

Face Detection using LBP features

Jo Chang-yeon
CS 229 Final Project Report

December 12, 2008

I. INTRODUCTION

FACE DETECTION is a fundamental task for applications such as face tracking, red-eye removal, face recognition and face expression recognition[1]. To build flexible systems which can be executed on mobile products, like handheld PCs and mobile phones, efficient and robust face detection algorithms are required.

Most of existing face detection algorithms consider a face detection as binary (two-class) classification problem. Even though it looks a simple classification problem, it is very complex to build a good face classifier. Therefore, learning-based approaches, such as neural network-based methods or support vector machine (SVM) methods, have been proposed to find a good classifiers [2][3][4][5]. Most of proposed algorithms use pixel values as features. However, they are very sensitive to illumination conditions and noises [6].

Papageorgiou et al. [7] used new feature, it is called Haar-like features. These features encode differences in average intensities between two rectangular regions, and they are able to extract texture without depending on absolute intensities.

Recently, Viola and Jones proposed an efficient system for evaluating these features which is called an integral image [8]. And, they also introduced an efficient scheme for constructing a strong classifier by cascading a small number of distinctive features using Adaboost. Its result is more robustness and computationally efficient.

Base on Viola and Jones' work, many improvements or extensions have been proposed. Mainly, there are two approaches to enhance their scheme. The first approach is an enhancement of the boosting algorithms. Boosting [9] is one of the most important recent developments in classification methodology and, therefore, many variant of AdaBoost such as Real AdaBoost, LogitBoost, Gentle Adaboost, KLBoosting, etc[10], have been proposed. The second approach is an enhancement of used features. Base on original propose of Haar-like features, (a), Viola and Jones extend feature set as shown in Figure 1.

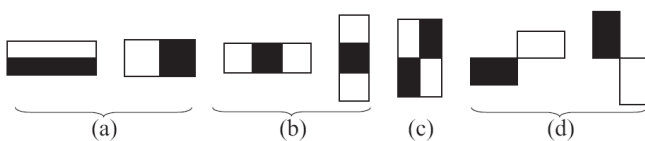


Figure 1. Example of Harr-like feature sets

(b), (c) and (d) in different size are used in [8][10] to extract features. And, Lienhart et al. [11] introduced an efficient scheme for calculating 45° rotated features. And, Mita and Kaneko introduced a new scheme which makes Haar-like features be more discriminative [6]. Though Haar-like feature provides good performance in extracting textures and cascading architecture and integral image representation make it computationally efficient, it is still not feasible on mobile products.

T. Ojala et al. [12] a new rotation invariant and computationally lighter feature sets. It should be noted that the basic LBP features have performed very well in various applications, including texture classification and segmentation, image retrieval and surface inspection [13].

II. LOCAL BINARY PATTERN

Local Binary Pattern (LBP) features have performed very well in various applications, including texture classification and segmentation, image retrieval and surface inspection.

The original LBP operator labels the pixels of an image by thresholding the 3-by-3 neighborhood of each pixel with the center pixel value and considering the result as a binary number. Figure 2 shows an example of LBP calculation.

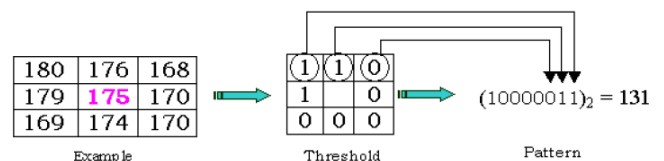


Figure 2. Example of LBP calculation

The 256-bin histogram of the labels computed over an image can be used as a texture descriptor. Each bin of histogram (LBP code) can be regarded as a micro-texton. Local primitives which are codified by these bins include different types of curved edges, spots, flat areas, etc. Figure 3 shows some examples

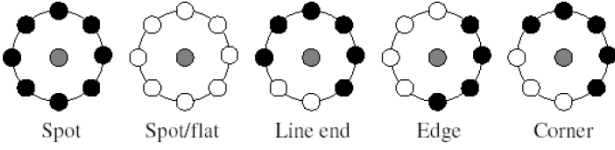


Figure 3. Examples of texture primitives

The LBP operator has been extended to consider different neighbor sizes. For example, the operator $LBP_{4,1}$ uses 4 neighbors while $LBP_{16,2}$ considers the 16 neighbors on a circle of radius 2. In general, the operator $LBP_{P,R}$ refers to a neighborhood size of P equally spaced pixels on a circle of radius R that form a circularly symmetric neighbor set. $LBP_{P,R}$ produces 2^P different output values, corresponding to the 2^P different binary patterns that can be formed by the P pixels in the neighbor set. It has been shown that certain bins contain more information than others. Therefore, it is possible to use only a subset of the 2^P LBPs to describe the textured images. Ojala et al. defined these fundamental patterns as those with a small number of bitwise transitions from 0 to 1 and vice versa. For example, 00000000 and 11111111 contain 0 transition while 00000110 and 01111110 contain 2 transitions and so on. Accumulating the patterns which have more than 2 transitions into a single bin yields an LBP descriptor.

