McGill University
ECSE 415 - Intro to Computer Vision
Final Project - Face Detection and Age Classification
Due: 11:59pm - Nov. 30th, 2017

Preface

In this project, you will be working in groups of 3-4 students to develop a software system for face detection and age classification. Specifically, the outline of this project is two-fold: (1) you will be exploring the popular Viola-Jones Adaboost detection algorithm [4] to help locate facial and eye images, (2) you will be implementing an age classification method to categorize facial images into four age groups: babies, young adults, middle-aged adults, and old adults [2]. Please submit your project solutions electronically via the myCourse assignment dropbox. You only need to submit one project per group. Your project report should be in a **pdf** format and your code should be in a **Jupyter Notebook** file. The complete assignment submission should be a zip file containing your report (.pdf) and Jupyter code (.ipynb). Attempt all parts of this project. The project is out of a total of **100 points**. The student is expected to write his/her own code. Assignments received up to 24 hours late will be penalized by 30%. Assignments received more than 24 hours late will not be marked.

Part 1 - Acquiring Dataset [15 points total]

Part I. Acquiring [5 points]

For this portion you will be creating your own dataset. You will need to collect 250 grayscale mugshot facial images. You must incorporate various age groups in your dataset. When creating your dataset you should try to adhere to the following standards: (1) the facial image should be front-facing, (2) the face should be centered in the image and take up about one-half of the image width, (3) the subject's eye should be positioned in the top half of the image (but close to the center). Each image size should be of dimension 150×200 (width $\times$ height). Figure 1 demonstrates a subset of collected facial images. You may find the *CelebA* dataset useful when creating your dataset[3]. In your report you should display a random sampled subset of these images.

Part II. Labeling [5 points]

Once you have a diverse dataset you will need to classify, subjectively, each image into twelve age intervals, as shown in Table 1. Each group member should annotate all images in the dataset and the final weighted average age of these subjective judgments are then computed. For example, figure 2 below may be classified between $[30 - 39]$ by 3 people and $[20 - 29]$ by 1 person. Thus, the weighted average age is $(35 \times 3 + 25 \times 1)/4 = 32.5$ Using this system, images with weighted averages below 3 are categorized as babies; those between 3 and 39 are categorized as young-adults; those between 40 and 59 are categorized as middle-aged adults; and those over 60 are categorized as old adults. For example, the person in figure 2 may be classified as a young adult. In your report you should display, randomly, 5 images from each age label along with their respective weighted average.

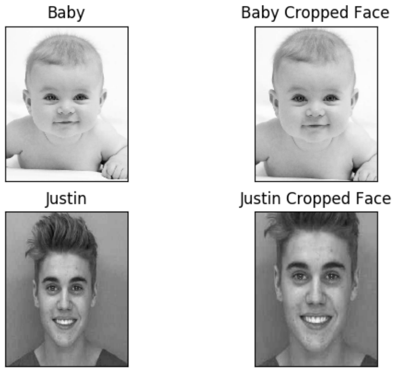

Figure 1 - Cropped Frontal Facial Images

Figure 2 - Sample Facial Image

Part III. Partitioning [5 points]

Next, you will need to partition your dataset into a training and testing set. Half of the images in each group are used as training data and the remaining half are used as test images.

Table 1: Age Intervals and Age Groups

Age Group	Age Intervals
Baby	0 - 2
Young Adults	3 - 12
	13 - 19
	20 - 29
	30 - 39
Middle-Aged Adults	40 - 49
	50 - 59
Old Adults	60 -69
	70 - 79
	80 - 89
	90 - 99
	100+

Part 2 - Age Classification [65 points total]

The overall age classification system is composed of 3 stages: location, feature extraction, and classification. In the location stage the framework is required to first find common facial landmarks (e.g. eyes, nose, etc.) In the second stage, both geometric and wrinkle features are extracted. In the last stage you will be testing various classification algorithms that you have seen in class, namely Support Vector Machines, Random Forests, or Neural Networks. You will be testing and comparing at least two of these classifiers.

