The Viola-Jones Face Detection Algorithm Analysis: A Survey

Ahmed A. Elngar[1], Mohamed Arafa[2], Abd El Rahman Ahmed Naeem[3], Ahmed Rushdy Essa[4], Zahra Ahmed shaaban[5*].

[1]Faculty of Computers and Artificial Intelligence, Beni-Suef University, Beni Suef City, 62511, Egypt
E-mail: elngar_7@yahoo.co.uk

[2] Teaching assistant, Department of Computer Science, Scientific Innovation Research Group (SIRG) member, Beni Suef University of Computers and Artificial Intelligence, E-mail: jenuiman4rt@gmail.com

[3] Under Graduated Student, Department of Computer Science, Scientific Innovation Research Group (SIRG) member, Beni Suef University of Computers and Artificial Intelligence, E-mail: dsaa2295@gmail.com

[4] Under Graduated Student, Department of Computer Science, Scientific Innovation Research Group (SIRG) member, Beni Suef University of Computers and Artificial Intelligence, E-mail: ahmed.639963@gmail.com

[5] Under Graduated Student, Department of Computer Science, Scientific Innovation Research Group (SIRG) member, Beni Suef University of Computers and Artificial Intelligence, E-mail: zuziiahmed95@gmail.com

Corresponding Author: (Zahra Ahmed shaaban[5*], zuziiahmed95@gmail.com)

Abstract: In this paper, we analysis the Viola-Jones algorithm, the most real-time face detection system has been used. It is consisting from three main concepts to enable a robust detection: the integral image for Haar feature computation, Adaboost for selecting feature and cascade to make resource allocation more efficient. Here we propose each stage starting from Integral image to the end with Cascading and some of algorithmic description for stages. The Viola-Jones algorithm gives multiple detections, a post-processing step which reduce detection redundancy using Adaboost and cascading.

Keywords: face detection, Viola-Jones algorithm, Integral Image, Adaboost, Haar feature, cascade.

Received: February 25, 2021 Revised: March 30, 2021 Accepted: April 10, 2021
DOI: 10.5281/zenodo.4898039

1. Introduction.

The main purpose of face detection is to tell us if there are a face in an arbitrary size contains a human face and if it detects a face tell us where it is. Binary classification used to determine if there is face or not, on the other hand the classifier is constructed to make the misclassification risk minimized. Since no objective distribution can describe the particular prior probability for a given image to possess a face, the algorithm must minimize both the false negative and false positive rates so as to realize a good performance. This task needs an accurate numerical description of what sets human faces aside from other objects.

There are three main components of our object detection framework. We will mention these ideas shortly below and then describe them in detail in subsequent sections.

The first section of this paper is an image representation called an integral image that gives a very fast feature evaluation. Haar Basis functions used. The integral image representation for images used to compute these features rapidly at different scales. And by performing few operations per pixel on the original image the integral image can be computed. After computed, any one of Harr-like features computed easily at any scale or location in constant time.

The second section of this paper is the way to constructing a classifier to minimize the selected important features using AdaBoost. Inside any sub window, there is a large number of Harr-like features, far larger than the number of pixels. To ensure making the classification faster, unimportant features must be excluded from the large majority of the available features, and focus on a small set of important features. The feature selection is achieved through by some modification of the AdaBoost algorithm: the weak learner is constrained so that each weak classifier returned can depend on only a single feature. As a result, each stage of the boosting process, a new weak classifier selected. AdaBoost gives us an effective learning algorithm and strong bounds on generalization performance.

The third section of this paper is a method for merging more complex classifiers in a cascade structure which make the detector faster by focusing on particular regions of the image.

Received: February 25, 2021 Revised: March 30, 2021 Accepted: April 10, 2021
DOI: 10.5281/zenodo.4898039

The notion behind focus of attention approaches is that it is often possible to rapidly determine where the object might be found in the image. These particular regions need more complex processing.

2. Haar Features discussion

The value of simple features is the base of Viola-Jones object detection procedure classifies images. There is much encouragement for using features instead of the pixels directly. The most familiar purpose is that features can act to encode ad-hoc domain knowledge that is heavy to learn using a limited quantity of training data. data. For this system there is also a second critical motivation for features: the feature-based system operates much faster than a pixel-based system. [1]

The simple features used are like Haar basis functions which are used by Papa Georgiou et al. More specifically, we use three different features. The result of a two-rectangle feature is that the difference between the sum of the pixels within two rectangular regions. The regions have an equivalent size and shape and are horizontally or vertically adjacent (see Figure 1). A three-rectangle feature computes the sum within two outside rectangles subtracted from the sum during a center rectangle. Finally, the difference between diagonal pairs of rectangles is computed by a four-rectangle feature.[3]