The most important properties of LBP features are their tolerance against monotonic illumination changes and their computational simplicity.

A. LBP based Facial Representation

Each face image can be considered as a composition of micro-patterns which can be effectively detected by the LBP operator. Ahonen et al. introduced a LBP based face representation for face recognition. To consider the shape information of faces, they divided face images into M small non-overlapping regions R_0, R_1, \dots, R_M (as shown in Figure 4). The LBP histograms extracted from each sub-region are then concatenated into a single, spatially enhanced feature histogram defined as:

$$H_{i,j} = \sum_{x,y} I(f_l(x,y) = i)I((x,y) \in R_j)$$

where $i = 0, \dots, L-1, j = 0, \dots, M-1$. The extracted feature histogram describes the local texture and global shape of face images.

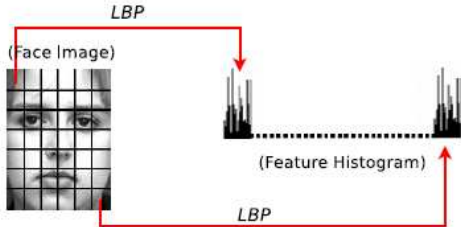


Figure 4. LBP based facial representation

III. LEARNING CLASSIFICATION FUNCTIONS

In this system, a variant of AdaBoost, Gentle AdaBoost is

used to select the features and to train the classifier. The formal guarantees provided by the AdaBoost learning procedure are quite strong. It has been proved in [15] that the training error of the strong classifier approaches zero exponentially in the number of rounds. Gentle AdaBoost takes a newton steps for optimization.

Gentle AdaBoost

1. Start with weights $w_i = 1/N, i = 1, 2, \dots, N, F(x) = 0$.
2. Repeat for $m = 1, 2, \dots, M$:
 - (a) Estimate $f_m(x)$ by weighted a fit of y to x .
 - (b) Update $F(x) \leftarrow F(x) + f_m(x)$
 - (c) Update $w_i \leftarrow w_i e^{-y_i f_m(x_i)}$ and renormalize.
3. Output the classifier $\text{sign}[F(x)] = \text{sign}[\sum_{m=1}^M f_m(x)]$

Figure 5. Gentle AdaBoost

The weak classifier is designed to select the single LBP histogram bin which best separates the positive and negative examples. Similar to [8], a weak classifier $h_j(x)$ consists of a feature f_j which corresponds to each LBP histogram bin, a threshold θ_j and a parity p_j indicating the direction of the inequality sign:

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) \leq p_j \theta_j \\ 0 & \text{otherwise} \end{cases}$$

Found weak classifiers are used to compose a strong classifier.

IV. THE ATTENTIONAL CASCADE

A cascade of classifiers is used, which achieves increased detection performance while obviously reducing computation amount. Simpler classifiers are used to reject the majority of sub-windows before more complex classifiers are used to achieve low false alarm rates.

Stages in the cascade are constructed by training classifiers using Gentle AdaBoost. A positive result from earlier strong classifier triggers next strong classifier which has also been adjusted to achieve higher detection rate than previous one. A negative result is immediately rejected at any stage of cascade structure.

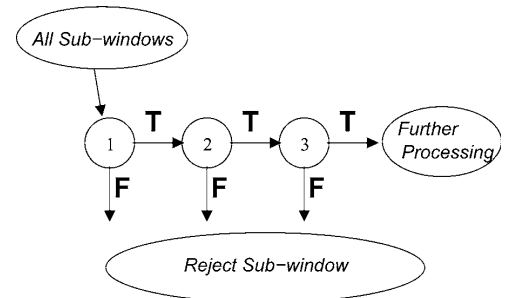


Figure 6. Schematic depiction of the detection cascade.

Because an overwhelming majority of input sub-window is negative, this method can significantly reduce the number of sub-windows which is processed with more complex classifier.

V. TRAINING DATASETS

Because I am building an appearance-based detection scheme, large training sets of images are needed in order to capture the variability of facial appearances. Some training examples are shown in Figure 7.



Figure 7. Example face images from the training set with rotation.

As shown in above figure, my training set also includes rotated face training examples to enable to detect rotated faces. Because rotated face in every 90° can be detected by rotating LBP operator, only $\pm 18^\circ$, 12° and 6° rotated face examples are added to training set.

With above training set, face detection works well; it can detect faces in images with low false alarm rate. But, it can not detect faces in low light condition and dark skin faces. To solve this problem, there are two approaches; one is image preprocessing and another is enhancing training set. Lizuo Jin and Shin'ich Satoh introduced a good method to estimate an illumination condition of image and enhance its quality [14]. But their method is computationally, it is not feasible on mobile product.

