Then we have converted RGB image into Grey Scale image. So input size would be 100*100. There are 3 convolution layers, followed by 3 fully connected layers.

4.1 Logistic Regression

Logistic regression is a probabilistic, linear classifier. It is parametrized by a weight matrix W and a bias vector b. Classification is done by projecting an input vector onto a set of hyperplanes, each of which corresponds to a class. The distance from the input to a hyperplane reflects the probability that the input is a member of the corresponding class. Mathematically, the probability that an input vector x is a member of a class i, a value of a stochastic variable Y, can be written as:

$$P(Y = i|x, W, b) = softmax_i(Wx + b) = \frac{e^{W_i x + b_i}}{\sum_j e^{W_j x + b_j}}$$

The model's prediction y_{pred} is the class whose probability is maximal, specifically:

$$y_{pred} = \text{argmax}_i P(Y = i|x, W, b)$$

4.2 Convolution Neural Networks (ConvNets)

A Convolutional Neural Network (CNN) is comprised of one or more convolutional layers (often with a subsampling step) and then followed by one or more fully connected layers as in a standard multilayer neural network. The architecture of a CNN is designed to take advantage of the 2D structure of an input image (or other 2D input such as a speech signal). This is achieved with local connections and tied weights followed by some form of pooling which results in translation invariant features. Another benefit of CNNs is that they are easier to train and have many fewer parameters than fully connected networks with the same number of hidden units. In this article we will discuss the architecture of a CNN and the back propagation algorithm to compute the gradient with respect to the parameters of the model in order to use gradient based optimization.

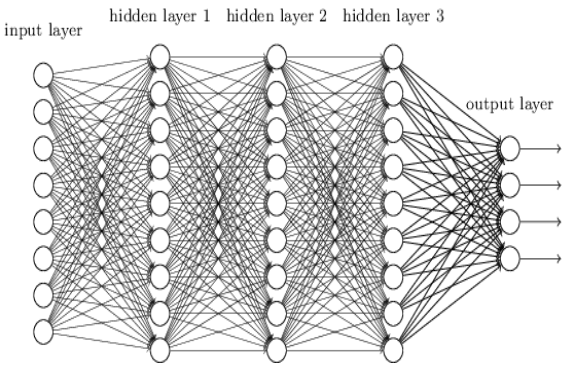

Fig 4. General Architecture of CNN

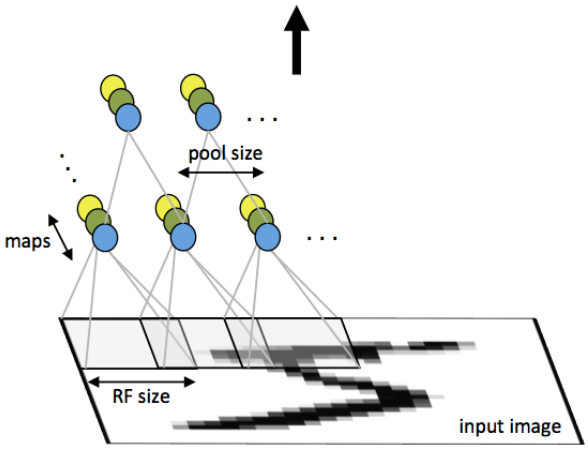

Fig 5. First layer of a convolutional neural network with pooling.