International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682
Vol.4, No.2, February 2015 DOI:10.15693/ijaist/2015.v4i2.193-198

Real Time Face Detection Using AdaBoost Algorithm

C. Anusuya.
PG scholar
M.E.VLSIDESIGN
Sethu Institute of Technology
(An Autonomous Institution)

Dr. R. Ganesan
PG Progam Head
M.E.VLSIDESIGN
Sethu Institute of Technology
(An Autonomous Institution)

C. Sujatha.
Associate Professor
Department of ECE
Sethu Institute of Technology
(An Autonomous Institution)

Abstract: A face detection framework that is capable of processing images extremely rapidly while achieving high detection performance. It consists of three key contributions. First is the introduction of a new image representation called the "Integral Image" which allows the features used by our detector to be computed frequently. The second is an efficient and simple classifier which is built using the AdaBoost learning algorithm to select a small number of critical visual features from a very large set of features. The third contribution is a method for combining classifiers in a "cascade" which allows background regions of the image to be quickly discarded while spending more computation on promising face regions. The system yields face detection performance comparable to the best previous methods, Implemented on a desktop, face detection proceeds at 15 frames per second.

Key Words: boosting, face detection, human sensing

I. INTRODUCTION

This paper brings together new algorithms and insights to construct a framework for robust and extremely rapid visual detection. Toward this end we have constructed a frontal face detection system which achieves detection and false positive rates which are equivalent to the best published results [7, 9, 13]. This face detection system is most clearly distinguished from previous approaches in its ability to detect faces rapidly. Operating on 384 by 288 pixel images, faces are detected 15 frames per second on a conventional 700 MHz .In face detection systems, auxiliary information, such as image differences in video sequences, or pixel color in color images, have been used to achieve high frame rates. Our system achieves high frame rates working only with the information present in a single grey scale image. These alternative sources of information can also be integrated with our system to achieve even higher frame rates.

There are three main contributions of our face detection framework. We will introduce each of these ideas briefly below and then describe them in detail in subsequent sections.

The first contribution of this paper is a new image representation called an *integral image* that allows for very fast feature evaluation. Motivated in part by the work of [11] our detection system does not work directly with image intensities. Likethese authors we use a set of features which are reminiscent of Haar Basis functions (though we will also use related filters which are more complex than Haar filters). In order to compute these features very rapidly at many scales we introduce the integral image representation for images (the integral image is very similar to the summed area table used in computer graphics [2]for texture mapping). The integral image can be computed from an image using a few operations per pixel. Once computed, any one of these Haar-like features can be computed at any scale or location in *constant* time.

The second contribution of this paper is a simple and efficient classifier that is built by selecting a small number of important features from a huge library of potential features using AdaBoost [5]. Within any image sub-window the total number of Haar-like features is very large, far larger than the number of pixels. In order to ensure fast classification, the learning process must exclude a large majority of the available features, and focus on a small set of critical features. Motivated by the work of [20]feature selection is achieved using the AdaBoost learning algorithm by constraining each weak classifier to depend on only a single feature. As a result each stage of the boosting process, which selects a new weak classifier, can be viewed as a feature selection process. AdaBoost provides an effective learning algorithm and strong bounds on generalization performance [16].

The third major contribution of this paper is a method for combining successively more complex classifiers in a cascade structure which dramatically increases the speed of the detector by focusing attention on promising regions of the image. The notion behind focus of attention approaches is that it is often possible to rapidly determine where in an image a face might occur [1, 3, and 20]. More complex processing is reserved only for these promising regions. The key measure of such an approach is the "false negative" rate of the attentional process. It must be the case that all, or almost all, face instances are selected by the attentional filter.

