MORPH-II: Feature Vector Documentation

Troy P. Kling

NSF-REU Site at UNC Wilmington, Summer 2017

1 MORPH-II Subsets

Four different subsets of the MORPH-II database were selected for a wide range of purposes, including age estimate, gender and race classification, and facial recognition.

- The "Full" data set contains all 55,134 mugshots [1].
- The "Partial" data set contains 1,000 mugshots randomly selected from the full data set.
- The "Partial (Even)" data set contains 1,000 mugshots selected from the full data set according to very strict rules and is intended mainly for age estimation tasks. The subjects range in age from 21 to 45, with exactly 40 subjects in each age category (thus the term "even" in the name of the data set). Of these 40 subjects in each age group, exactly 30 are male and 10 are female, giving rise to a 3:1 gender ratio. Additionally, half of the males in each age group are black, and the same goes for the females, so there is a precise 1:1 ratio of black to white individuals. No subject is represented more than once in this data set, so it should not be used for face recognition tasks.
- The "Recognition" data set contains 1,660 mugshots selected from the full data set according to certain rules and is intended to be used for facial recognition tasks. There are 166 subjects present in the data set – 83 males and 83 females – each of whom has exactly 10 images, usually taken over the span of multiple years. No restrictions on age or race were placed on this data set.

2 Image Preprocessing

Due to the variety of poses and lighting conditions present in the MORPH-II database, certain preprocessing techniques were used to standardize the images and make them more suitable for machine learning tasks.

2.1 Original Images

The "Original" data set of 55,134 images exhibits a wide range of differences that impede the performance of machine learning algorithms. The most obvious dissimilarity is that not all images in the original data set have the same resolution. Rather, they come in two varieties: 200x240 pixels and 400x480 pixels (measured width-by-height).

The next challenge is that the subjects' distance from the camera varies wildly, with some pictures being taken up-close (thereby causing the subject's face to take up the entire frame) and other pictures being taken from far away (thereby causing the subject's face to be much smaller relative to the picture as a whole and surrounded by a blank, uninformative background). This means that a subject's face could occupy a region of the image much smaller than 200x240 pixels.

In addition to the subjects' faces being of varying size relative to the image as a whole, many are also tilted to the side by up to 20 or 30 degrees in either direction. This causes a majority of images to contain faces that are not level, which further impedes the performance of machine learning algorithms. Some subjects also have their head tilted toward or away from the camera, which introduces another layer of complexity (and one that is far more difficult to correct for).

Finally, the lighting conditions in the images are not uniform. Some images are significantly darker than others, and some are so bright that important details of the face are lost. Ideally, one would want the distribution of light levels to be roughly equal across all images to prevent changes in illumination from affecting the accuracy of classification and regression methods.

These four problems (image resolution, distance from the camera, head tilt, and illumination) are the major issues that are corrected in the following data sets.

2.2 Cropped Images

Cropping the images corrects three of the four problems (all except illumination). The "Cropped" data set was created through the following steps: (1) Convert the images to greyscale, (2) Locate the eyes of the subject in the photograph, (3) Calculate the angle between the eyes and rotate the image until level, (4) Scale the image so that the inter-ocular distance is 30 pixels, and (5) Crop the image to 60x70 pixels, ensuring that the eyes are approximately 15 pixels from the side of the image and 25 pixels from the top of the image.

This process ensures that the areas of the face that are most important to age estimation, gender/race classification, and facial recognition (i.e. eyes, nose, and mouth) are located in roughly the same location across all images.

2.3 Histogram-Equalized Images

In addition to cropping the images to handle positional variation, histogram equalization was performed to account for changes in illumination. This not only increases the global contrast across an image, but also brings all 55,134 images onto a common illumination scale. The Python package scikit-image has a method called "equalize_hist" which was used to achieve this goal, giving us the "Equalized" dataset – a collection of cropped black-and-white images where most major sources of noise have been removed.

3 Feature Extraction

Depending on which machine learning task was performed, different features were extracted from the face in order to preserve the most relevant information. For our purposes, we experimented with four different feature extraction techniques: Local Binary Patterns (LBPs), Histograms of Oriented Gradients (HOGs), Gabor Wavelets (GWs), and Bio-Inspired Features (BIFs).

3.1 Local Binary Patterns

Local Binary Patterns (LBPs) were originally developed for the purpose of texture classification, but have since been shown to be useful for various other classification and regression tasks [2]. The LBP feature vector for a single image is created according to the following procedure:

1. Divide the image into blocks of a given size.
2. For each pixel in the block, consider a window of a given radius around that pixel.
3. For each pixel in the window, determine whether it is lighter or darker than the center pixel, and assign a value of 0 or 1 to that pixel.
4. Concatenate all 1s and 0s in the window into a single binary string representing the center pixel.
5. Compute the histogram of all binary strings across the current block.
6. Repeat this process for all blocks of the image, and concatenate all the histograms together to form a single feature vector.

There are various ways to optimize this process and reduce the size of the feature vectors without losing too much information. One possibility is to (a) throw away all binary strings that are sufficiently noisy; another is to (b) correct for rotation invariance. For our purposes, we used Python's scikit-image package with the option "nri_uniform", which performs (a), but not (b), since rotation invariance would ultimately degrade our results.

The two main tuning parameters for LBPs are block size and window radius. We experimented with a variety of different parameters – block sizes 10, 12, 14, 16, 18, and 20, and window radii 1, 2, and 3 (with 8, 12, and 16 sample points respectively).