Stage 1 - Location [20 points]

Assuming that the width and height of a given image are W and H respectively, we can assume that the facial region is about $W/2$. It is worth noting that all ratios may change depending on how you cropped your images in Part 1. The face vertical line can be easily found due to human face symmetry. Since most of the eyes are located in the middle height, $H/2$, of the image, we can define the *eyes' searching region* as per figure 3. The left and right boundaries are located from the vertical center line of the face by extending $W/4$ horizontally in both directions. The top and bottom boundaries are positioned at $H/3$ and $3H/5$ respectively. Once the eyes' region of interest in found, the eyes' center line must be located. The procedure for finding this center line is as follows: (1) The Sobel edge operator is applied to the original image to find the edges. You will have to convert this edge image into a binary image. A threshold of 100 was used in testing but will vary depending on your dataset. (2) Perform a horizontal projection in the eyes' searching area. A horizontal projection is simply a histogram in the vertical direction. Due to the fact that eyes have strong intensity variations, the major peak in the histogram represents the position of the eyes. Figure 4 demonstrates this property.

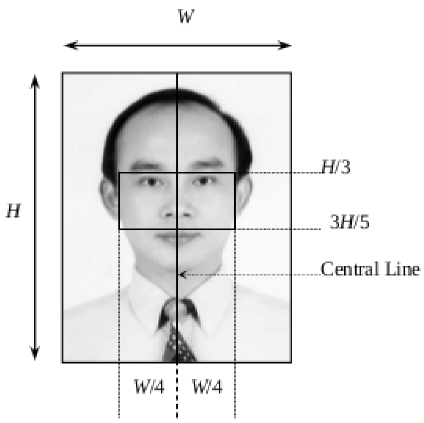

Figure 3 - Eyes' Searching Region [2]

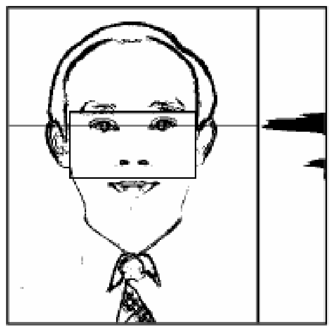

Figure 4 - Position of Eyes [2]

Next, the areas of the eyes must be located. We will be testing two methods to find eye areas. The first method requires finding the nearest big enough regions (an area sized of 30 was used in experimentation) to the left and right from the vertical center line via region labeling. You can use opencv's connectedComponets() function for quick region labeling. The second method utilizes Viola-Jones adaboost classifier to detect eye locations. You should use opencv's pre-built model and functions for this part. You should display to results of both methods and choose the method that yields the best results.

Next, the nose and mouth in a given image can be located in a similar way to finding the eyes' center line. The distance between the outer margins of the eyes, as shown in figure 5, is defined as D_{eyes}. The distance from eyes to the mouth is around $2D_{eyes}/3$ and the left and right boundaries of the mouth are located about the centers of their corresponding eyes. The *nose-mouth's searching region* is illustrated in figure 5. Note, the top boundary is located at the lower bottom boundary of the two eyes. Similar to finding the eyes' area in method one, the horizontal projection is first performed on the nose-mouth's searching region (of the Sobel image). The projection is then smoothed with an average filter, a kernel width of 5 was used in experimentations. The position of the nose and mouth are located at the two histogram peaks. Using simple calculus you can find the peaks by locating the zero-crossings after applying a first-order derivative filter. This process will result in the location of the two highest peaks. The peak closest to the eyes is the position of the nose and the other is that of the mouth (see figure 5). Once the mouth is located, similar to the eyes, region labeling can be applied to find the area of the mouth. Figure 6 demonstrates the result of the location stage.