We will describe a process for training an extremely simple and efficient classifier which can be used as a "supervised" focus of attention operator.1 A face detection attentional operator can be learned which will filter out over 50% of the

image while preserving 99% of the faces (as evaluated over a large dataset). This filter is exceedingly efficient; it can be evaluated in 20 simple operations per location/scale (approximately 60 microprocessor instructions). Those sub-windows which are not rejected by the initial classifier are processed by a sequence of classifiers, each slightly more complex than the last. If any classifier rejects the sub-window, no further processing is performed. The structure of the cascaded detection process is essentially that of a degenerate decision tree, and as such is related to the work of [1,3].

The complete face detection cascade has 38 classifiers, which total over 80,000 operations. Nevertheless the cascade structure results in extremely rapid average detection times. On a difficult dataset, containing 507 faces and 75 million sub-windows, faces are detected using an average of 270 microprocessor instructions per sub-window. In comparison, this system is about 15 times faster than an implementation of the detection system constructed by [14]. An extremely fast face detector will have broad practical applications. These include user interfaces, image databases, and teleconferencing. This increase in speed will enable real-time face detection applications on systems where they were previously infeasible. In applications where rapid frame-rates are not necessary, our system will allow for significant additional post processing and analysis. In addition our system can be implemented on a wide range of small low power devices, including hand-helds and embedded processors. In our lab we have implemented this face detector on a low power 200 mips *Strong Arm* processor which lacks floating point hardware and have achieved detection at two frames per second.

A.Overview

The remaining sections of the paper will discuss the implementation of the detector, related theory, and experiments. Section 2 will detail the form of the features as well as a new scheme for computing them rapidly. Section 2.2 will discuss the method in which these features are combined to form a classifier. The machine learning method used, a application of AdaBoost, also acts as a feature selection mechanism. While the classifiers that are constructed in this way have good computational and classification performance, they are far too slow for a real-time classifier. Section 2.3 will describe a method for constructing a cascade of classifiers which together yield an extremely reliable and efficient face detector. Section 3 will describe a number of experimental results, including a detailed description of our experimental methodology. Finally Section 6 contains a discussion of this system and its relationship to related systems.

II. PROPOSED WORK

Our face detection procedure classifies images based on the value of the features. There are lots of motivations for using features rather than the pixels directly. The reason is that features can act to encode ad-hoc domain knowledge that is difficult to learn using a finite quantity of training data. There is also a second critical motivation for features: the feature-based system operates much faster than a pixel-based system. The features used are reminiscent of Haar basis functions which have been used by [11]. More specifically, we here use the three kinds of features.

The value of a two-rectangle feature is the difference between the sums of the pixels within two regions. The regions have the same size and shape and are horizontally or vertically adjacent. A three-rectangle feature computes the sum within two outside rectangles subtracted from the sum in a center rectangle .At last a four-rectangle feature computes the difference between diagonal pairs of rectangles.

A. Integral Image

Rectangle features can be computed rapidly using an intermediate representation for the image which we call the integral image. The integral image at a location *x, y* contains the sum of the pixels up and the left of *x, y* inclusive:

(1)

where $ii(x, y)$ is the integral image and $i(x, y)$ is the original image (see Fig. 2). Using the following pair of recurrences:
$s(x, y) = s(x, y-1) + i(x, y)$ (1)
$ii(x, y) = ii(x-1, y) + s(x, y)$ (2) (where $s(x, y)$ is the cumulative row sum, $s(x,-1) = 0$, and $ii(-1, y) = 0$) the integral image can be compute din one pass over the original image. The authors point out that in the case of linear operations (e.g. $f \cdot g$), any invertible linear operation can be applied to f or g if its inverse is applied to the result. In the case of convolution, if the derivative operator is applied for both to the image the result must then be double integrated:

(2)

The authors go on to show that convolution can be significantly accelerated if the derivatives of f and g are sparse. A similar insight is that an invertible linear operation can be applied to f if its inverse .Viewed in this framework computation of the rectangle sum can be expressed as a dot product, $i \cdot r$, where i is the image and r is the box car image (with value 1 within the rectangle of interest and 0 outside).

