UDACITY - MACHINE LEARNING ENGINEER NANODEGREE

Capstone Proposal - Age Estimator[1]

Khalid A. Al-Abbad

26 November 2018

[1] Image produced using a collage from 1200 images from the UTKFace database

Domain Background

Computer vision is one of the primary application areas in artificial intelligence. With the advent of deep learning, and convolutional networks, computer vision has become a fast progressing part of machine learning. An important part of computer vision is face perception related technologies. Despite being an automatic thing for us as humans, a lot goes on in the human brain to effectively perceive faces and associated emotions, and this happens very early on in human development. A newborn (less than 3 days old) can recognize faces already[2]. Some brain damage may cause the brain to lose its ability to detect faces (face blindness, or Prosopagnosia[3]). This makes the problem an attractive one for us, as humans, to solve using AI and ML techniques.

Age determination given a face is an even more challenging problem. Identifying whether a given photo belongs to a child or an elderly person is a reasonably easy task for a human to carry out. However, identifying the age of a person within a few years ± is not as easy. There are many potential uses for age estimation such as forensics[4] and security[5].

A review of previous face estimation techniques shows this is a lively field with many possible approaches[6]. It has been approached using many areas of ML, including PCA[7], CNN[8], SVR[9], and there are many products that provide ready made age prediction[10]. A few years back, I saw an early prototype of a screen that attempts to predict your age and emotions as you pass by it, and it felt like it was doing the impossible. I chose this project particularly because it feels great to be able to do what I thought was impossible a few years back.

Problem Statement

The goal behind this project is to build a model that predicts the age of a person given a photo that includes the face of the person. This model will be used to build a basic application which - for entertainment purposes - allows a person to upload a photo, and predicts the age of the person.

In order to add some educational value, the application will also include a sampling of photos from the dataset, and ask the user to estimate the age of the given photo, and see if they can beat the estimation of the developed model.

[2] https://en.wikipedia.org/wiki/Face_perception

[3] https://www.nhs.uk/conditions/face-blindness/

[4] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4760148/

[5] https://www.um.edu.mt/library/oar/handle/123456789/8602

[6] http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=72B9E841852E2C33921B8F0004955069?doi=10.1.1.221.213&rep=rep1&type=pdf

[7] https://www.researchgate.net/publication/263547734_How_Old_Are_You_Age_Prediction_using_Eigen_Face

[8] https://arxiv.org/pdf/1709.01664.pdf

[9] https://core.ac.uk/download/pdf/21749390.pdf

[10] https://www.sighthound.com/products/sighthound-video, https://azure.microsoft.com/en-us/services/cognitive-services/face/

Datasets and Inputs

The primary dataset that will be used for this project is the UTKFaces[11] dataset. This dataset contains over 20,000 photos along with information about each photo, including: Age, Gender, and Race. This information is conveniently included as part of the file name.
Additionally, the dataset includes two versions of each photo, one is cropped to just the face and aligned (using Dlib[12]), and the other is unprocessed. The ground truth for the age is actually an estimate reviewed by a human reviewer. Only the age will be used to train the model. This dataset is provided for non-commercial research purposes only.

Other potential datasets that may be used (if necessary) to augment the previous dataset, and to ensure the generality of the model:

- IMBD-WIKI dataset[13], which includes over 500K+ images obtained from IMDB and wikipedia, and containing metadata about each element including date of birth, gender, and face location. The dataset is released for research purposes only.
- OUI-Adience Face Image Project dataset[14] which includes over 26,000 photos obtained from Flickr uploads (CC). The dataset includes age-group (8 classifications), gender, and subject (There are 2000+ subjects). The dataset is provided for research purposes only.

Solution Statement

The target prediction model is a regression model built using a convolutional neural network through transfer learning. The model will take a resized image as an input, and first pass it through one of the leading architectures[15] (such as ResNet50 or InceptionV3) pre-trained on imagenet using keras to extract bottleneck features. The bottleneck features will then be taken and passed into a DNN (with one or more dense layers). A final dense layer with a single unit that predicts the age. Since this is a regression problem, a regression loss function will be used[16].