Therefore, to enable the system to detect faces in low light conditions, faces in various illuminations and dark skin faces are also added to my training set.



Figure 8. Example face images from the training set with various illumination conditions.

I collected 57,134 face images and used it as a positive training set.

To collect nonface patterns, I used the "bootstrap" strategy in five iterations [2]. First, my system extracts 200 patterns per an image from a set of false-alarm-causing image set which do not contain faces. Because most of false alarms are come from trees, characters, handwritings and fabrics, I used these kinds of images as a false-alarm-causing image set. Some examples are shown in Figure 9.

Then, at the end of each training iterations, I run the face detector and collected all those nonface patterns that were wrongly classified as faces and used them for training. And, extract negative training examples on false-alarm-causing image set again. To get more efficient negative examples, I used classifiers which were found in previous iteration and chose negative examples which were mis-classified as a face.

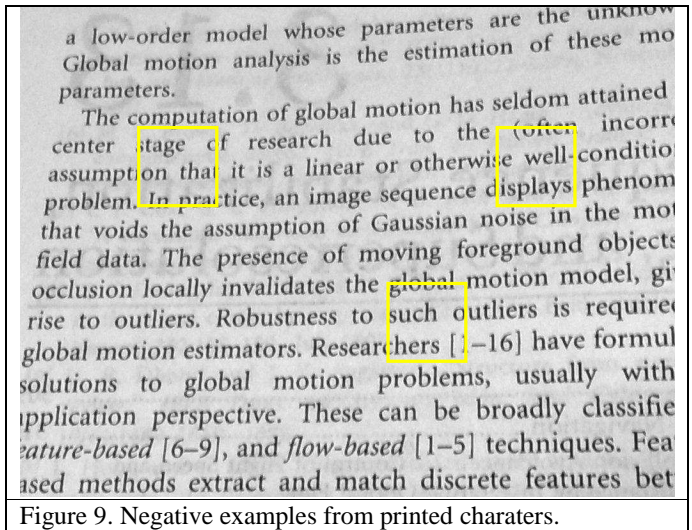


Figure 9. Negative examples from printed charaters.

By this method, I obtained 298,000 nonface patterns as negative training examples.

VI. EXPERIMENT RESULTS

From the collected training sets, I built LBP histograms. Then I used these histograms as inputs to the learning application and trained the face classifier.



Figure 10. Face detection results

Some of face detection results are shown in Figure 10. Even though trained classifier still suffers from dark skin faces, its

detection result was quite good.

Finally, I implement a face detector which uses trained face classifier as a S60 Symbian application. On VGA input, a face detection which use LBP-based classifier took 3.2 second to process whole image while Haar-like feature based classifier took 6.3 seconds to process image.

VII. CONCLUSIONS

In this project, I introduced and implemented a face detection algorithm, based on LBP features. Motivated by the fact that computing Haar-like features are too computationally heavy to work on mobile product, I utilize another feature which is computationally simpler than Haar-like feature.

Although LBP feature is simpler, my implementation shows that it is enough to discriminate faces and non faces faster.

REFERENCES

- [1] M. H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: a survey. *IEEE Trans. on PAMI*, 24(1):34-35, 2002
- [2] K. K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Trans. on PAMI*, 20(1):39-51, 1998..
- [3] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Trans. on PAMI*, 20(1):23-38, 1998.
- [4] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: an application to face detection. *Proc. of CVPR*, pages 130-136, 1997.
- [5] B. Heisele, T. Poggio, and M. Pontil. Face detection in still gray images. *A.I. Memo*, (1687), 2000
- [6] T.Mita, T. Kaneko, and O. Hori. Joint haar-like features for face detection. In *ICCV*, 2005.
- [7] C. P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. *Proc. of ICCV*, pages 555-562, 1998.
- [8] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proc. of CVPR*, pages 511-518, 2001.
- [9] J. Friedman, T. Hastie and R. Tibshirani. Additive Logistic Regression: a Statistical View of Boosting.
- [10] B. Wu, H. Ai, C. Huang, and S. Lao. Fast rotation invariant multi-view face detection based on Real AdaBoost. *Proc. of IEEE Conf. on Automatic Face and Gesture Recognition*, pages 79-84, 2004.
- [11] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. *Proc. of ICIP*, 1:900:903, 2002..
- [12] T. Ojala and M. Pietikainen. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns, *IEEE Trans on Pattern Analysis and Machine Intelligence*, Vol. 24. No.7, July, 2002.
- [13] A. Hadid, M. Pietikainen and T. Ahonen. A Discriminative Feature Space for Detecting and Recognizing Faces. *Proc of CVPR 2004*.
- [14] L. Jin, S. Satoh and M. Sakauchi. A Novel Adaptive Image Enhancement Algorithm for Face Detection. *Proc of ICPR 2004*.
- [15] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Machine Learning: Proc of the 13th International Conf.*