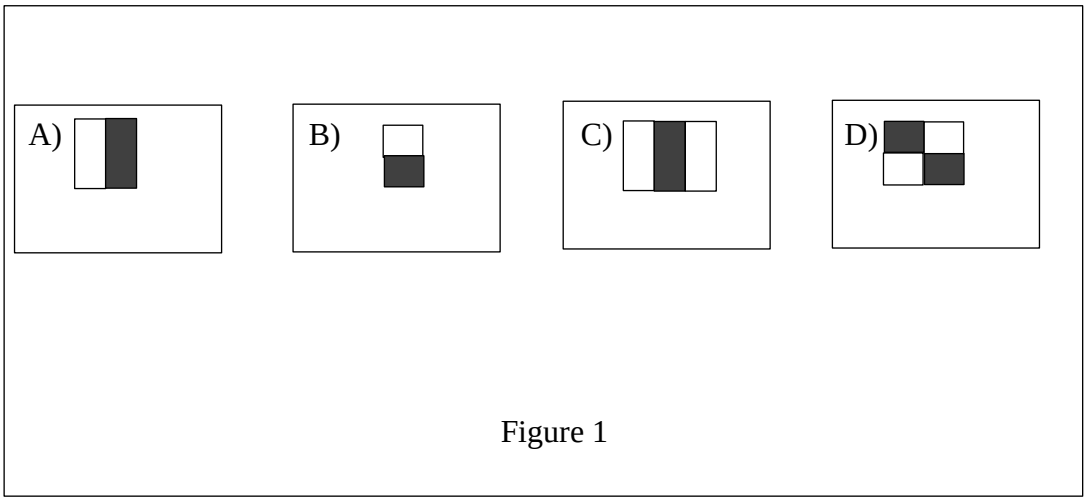

Figure 1

Received: February 25, 2021 Revised: March 30, 2021 Accepted: April 10, 2021
DOI: 10.5281/zenodo.4898039

Given that the base resolution of the detector is 24x24, the exhaustive set of rectangle features is sort of large, over 180,000. Note that unlike the Haar basis, the set of rectangle features is overcomplete.[2]

2.1. Integral Image

Using an intermediate representation for the image which we call the integral image rectangle features can be computed very rapidly. The integral image at location contains the sum of the pixels above and to the left of, inclusive.[1]

$$ii(x,y) = \sum_{x' \le x, \dot{y}' \le y} i(x', y')$$

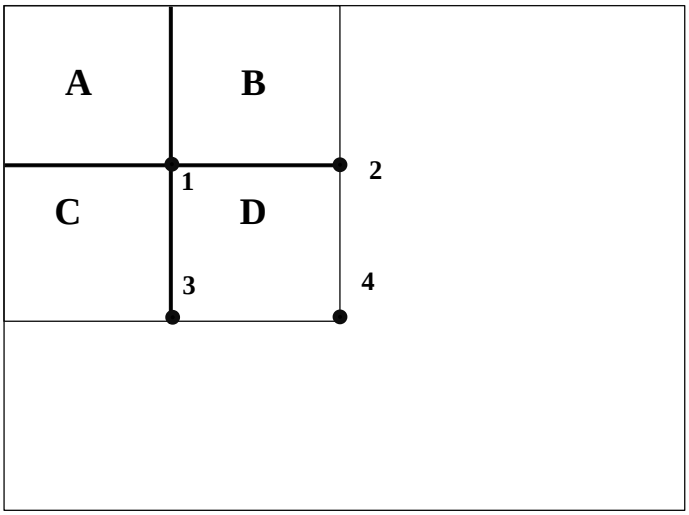

Figure 2: The sum of the pixels within rectangle **D** can be computed with four array references. The value of the integral image at location **1** is the sum of the pixels in rectangle **A**. The value at location 2 is **A + B** at location **3** is **A + B + D**, and at location **4** is **A + B + C + D**. The sum within **D** can be computed as

4 + 1 - (3 + 2)

Received: February 25, 2021 Revised: March 30, 2021 Accepted: April 10, 2021
DOI: 10.5281/zenodo.4898039

Where $ii(x, y)$ is the integral image and $ii(x, y)$ is the original image. Using the following pair of recurrences:

$$s(x, y) = s(x, y - 1) + i(x, y) \quad (1)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \quad (2)$$

(where $s(x, y)$ is the cumulative row sum, $s(x, -1) = 0$, and $ii(-1, y) = 0$ the integral image can be computed in one pass over the original image)[1]

Using the integral image any rectangular total is computed in four array references (see Figure 2). Clearly, the difference between 2-rectangular sums is computed in eight references. Since the 2-rectangle features outlined higher than involve adjacent rectangular sums they will be computed in six array references, eight within the case of the 3-rectangle features, and nine for 4-rectangle features.[1]

2.2 Feature Discussion

Rectangle features area unit somewhat primitive when put next with alternatives like steerable filters[4,5]. steerable filters, and their relatives, area unit wonderful for the elaborate analysis of boundaries, compression, and texture analysis. In distinction rectangle features, whereas sensitive to the presence of edges, bars, and alternative straightforward image structures, are quite coarse. in contrast to steerable filters, the sole orientations available are vertical, horizontal, and diagonal. The set of rectangle features do but offer an upscale image representation that supports effective learning. In conjunction with the integral image, the efficiency of the rectangle feature set provides ample compensation for his or her restricted flexibility.