The application that uses the model will be built as a web application with python flask as a backend. This will make it easy to directly interact with the model when predicting the age.

[11] https://susanqq.github.io/UTKFace/

[12] http://dlib.net/

[13] https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/

[14] https://talhassner.github.io/home/projects/Adience/Adience-data.html

[15] https://medium.com/comet-app/review-of-deep-learning-algorithms-for-image-classification-5fdbca4a05e2

[16] https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0

Benchmark Model

As mentioned previously, there are many available benchmark models for the attempted task. "Deep Convolutional Neural Network for Age Estimation based on VGG-Face Model[17]" has been selected as a benchmark model to describe in some detail.

This model is a classification model that uses transfer learning. It is based on the VGG-Face model[18], which is a CNN model specifically designed to extract facial features. This model has 11 layers, 8 of them are convolutional, and the last 3 are dense layers, each of the convolutional layers is followed by a ReLU activation, and max pooling, dropout and normalization layers are also included after convolutions. The team has replaced the last 3 dense layers with 4 dense layers, the first dense layer is 4096 units, followed by 2 layers of 5000 units, and finally an 8 unit layer that outputs a class related to the age group.

Images are scaled to 256 by 256 pixels, and as a form of augmentation, cropped into patches of 224 by 224 pixels (Same approach is followed for prediction, where 3 patches are extracted from the image from specified locations and the resulting probabilities are averaged to come up with the final class).

The team has opted not to change the weights for the pre-trained layers, and only modify the weights for the dense layers during the training.

The model was trained using the OUI-Adience dataset and obtained 59.9% overall accuracy, and 90.57% 1-off accuracy (where the team also considered classifications that were only one group off as accurate).

Evaluation Metrics

The most common evaluation metric used by the different studies and applications is the mean absolute error (MAE). The formula for the MAE is as follows[19]:

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

Table 1 from Age Synthesis and Estimation via Faces survey[20] shows different values form MAE for different studies. They range - in case of regression - from 3 to 10. The target of this project will be to get below 10 mean absolute error.

An additional metric that will be used is what I refer to as "soft accuracy." Rather than categorizing the age into specific age groups, which has the disadvantage of high error rate in cases at the edges of the categories (unless 1-off-accuracy is used). I chose this metric that simply considers the estimate accurate if it falls within X

[17] https://arxiv.org/pdf/1709.01664.pdf

[18] http://www.robots.ox.ac.uk/~vgg/software/vgg_face/

[19] https://en.wikipedia.org/wiki/Mean_absolute_error

[20] http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=72B9E841852E2C33921B8F0004955069?doi=10.1.1.221.213&rep=rep1&type=pdf page 1970.

years from the real value. I propose to use two soft accuracies, the first is within 5 years, and the second is within 10 years.

Project Design

The following workflow is planned to be used for this project:

1. Workspace preparation:

Training a deep neural network is a computing intensive task, and requires powerful hardware. For CNN training, the best performance is achieved using GPU. Udacity provides a workspace that can use GPU, however, it is limited to 50 hours of usage (part of it has been already used in previous projects). In order not to consume these resources during the analysis portion of this project, the following approach will be followed:

- Google colab[21] provides free GPU/TPU empowered notebooks for research purposes. As an advantage, these notebooks can be connected to Google Drive where all the data can be stored (up to 15GB). Google colab will be used in order to prepare the data, extract bottleneck features, try different versions of the model, and save any resulting data into Google Drive.
- Prior to submitting the project, I will attempt to move the data and final computation to Udacity workspace - if workspace limits allow it - in order to simplify the review process.

