Capstone Proposal

Ahmed Hamdy

- ## Domain Background

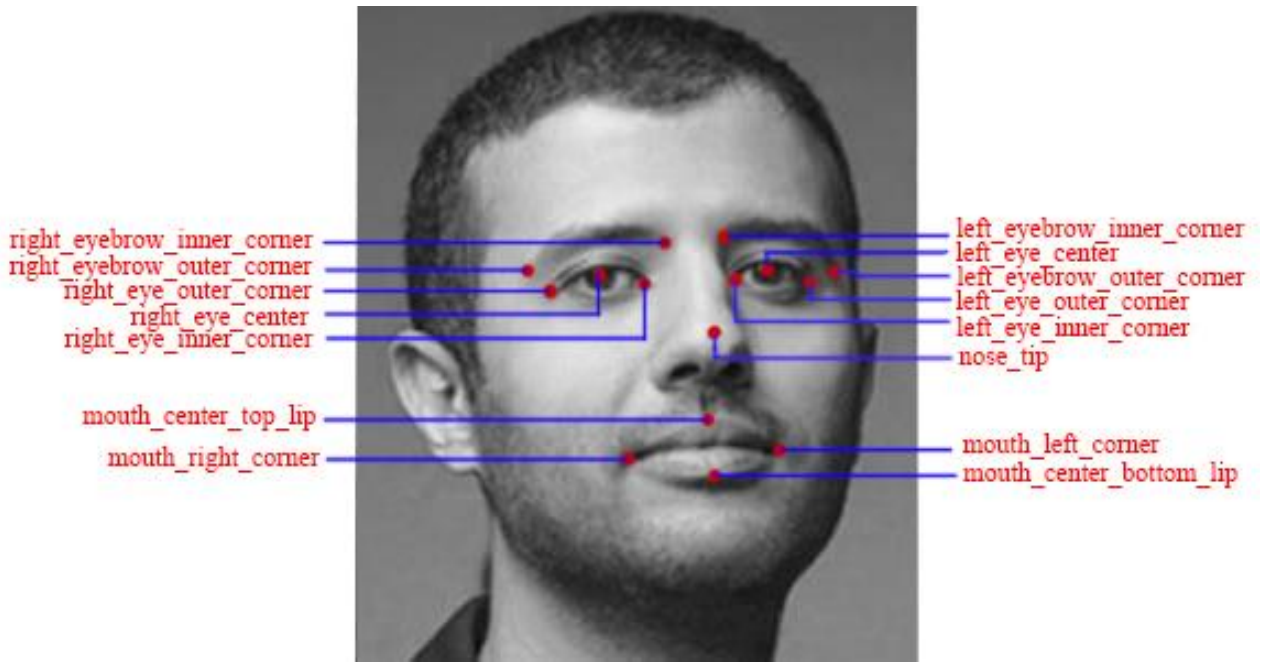
  Detecting facial key-points is a very challenging problem and has been studied extensively in recent years [1, 2, 3]. Facial features vary greatly from one individual to another, and even for a single individual, there is a large amount of variation due to 3D pose, size, position, viewing angle, and illumination conditions. Computer vision research has come a long way in addressing these difficulties, but there remain many opportunities for improvement.

  Detecting facial key-points can be used as a building block in several applications, such as: tracking faces in images and video, analyzing facial expressions, detecting dysmorphic, facial signs for medical diagnosis, biometrics / face recognition.

  My personal motivation for investigating this problem is my interest in human communication.

- ## Problem Statement

  We want to predict several features on the human face. These features are clearly represented In the picture below (picture is not from the dataset, and it is used only for illustration purpose)



- # Datasets and Inputs

  Each predicted keypoint is specified by an (x,y) real-valued pair in the space of pixel indices. There are 15 keypoints, which represent the following elements of the face: left_eye_center, right_eye_center, 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, nose_tip, mouth_left_corner, mouth_right_corner, mouth_center_top_lip, mouth_center_bottom_lip

  (Left and right here refers to the point of view of the subject.)

  In some examples, some of the target keypoint positions are missing (encoded as missing entries in the csv, i.e., with nothing between two commas).

  The input image is given in the last field of the data files and consists of a list of pixels (ordered by row), as integers in (0,255). The images are 96x96 pixels.

  Data files:

  training.csv: list of training 7049 images. Each row contains the (x,y) coordinates for 15 keypoints, and image data as row-ordered list of pixels.

  Dataset is taken from a Kaggle.com : https://www.kaggle.com/c/facial-keypoints-detection/data

- # Solution Statement

  Our dataset consists of images, and the best way to extract spatial information in images datasets is by using Convolution Neural Network.

  The solution will be a model of consecutive convolutional and max pooling layers such that the area of the image gradually decreases while its depth increases and carrying on only the important information that help in the prediction of the required features.

- # Benchmark Model

  I will compare my model against a model I have made which is a multilayer perceptron neural network which has 4 layers: input layer = 9216 neurons, two 512 neurons hidden layers, output layer = 30 (one neuron for each feature that the model will predict)

  This model gave an MSE = 27.85 which I predict to decrease significantly after using a CNN model.

  Benchmark model link : https://github.com/Ahmed-H-N/machine-learning/blob/master/ML%20nd/projects/capstone/facial_keypoints_detection/Capstone%20-%20Facial%20Keypoints%20Detection%20(benchmark%20model).ipynb

- # Evaluation Metrics

Root mean squared error is used as the evaluation metric. RMSE is very common and is a suitable general-purpose error metric. Compared to the Mean Absolute Error, RMSE punishes large errors:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

where $\hat{y}$ is the predicted value and y is the original value.

- # Project Design

- Starting by exploring the data, to have a sense of the range of values and any missing information.

- Then visualize some of the images and display the keypoints on them.

- Then any missing information will be handled and data will be prepared for training (data type and shape)

- Then the CNN model is defined, and training is carried on.

- Last, model is tested, and results are shown and a comparison is made against the benchmark model.

References

(1) Yi Sun, Xiaogang Wang, Xiaoou Tang. Deep Convolutional Network Cascade for Facial Point Detection. In Proc. CVPR 2013
(2) R. Herpers, M. Michaelis, K.-H. Lichtenauer, G. Sommer. Edge and keypoint detection in facial regions
(3) Stefano Berretti, Boulbaba Ben Amor, Mohamed Daoudi, Alberto del Bimbo. 3D facial expression recognition using SIFT descriptors of automatically detected keypoints