3. Boosting algorithm discussion

The algorithms which are converting the weak classifier to strong one is called Boosting. The weak learners are classifiers which classifies just slightly better than a random classifier. These

Received: February 25, 2021 Revised: March 30, 2021 Accepted: April 10, 2021
DOI: 10.5281/zenodo.4898039

are thus classifiers which have piece of information on how to predict the right outputs, but not like strong classifiers have like: Naive Bayes, Neural Networks or SVM. So, the strong classifier has been found by combining the prediction of weak classifiers by using weighted average and Considering prediction has higher vote

3.1 Boosting Algorithm process

We apply base learning (ML) algorithms with a different distribution to find weak learner. On all times base learning algorithm is applied, a new weak prediction rule is generated. And it's an iterative process. After many iterations, a single strong prediction rule has been created by the combines of these weak rules. And the choose of the different distribution for each round is done by:

1: The base learner takes all of the distributions and give all of them an equal weight.

2: In case of any prediction error happened by first base learning algorithm, then we increase attention for the observations that have prediction error.

3: Repeat the second step until reaching the limit of base learning algorithm or achieving higher accuracy.

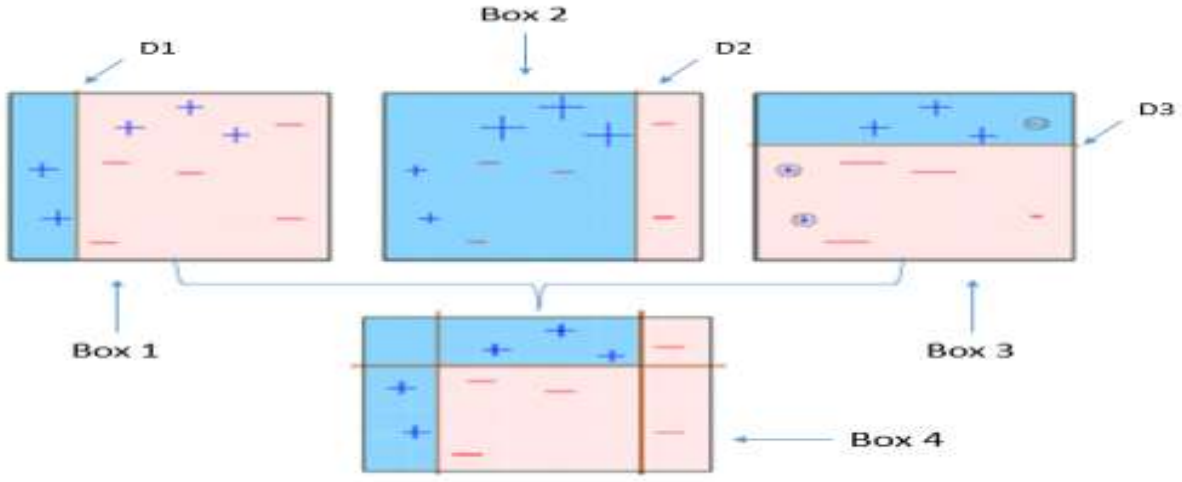

Figure 3: adaboosting steps

At the end, it combines the outputs of weak classifiers and creates a strong one which get better results of the prediction power of the model.

Received: February 25, 2021 Revised: March 30, 2021 Accepted: April 10, 2021
DOI: 10.5281/zenodo.4898039

4.2 Types of Boosting Algorithms

1. AdaBoost (**Ada**ptive **Boost**ing)
2. Gradient Tree Boosting
3. XGBoost

AdaBoost (Adaptive Boosting) working on similar method. It fits a set of weak learners on different weighted data for training. At start it gives equal weights to each observation in the original data set. If there is an error in the prediction when using the first classifier, then it takes higher weight in the next iterate for the incorrect predictions. Being an iterative process, it repeats by adding learner(s) until a limit is reached in the number of models or accuracy. Mostly, we use decision stamps with AdaBoost. [2]

3.3 Adaboost with face detection

- Adaboost helps to find only the best features among all these 160,000+ features that have been created by Haar features. After these features are found a weighted combination of all these features in used in evaluating and deciding any given window has a face or not. Each of the selected features are considered okay to be included if they can at least perform better than random guessing (detects more than half the cases).