2. Data preparation and analysis:

The proposed UTKFaces dataset provides two variations of the data. One with cropped and aligned faces, and another with faces-in-the-wild. I will consider three options, to see which is the more feasible:

- Use the full images as-is during training and testing. Cropping them to the actual faces may not be necessary (especially that the dataset should have only 1 face per image) since a CNN should be able to handle it. One option would be to perform some image augmentation such as scaling and rotation to even make the model more robust.
- Use the aligned and cropped faces dataset, which is the smallest dataset. If this dataset is used, the application should crop the images uploaded before passing them to the model (and possibly rotate them, or require straight faces from the user).
- Use the full images, and crop them before training/testing, and subsequently crop images uploaded before passing them to the model. This may have an advantage over the previous option despite requiring more work, since it will ensure that the cropping/alignment process is consistent for training/testing.

Additionally, prior to proceeding further (regardless of which option to pick), and images should be resized to a standard size (whatever the default input for the selected leading architecture is). For instance, the default size for InceptionV3 is 299 by 299 pixels, and that for ResNet50 is 244 by 244 pixels[22].

[21] https://colab.research.google.com/

[22] https://keras.io/applications/

While the primary dataset has over 20,000 images, this number of images would be very difficult to process in our workspace, and will make the project review very complex. Even though more data would allow for a better model, training / testing data will be done on a subset of this dataset. As an initial estimate, 2000 training/validation samples, and 2000 testing samples may be used. Care will be taken to ensure that both samples have a similar distribution to the full dataset.

3. Extract Bottleneck Features

There are two approaches to transfer learning. One approach is to remove the top layers from the pre-existing model and add your own additional layers to the model itself. The other approach is to remove the top layers from the pre-existing model and use it to extract bottleneck features into an array, and then pass this array to a new model which contains the additional layers.

The advantage of the first option is that you can further train the initial layers (or any layers of your choice from the pre-existing model) using your training data. The disadvantage is that the process will be much more time consuming since you need to go through the pre-existing model every single time you train / test your data. In our case, we do not need to further train the convolutional layers, since our dataset will be relatively small, and the imagenet weights should be sufficient for our needs. A very large amount of data, and processing time, would be required otherwise.

There are many models that can be used as a basis for our estimator. Our benchmark opted to go for a pre-existing model that already captures faces. In the case of this project, I would like to go for a more general model, and use one of the leading architectures. The project will attempt choose among leading architectures such as ResNet50[23] and InceptionV3[24] with imagenet weights.

4. Model Architecture and Training

With the bottleneck features extracted. These features will be passed to another DNN. Several architectures will be tested for the DNN, and they will include the following layers:

- An initial Flatten or Max Average Pooling layer: One of these will be used in order to convert the convolutional structure received from the bottleneck features into a flat structure that can be passed to dense layers. The advantage of flattening layers is that they keep the maximum amount of information. The max average pooling layers, on the other hand, act like a lossy compression, which has the effect of speeding up the training/prediction and keeping the number of parameters in the network lower. A flatten layer will be the preferred option.

- Dense layers: These layers will be the main part of our deep neural network. There will be 1 final dense layer with 1 unit that would estimate the age. Feeding into that will be one or more dense layers. Each dense layer will be followed by ReLU activation.

[23] https://arxiv.org/pdf/1512.03385.pdf

[24] https://arxiv.org/pdf/1512.00567.pdf

- Batch normalization layers[25]: These layers protect against overfitting, speed up training, and balance the network. They do so by scaling the values of the previous layer.
- Dropout layers: Dropout layers allow the network to train more evenly by randomly disabling a percentage of the units in a given training run. This will again protect against overfitting and make the network more robust.

Additionally, since this is a regression problem, a suitable loss function will be selected for the model. Mean Absolute Error may be a good option. However, other options such as Mean Squared Error, and Quantile Loss, will be considered[26].

5. Application Development

A basic web application with a python flask backend will be developed. The application will contain the following components:

- Web Interface: The interface will be developed using a web framework such as angular. It will contain 2 views, one to estimate your own age, and the other to test your age estimation capabilities against that of the model.
- Python Flask Backend: This will provide an endpoint to predict age given an image, and another to pick a random image from a sample (which will be small for the purposes of this project) to present to the user to predict its age.

[25] https://towardsdatascience.com/batch-normalization-in-neural-networks-1ac91516821c

[26] https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0