B. Feature Discussion

Rectangle features are somewhat primitive when compared with alternatives such as steerable filters [4, 6]. Steerable

filters and their relatives are excellent for the detailed analysis of the boundaries, image compression and the texture analysis. Rectangle features are also sensitive to the presence of the edges, bars and other simple image structure, they are quite coarse. Unlike steerable filters the orientations available are vertical, horizontal and diagonal. Since orthogonality is not central to this feature set, we choose to generate a very large and varied set of rectangle features. Typically the representation is about 400 times over complete. This over complete set provides features of arbitrary aspect ratio and of finely sampled location. Empirically it appears as though the set of rectangle features provide a rich image representation which supports effective learning. The computational efficiency of rectangle features provides ample compensation for their limitations.

In order to a *Feature Discussion* Rectangle features are somewhat primitive when compared with alternatives such as steerable filters [3, 8]. Steerable filters and their relatives are excellent for detailed analysis of the boundaries, image compression and the texture analysis. Rectangle features are also sensitive to the presence of the edges, bars and other simple image structure they are quite coarse. Unlike steerable filters the orientations available are vertical, horizontal and diagonal. Since orthogonality is not central to this feature set, we choose to generate a very large and varied set of rectangle features.

Typically the representation is about 400 times over complete. This over complete set provides features of arbitrary aspect ratio and of finely sampled location. Empirically it appears as though the set of rectangle features provide a rich image representation which support effective learning. The computational efficiency of rectangle features provides ample compensation for their limitations. In order to aim our system a variant of AdaBoost is used both to select the features and to train the classifier [5]. In its original form, the AdaBoost learning algorithm is used to boost the classification performance of a simple learning algorithm. It done by, combining a collection of weak classification functions to form a stronger classifier. The language of boosting the simple learning algorithm is called a weak learner. For example the perceptron learning algorithm searches over the set of possible perceptron and returns the perceptron with the lowest classification error. So, the learner is called weak because we do not expect even the best classification function to classify the training data well (i.e. for a given problem the best perceptron may only classify the training data correctly 51% of the time). For the weak learner to be boosted it is called to solve a sequence of learning problems. After first round of learning, the example is re-weighted in order to emphasize those which were incorrectly classified by the previous weak classifier. The strong classifier takes the form of a perceptron a combination of weak classifiers followed by a threshold.

The conventional AdaBoost procedure can be easily interpreted as the greedy feature selection processes .The general problem of boosting in which a set of classification function is combined using a majority vote. The main challenge is to associate a large weight with each good classification function and a smaller weight with poor function.

AdaBoost algorithm is an aggressive mechanism for selecting a small set of good classification functions which have significant variety. An analogy between weak classifiers and features AdaBoost is an efficient procedure for searching out a small number of good “features” which have significant variety. One method for completing this analogy is to restrict the weak learner to the set of classification functions each of which depend on a feature. In base of this goal, the weak learning algorithm is designed to select the single rectangle feature which best separates the positive and negative examples (this is similar to the approach of Tieu and Viola (2000) in the domain of image database retrieval). For each feature, the weak learner determines the optimal threshold classification function that the minimum number of examples is misclassified. A weak classifier ($h(x, f, p, \theta)$) thus consists of a feature (f), a threshold (θ) and a polarity (p) indicating the direction of the inequality:

(3)

Here x is a 24 × 24 pixel sub-window of an image. In practice no single feature can perform the classification task with low error. Features are selected early in the process yield error rate between 0.1 and 0.3. Features selected in later rounds as the task become more difficult error rate between 0.4 and 0.5. This section describes an algorithm for constructing a cascade of classifiers which achieves increased detection performance while radically reducing computation time. The key is smaller, and more efficient, the boosted classifiers can be constructed which reject many of the negative sub-windows while detecting all the positive instances. Classifiers are used to reject the majority of sub-windows before more complex classifiers are called upon to achieve low false positive rates. The detection performance of the two-feature classifier is far from acceptable as a face detection system. The classifier can significantly reduce the number of sub-windows that need further processing with very few operations:

1. Evaluate the rectangle features (requires between 6 and 9 array references per feature).
2. Compute the weak classifier for each feature (requires one threshold operation per feature).
3. Combine the weak classifiers (requires one multiply per feature, an addition and a threshold).