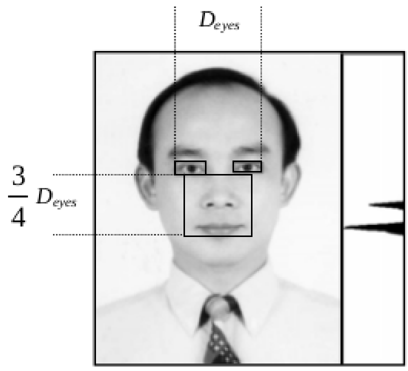

Figure 5 - Nose-Mouth's Region [2]

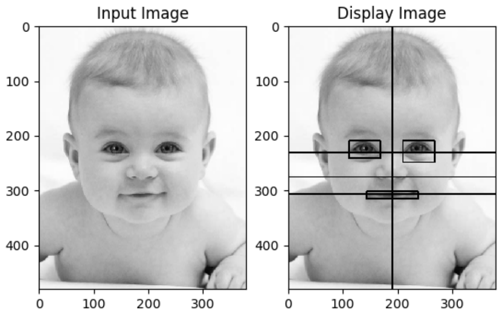

Figure 6 - Locations

In your report you should display the locations of each facial landmark (i.e. similar to figure 6) for 5 randomly sampled images in each age category.

Stage 2 - Feature Extraction [25 points]

For this stage, we are dividing the features into two categories: wrinkle features and geometric features.

Stage 2.1 - Wrinkle Features

To compute wrinkle features we will be looking at the following recognizable wrinkle structures: (1) horizontal furrows on the forehead, (2) crows' feet on eye corners, and (3) obvious cheekbones and crescent-shaped pouches in the cheeks along with deep lines between the checks and the upper lips. Thus, as shown in figure 7, the five areas of interest are: the forehead, two eye corners, and two cheeks. You should use the results from the previous location phase to help extract these new interest regions.

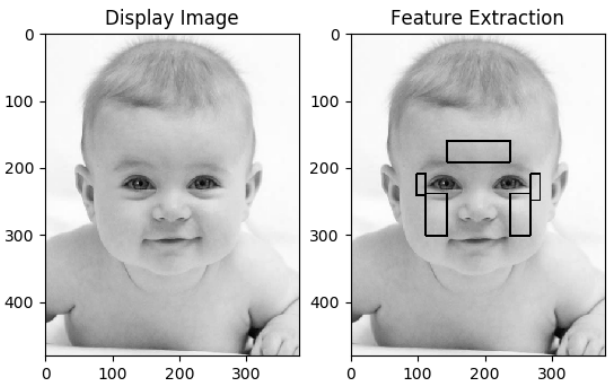

Figure 7 - Regions for Wrinkle Extraction

To judge the degree of wrinkles, the Sobel edge magnitude is used (wrinkles have an obvious change in intensity). If a given pixel belongs to a wrinkle then its Sobel edge magnitude is larger than some threshold (40 used in experimentation). In order to quantify the degree of wrinkles three functions are used:

Wrinkle Density: The density of wrinkles in area A is defined as:

$$D_{I,A} = \frac{|W_A|}{|P_A|}$$

where W_A represents the set of all wrinkle pixels in area A and P_A is the set of all pixels in area A.

Wrinkle Depth: The depth of wrinkles in area A is defined as:

$$D_{2,A} = \frac{1}{\alpha|W_A|} \sum_{(x,y)\in W_A} M(f(x,y))$$

where $M(f(x,y))$ represents the Sobel edge magnitude in W_A. α helps control the size of depth and is set to 255.

Average Skin Variance: The average skin variance in an area A is defined as:

$$V_A = \frac{1}{\alpha|P_A|} \sum_{(x,y)\in P_A} M(f(x,y))$$

The less the average skin variation in area A, the smaller the value of V_A and thus the smoother the skin area.

Stage 2.2 - Geometric Features

In addition to using the wrinkle features listed above, geometric features will also be extracted from input images. The main issue with wrinkle features is the unreliability when it comes to newborn babies, since newborn babies often have a lot of wrinkles on their faces. To help solve this problem geometric features are used to differentiate between head bones and ratio between primary features in infant-hood compared to other life periods. The two geometric features used are as follows:

First Geometric Feature: The first geometric feature is the ratio defined as:

$$R_{em} = \frac{D_e m}{D_e e}$$

where D_{em} is the distance between the eyes and the mouth, and D_{ee} is the distance between two eyes' centers. For babies, R_{em} is generally small.