3.2 Histograms of Oriented Gradients

Histograms of Oriented Gradients (HOGs) were originally developed for the purpose of object detection [3], and we thought it might be possible to extend HOGs to work in other domains as well. The HOG feature vector for a single image is created according to the following procedure:

1. Calculate the horizontal gradient, g_x, and vertical gradient, g_y, of the image using appropriate edge masks.

2. Calculate the overall magnitude, g, and direction, θ, of the gradient using the formulas

$$g = \sqrt{g_x^2 + g_y^2},$$
$$\theta = \tan^{-1}\left(\frac{g_y}{g_x}\right).$$

3. Divide the image into blocks of a given size.

4. Calculate the histogram across each block.

5. Concatenate all the histograms together to form a single feature vector.

The two main tuning parameters for HOGs are block size and number of orientations. We experimented with a variety of different parameters – block sizes 4, 6, 8, 10, 12, and 14, and number of orientations 4, 6, and 8.

3.3 Gabor Wavelets

Gabor Wavelets (GWs) are exceptionally large feature vectors built from a "bank" of filters [4]. Each filter takes the form of a sinusoidal wave modulated by a Gaussian kernel, i.e.

$$\exp\left(-\frac{X^2 + \gamma^2 Y^2}{2\sigma^2}\right)\cos\left(\frac{2\pi X}{\lambda}\right),$$

where $X = x\cos\theta + y\sin\theta$ and $Y = -x\sin\theta + y\cos\theta$. The GW feature vector for a single image is created according to the following procedure:

1. Given a value for γ (ranging from 0 to 1), create a bank of 32 filters, one for each combination of rotation angle (θ, ranging from 0 to π) and scale (an odd number ranging from 15 to 29).

2. Using a sliding window, convolve the image with each filter from the filter bank.

3. Vectorize the resulting images, and concatenate them together to form a single feature vector.

It should be noted that the parameters σ and λ can be set according to the scale, but the parameter γ (which represents the spatial aspect ratio) must be tweaked by hand to find the appropriate value. This leaves us with two main parameters to tune: block size and γ. We experimented with a variety of different parameters – we only used block sizes 15 to 29, because we found using the full range of 7 to 37 to be redundant, and we tried out γ values 0.1, 0.2, ..., 1.0.

3.4 Bio-Inspired Features

Bio-Inspired Features (BIFs) are an extension of Gabor Wavelets that more closely mimic primate visual cortex processing [5]. The BIF feature vector for a single image is created according to the following procedure:

1. Create the S_1 (simple) layer by generating a full bank of Gabor-filtered images as described above.

2. Create the C_1 (complex) layer by pooling over the same orientation and scale band from the S_1 layer. Pooling is done over a granular sliding window whose size is determined by the size of the corresponding Gabor filter. Either the maximum pixel value over the window can be used or the standard deviation of all pixels in the window, leading to two possible pooling operations: MAX and STD.

3. The resulting bank of images are vectorized and concatenated together to form a single feature vector.

More advanced implementations of BIF use an additional two layers called S_2 and C_2 [6], but we use the more basic implementation. It should be noted that pooling significantly reduces the size of the feature vector, because (1) the sliding window is half-overlapping and (2) two images from similar scales in the S_1 layer are combined into one image in the C_1 layer.

The three main tuning parameters for BIFs are block size and γ (for the same reason these has to be tuned for GWs) and pooling operation. We experimented with a variety of different parameters – block sizes 7 to 37 and 15 to 29, γ values 0.1, 0.2, ..., 1.0, and the pooling operations previously mentioned, MAX and STD.

4 Conclusion

In conclusion, we have four kinds of subsetting (full, partial, partial (even), and recognition), three levels of pre-processing (original, cropped, and equalized), and four types of feature extraction (Local Binary Patterns, Histograms of Oriented Gradients, Gabor Wavelets, and Bio-Inspired features). Then within each feature extraction method, several parameters were altered to produce a rather large amount of data. Altogether, this leaves us with hundreds of different data sets, each one individually tailored to a specific task.

When navigating the directories containing the feature vectors, one should note that the naming convention used for the .csv files is:

"MORPHII_[*LBP/HOG/Gabor/BIF*]_[*parameter-list*]_[*full/partial/partial_even/recog*].csv"

Feature vectors were only generated for images that had been both cropped *and* histogram equalized, so this is not mentioned in the .csv names.

Of greatest utility are likely the cropped and histogram-equalized partial (even) data sets and facial recognition data sets. The most appropriate feature extraction method depends on the task at hand, but Histograms of Oriented Gradients are likely to underperform compared to the other three methods.

5 Acknowledgements

This material is based in part upon work supported by the National Science Foundation under Grant Numbers (NSF DMS-1659288). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Karl Ricanek Jr and Tamirat Tesafaye, "MORPH: A Longitudinal Image Database of Normal Adult Age-Progression," IEEE 7th International Conference on Automatic Face and Gesture Recognition, Southampton, UK, April 2006, pp 341-345.

[2] Ojala, Timo, Matti Pietikainen, and Topi Maenpaa. "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns." IEEE Transactions on pattern analysis and machine intelligence 24.7 (2002): 971-987.

[3] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 1. IEEE, 2005.

[4] Fogel, Itzhak, and Dov Sagi. "Gabor filters as texture discriminator." Biological cybernetics 61.2 (1989): 103-113.

[5] Guo, Guodong, et al. "Human age estimation using bio-inspired features." Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009.

[6] Serre, Thomas, Lior Wolf, and Tomaso Poggio. "Object recognition with features inspired by visual cortex." Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 2. Ieee, 2005.