C. Training a Cascade of Classifiers

The cascade design process is driven from a set of detection and performance. For the face detection process, some systems have achieved good detection rates (between 85 and 95 percent) and extremely low false positive rates (on the order of 10−5 or 10−6). The number of cascade steps and the size of each step must be sufficient to achieve similar detection performance while minimizing computation. A trained cascade of classifiers the false positive rate of the cascade is

(4)

where *F* is the false positive rate of the cascaded classifier, *K* is the number of classifiers and *fi*, is the false positive rate of the *i* th classifier. The detection rate is

(5)

where *D* is the detection rate of the cascaded classifier, *K* is the number of classifiers and *di*, is the detection rate of the *i*th classifier. The number of features evaluated when scanning real images is necessarily a probabilistic process. Sub-window will progress down the cascade classifier at a time. Until it decided that the window is negative or the window succeeds in each test and is labeled proved. The expected behavior of this process is determined by the distribution of image windows on a typical test set. The key measure of each classifier is it positive rate the proportion of windows which are labeled as potentially containing a face. The expected number of features which are evaluated is:

(6)

Where *N* is the expected number of features evaluated, *K* is the number of classifiers, p*i* is the positive rate of the *i*th classifier and *ni*, are the number of features in the *i*th classifier. Faces are extremely rare, the "positive rate" is effectively equal to the false positive rate. The process by which each element of the cascade is trained wants some care.

The AdaBoost learning procedure attempts only to minimize error and is not specifically designed to achieve high detection rates at the expense of large false positive rate. Simple and very conventional scheme for trading off errors is to adjust the threshold of the perceptron produced by AdaBoost. Higher thresholds classifiers with fewer false positives and a lower detection rate. Lower thresholds, with the classifiers, with more false positive and a higher detection rate. It is not clear whether adjusting the threshold in this way preserves the training and generalization guarantees provided by AdaBoost. The overall training process contains two types of the tradeoffs. In most the cases classifiers with more features will achieve higher detection rates and lower false positive rates. The classifiers with more features require more time to compute. One could define an optimization framework in which

- The number of classifier stages,
- The number of features, *ni*, of each stage,
- The threshold of each stage

are traded off in order to minimize the expected number of features. Finding this optimum is a tremendously difficult problem.

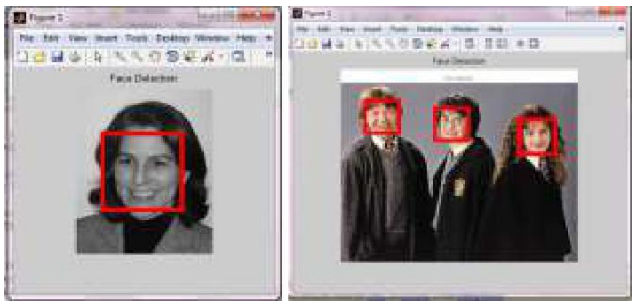
III. EXPERIMENTAL RESULTS

This section presents some of the considerations regarding the implementation of the Viola-Jones face detector along with intermediate and final results. The final detector will only detect faces similar to those in the training set so the training set should represent the most common facial variation.

	False detections			
detector	10	30	51	65
Viola-jones	81.1%	89.7%	92.1%	93.1%
Rowley-Baluja-Kanade	83.2%	86.0%	–	–

Table 3.1 Detection rates for various numbers of false positives

Table 3.1 lists the detection rate for various numbers of false detections for system. For the Rowley- Baluja-Kanade results, a number of different versions of their detector were tested yielding a number of different results.



