Ace of Face

Jorge Hernandez | Justin Jeng | Preethi Raju | Graham Schweer
October 26, 2020

I. Problem Statement

The objective of this task is to predict keypoint positions on face images using a convolutional neural network (CNN). Our project will use the facial keypoints detection data from Kaggle.

II. Motivation

This can be used as a building block in several applications, such as:

- Tracking faces in images and video;
- Analyzing facial expressions;
- Detecting facial signs for medical diagnosis; and
- Biometrics / facial recognition.

Detecting the position of key points on face images is an important and challenging problem for multiple facial recognition tasks.

III. Inference Problem

For this project we will be predicting the location of facial keypoints on a set of images. Locations will be identified as x and y coordinates of pixel indices on each image. There are 7,049 images in the training set of data and 1,783 images in the test set of data, and there are up to 15 facial keypoints on each image which are identified by up to 30 features (the x and y coordinates). Not all 30 features will be predicted for each image. For images where certain facial positions are missing (i.e. someone turns their head), we won't predict the missing key points.

Fewer facial keypoints may be detected and predicted based on different factors of each image, including the profile in the picture, the lighting, photo quality, facial hair or other hair covering the face, et al.

Below is a categorized list of the 15 facial keypoints that will be predicted.

- Eyes (6): Left Eye Center, Right Eye Center, Left Eye Inner Corner, Left Eye Outer Corner, Right Eye Inner Corner, Right Eye Outer Corner
- Eyebrows (4): Left Eyebrow Inner, Left Eyebrow Outer, Right Eyebrow Inner, Right Eyebrow Outer
- Nose and Mouth (5): Nose Tip, Mouth Left Corner, Mouth Right Corner, Mouth Center Top, Mouth Center Bottom

Our inputs will be 96x96 pixels of images. Our outputs will be (x,y) coordinate pairs for each facial keypoint from the list above that can be predicted for an image. For the model, we will be using a

convolutional neural network.

IV. Exploratory Data Analysis

In our initial exploration of the training data, we noticed several features for which a majority of the observations are missing data. Each of the following eleven facial keypoints is missing x- or y-coordinate data for more than 67% of the training images: Left Eye Inner Corner, Left Eye Outer Corner, Right Eye Inner Corner, Right Eye Outer Corner, Left Eyebrow Inner End, Left Eyebrow Outer End, Right Eyebrow Inner End, Right Eyebrow Outer End, Mouth Left Corner, Mouth Right Corner, Mouth Center Top Lip and Mouth Bottom Lip. The other four facial keypoints -- Left Eye Center, Right Eye Center, Nose Tip and Mouth Center Bottom Lip -- are missing data for less than 0.5% of training images.

V. Approach

We detail our questions and approaches below to build a model with the **lowest RMSE**.

- Questions
    - How to handle images with missing facial points?
        - Prediction or label as missing
        - Feature engineering: Adding boolean features for "types" of images (e.g. left-side of face image, right-side of face image, etc.) based on patterns of missing feature data. This assumes we use all image examples provided in the training set.
    - What dimensions to keep features in? For features, does it make sense to flatten pixel images into a 1d- 9,216 array? Or keep them as 96x96 shape for the CNN keras input?
    - What dimensions to keep labels in? We are considering keeping them shaped as 1x30 arrays since the algorithms might be able to learn "distance" between facial keypoints.
    - Baseline ML regression models? We should be able to predict 1x30 vectors of x,y keypoint chords.
        - KNN regression, linear regression
- Feature Engineering / Improving accuracy (we will each take on at least one):
    - Color mapping
    - Rotation
    - Gaussian blur
    - Adding noise
    - Edge detection
    - Image grid / flat array (image shape)
- Tuning
    - We will decide the best parameters (epochs, learning rate, batch size, etc.).