Second Geometric Feature: The second geometric feature is the ratio defined as:

$$R_{enm} = \frac{D_e n}{D_n m}$$

where D_{en} is the distance between the eyes and the nose, and D_{nm} is the distance between the nose and the mouth. For babies, the value of R_{enm} is close to or less than 1.

In your report, you should calculate and discuss the average and variance for each feature in each age group.

Additional Features [5 points BONUS]

In addition to implementing the above features, you are allowed to create and test your own features and bonus marks will be given if the final testing accuracies improves.

Stage 3 - Age Classification [20 points (10 each)]

For this stage you will be testing two different classification algorithm. You should pick two algorithms that were covered in class, namely Support Vector Machines, Random Forests, Neural Networks, etc. We will be using python's popular machine learning library *scikit-learn* [1]. If you installed Anaconda as per the tutorial slides then you will have scikit-learn already installed. Otherwise you will have to install the library using *'pip install -U scikit-learn'* in your terminal window.
To use these machine-learning libraries you will need to organize your data into a matrix containing all your training/testing instances and their respective features. Thus, your training data should be of the dimensions $X_{train} = [125, 15]$. Since we are using half the dataset for training our first dimension is 125. You second dimension is 15 because this is the number of features we find for each number. Our training labels will have the dimensions $[125, 1]$. The second dimension is 1 because it simply represents which class the training example belongs to. The testing dataset will be represented similarly. Once you have your data represented correctly, you will be running and fine-tuning your algorithms on the training dataset. Once you receive an optimal score, you should then run your best configuration on the testing dataset to receive your final accuracy. In your report you should discuss accuracies (training and testing) for each algorithm. The accuracies (training and testing) for each bin should be computed and represented in a bar graph. You should also try various hyper-parameters and discuss their results.

Part 3 - Age Tagging [20 points total]

For this part of the assignment you will be using your best performing age classifier along with Viola-Jones adaboost face detector to locate faces and classify their respective ages in a group photo. You should use opencv's pre-build model and function for the face detection portion. Once faces are detected you will need to display a bounding box around each face. After detecting and locating all faces in the input image you can then treat each detected face as a separate image and use your age classifier to find the age label associated with each face. You should print the age category beside to each head on the image. If the correct label was assigned, you should print the age label as green, if the label was incorrect, you should print it as red.

Obtain five different group photos (each image should have the faces facing frontwards and a variation of ages) and perform age tagging. You can try to use a photo of all group members to see how old each group member is! Display all final results in the report.

Notes on Report:

In addition to the previous mentioned requirements, you should also include the following:

- Project title.
- List of group members, including their full name, email, and student number.
- Introduction: Briefly describe the problem and summarize the approached used.

- Problem Representation: Describe and support with evidence how well each feature worked, in addition if you decided to include additional features describe the methodology behind them.
- Algorithm Selection and Implementation: Include any decisions about training your data, describe you hyper-parameters and how you went about choosing values.
- Testing: Detailed analysis of results.
- Discussion: Discuss pros/cons of your approach.
- References (optional): Use any reference format you wish.

References

[1] Lars Buitinck et al. "API design for machine learning software: experiences from the scikit-learn project". In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. 2013, pp. 108–122.

[2] Wen-Bing Horng, Cheng-Ping Lee, and Chun-Wen Chen. "Classification of Age Groups Based on Facial Features". In: 2004.

[3] Ziwei Liu et al. "Deep Learning Face Attributes in the Wild". In: *Proceedings of International Conference on Computer Vision (ICCV)*. 2015.

[4] Paul Viola and Michael Jones. "Rapid object detection using a boosted cascade of simple features". In: 1 (Jan. 2001).