IV. CONCLUSIONS

An approach for face detection which minimizes computation time while achieving high detection accuracy .The approach was used to construct a face detection system which is

approximately 15 times faster than other approach. Experiments, which will be described show that highly efficient detectors for other objects, such as pedestrians or automobiles.

The integral image reduces the initial image processing required for face detection significantly. The second contribution of this paper is a simpleand efficient classifier built from computationally efficient features using AdaBoost for feature selection. The third contribution of this paper is a technique for constructing a cascade of classifiers which radically reduces computation time while improving detection accuracy. This dataset includes faces under a very wide range of conditions including: illumination, scale, pose, and camera size variation. Experiments on such a large and complex dataset are difficult and time consuming. Systems which work under these conditions are unlikely to be brittle or limited to a single set of conditions.

REFERENCES

[1] Amit, Y. and Geman, D. 1999. A computational model for visual selection. Neural Computation, 11:1691–1715.

[2] Crow, F. 1984. Summed-area tables for texture mapping. In Proceedings of SIGGRAPH, 18(3):207–212.

[3] Fleuret, F. and Geman, D. 2001. Coarse-to-fine face detection. Int. J. Computer Vision, 41:85–107.

[4] Freeman,W.T. and Adelson, E.H. 1991. The design and use of steerable filters. IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(9):891–906.

[5] Freund, Y. and Schapire, R.E. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. In Computational
Learning Theory: Eurocolt 95, Springer-Verlag, pp.23–37.

[6] Greenspan, H., Belongie, S., Gooodman, R., Perona, P., Rakshit, S., and Anderson, C. 1994. Overcomplete steerable pyramid filters and rotation invariance. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[7] Itti, L., Koch, C., and Niebur, E. 1998. A model of saliency-based visual attention for rapid scene analysis. IEEE Patt. Anal. Mach.Intell., 20(11):1254–1259.

[8] John, G., Kohavi, R., and Pfeger, K. 1994. Irrelevant features and the subset selection problem. In Machine Learning Conference Proceedings.

[9] Osuna, E., Freund, R., and Girosi, F. 1997a. Training support vector machines: An application to face detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[10] Osuna, E., Freund, R., and Girosi, F. 1997b. Training support vector machines: an application to face detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[11] Papageorgiou, C., Oren, M., and Poggio, T. 1998. A general framework for object detection. In International Conference on Computer Vision.

[12] Quinlan, J. 1986. Induction of decision trees. Machine Learning,1:81–106.

[13] Roth, D., Yang, M., and Ahuja, N. 2000. A snowbased face detector. In Neural Information Processing 12.

[14] Rowley, H., Baluja, S., and Kanade, T. 1998. Neural network-based face detection. IEEE Patt. Anal. Mach. Intell., 20:22–38.

[15] Schapire, R.E., Freund, Y., Bartlett, P., and Lee, W.S. 1997. Boosting the margin: A new explanation for the effectiveness of voting methods. In Proceedings of the Fourteenth International Conference on Machine Learning.

[16] Schapire, R.E., Freund, Y., Bartlett, P., and Lee, W.S. 1998. Boosting the margin: A new explanation for the effectiveness of voting
methods. Ann. Stat., 26(5):1651–1686.

[17] Schneiderman, H. and Kanade, T. 2000. A statistical method for 3D object detection applied to faces and cars. In International Conference on Computer Vision.

[18] Simard, P.Y., Bottou, L., Haffner, P., and LeCun, Y. (1999). Boxlets: A fast convolution algorithm for signal processing and neural networks.
In M. Kearns, S. Solla, and D. Cohn (Eds.), Advances in Neural Information Processing Systems, vol. 11, pp. 571– 577.

[19] Sung, K. and Poggio, T. 1998. Example-based learning for viewbased face detection. IEEE Patt. Anal. Mach. Intell., 20:39–51.

[20] Tieu, K. andViola, P. 2000. Boosting image retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[21] Tsotsos, J., Culhane, S., Wai, W., Lai, Y., Davis, N., and Nuflo, F. 1995. Modeling visual-attention via
selective tuning. Artificial Intelligence Journal, 78(1/2):507–545.

[22] S. Chen and C. Liu(2011),"Discriminant analysis of haar features for accurate eye detection," in Proc. 15th Int. Conf. Image Process., Comput. Vis., Pattern Recognit., Las Vegas, NV, USA, pp. 1–7.

[23] Z. Gu, J. Yang, and L. Zhang(2010), "Push-Pull marginal discriminant analysis for feature extraction," Pattern Recognit. Lett., vol. 31, no. 15, pp. 2345–2352.

[24] H. Kong, L. Wang, E. K. Teoh(2005), J.-G. Wang, and R. Venkateswarlu, "A framework of 2D Fisher discriminant analysis: Application to face recognition with small number of training samples," in Proc. IEEE Comput. Soc. Conf. CVPR, pp. 1083–1088.

[25] C. Liu and H. Wechsler(2002), "Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition," IEEE Transactions On Image Processing, vol. 11, no. 4, pp. 467–476.

[26] C. Liu(2003), "A Bayesian discriminating features method for face detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, no. 6, pp. 725–740.

[27] M.Loog and Robert P.W. Duin(2004), "Linear Dimensionality Reduction Via A Heteroscedastic Extension Of LDA: The Chernoff Criterion", IEEE Transactions On Pattern Analysis And Machine Intelligence, vol. 26, no. 6,pp.732-739.

[28] C. Papageorgiou, M. Oren, and T. Poggio(1998),"A general framework for object detection," in Proc. IEEE 6th Int. Conf. Comput. Vis., pp. 555–562.
[29] Shuo Chen and Chengjun Liu(2014), "Clustering based discriminant analysis for eye detection",IEEE Transactions On Image Processing, vol.23,no.4,pp. 1629-1638.
[30]Zhifeng Li, Dahua Lin, and Xiaoou Tang(2009), "Nonparametric Discriminant Analysis For Face Recognition", IEEE Transactions On Pattern Analysis And Machine Intelligence, vol. 31, no. 4,pp. 755-761.
[31]Zhiming Liu And Chengjun Liu(2008),"A Hybrid Color And Frequency Features Method For Face Recognition", IEEE Transactions On Image Processing, vol. 17, no. 10,pp. 1975-1980.

AUTHORS PROFILE

C.Anusuya received her **B.E.** degree in Electronics and Communication Engineering from K.L.Ncollege of Engineering , Madurai, Anna University, Chennai, India, in 2013. Pursuing **M.E.** degree in VLSI Design from Sethu Institute of Technology, Anna University, Chennai, India. She was published one conference. Her research interest includes VLSI design ,image processing and Testing.

Dr.R.Ganesan received his B.E. in Instrumentation & Control Engineering from Arulmigu Kalasalingam College Of Engineering and **ME** (Instrume- ntation) from Madras Institute of Tech- nology in the year 1991 and 1999 respectively. He has completed his **PhD** from Anna University, Chennai, India in 2010. He is presently working as Professor and head in the department of M.E-VLSI Design at Sethu Institute of Technology, India. He has published more than 25 research papers in the National & International Journals/ Conferences. His research interests are VLSI design, Image Processing, Neural Networks and Genetic algorithms.

MRS.C.SUJATHA received **B.E** degree in Electronics and Communication Engineering from P.S.N.A College of Engineering and Technology, Dindigul, Tamil Nadu,India in 1997 and **M.E** degree in Applied Electronics from P.S.N.A College of Engineering and Technology, Dindigul, Tamil Nadu, India in 2004. She has more than 13 years of teaching experience. Presently she is working as Associate Professor, Department of Electronics and Communication Engineering, Sethu Institute of Technology, Kariapatti, Tamil Nadu, India. She is doing research in the area of Image Processing based on FPGA under the guidance of Dr.D.Selvathi, Prof. /ECE Dept., Mepco Schlenk Engineering College, Sivakasi,. Tamilnadu, India.