- These features are also called as weak classifiers. Each Haar feature is a weak classifier. Adaboost constructs a strong classifier as a linear combination of these weak classifiers.

The equation of combining weak classifiers to generate strong classifier =:

$$\mathbf{F(x)} = \propto \mathbf{1}\boldsymbol{f}\mathbf{1}(\boldsymbol{x}) + \propto \mathbf{2}\boldsymbol{f}\mathbf{2}(\boldsymbol{x}) + \propto \mathbf{3}\boldsymbol{f}\mathbf{3}(\boldsymbol{x}) + \cdots$$

3.4 AdaBoost Algorithm

Received: February 25, 2021 Revised: March 30, 2021 Accepted: April 10, 2021
DOI: 10.5281/zenodo.4898039

- input images (x_1, y_1), ..., (x_n, y_n) where y_i = 0, 1 for the positive a negative example respectively.
- Giving weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, for m and l are the number of positive and negative.
- For t = 1, . . ., T:
    1. Normalizing the weights,

$$w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,i}} \qquad w_t \text{ is a probability distribution}$$

    2. For each feature j training a classifier h_j which is restricted to using a single feature. The error is calculated with respect to w_t, $\varepsilon_j = \sum_i w_i \mid h_j(x_i) - y_i \mid$.
    3. Choose the classifier, h_t with the lowest error.
    4. Update the weights:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

    5. Where e_i = 0 if example x_i is classified correctly, otherwise $e_i = 1$ and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$

- At the end, the strong classifier formatted as:

$$h_{(x)} = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & otherwise \end{cases}$$

4.Cascading process

The process of organizing the set of features in a multilevel classification form is called cascade classifier. In the cascade are constructed by training classifiers using AdaBoost and then

Received: February 25, 2021 Revised: March 30, 2021 Accepted: April 10, 2021
DOI: 10.5281/zenodo.4898039

adjusting the threshold to minimize false negatives. Within the default AdaBoost threshold is designed to yield a low error rate on the training data. Generally, a lower threshold yields higher detection rates and higher false positive rates.

There are at least three classifications to work out whether or not there are facial features in the selected feature area. Within the first stage classification filter, each sub-image can be classified using one feature, if the value of the feature on the filter does not match the expected output, it will be rejected. Then the algorithm moves next to the sub-window and calculates the value of the feature, if the results are following the desired threshold then it will proceed to the second filter stage until the number of sub-windows that pass the classification decrease until it approaches the detected face image. [2]

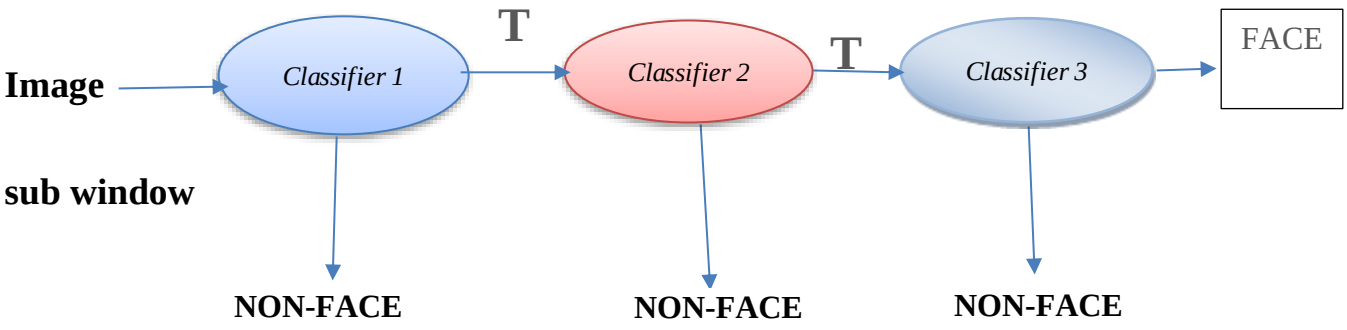

Figure 4: cascading steps

5.1 Training a Cascade of Classifiers

The process of cascading training contains two types of tradeoffs. In most cases classifiers that have more features will achieve lower false positive rates and higher detection rates. At an equivalent classifier with more features require longer time to compute. In principle one could define an optimization framework in which: I) the number of classifier stages, ii) the number of features in each stage, and iii) the threshold of each stage, are traded off in order to decrease the expected number of evaluated features. Unfortunately finding this optimum maybe a tremendously difficult problem. In practice using a simple framework to produce an effective classifier which have a highly efficient. In each step in the cascade optimize the false positive rate and decreases the detection rate. A target is selected for the minimum reduction in false positives and the maximum decrease in detection. Each step is trained by adding features until the target detection and false positives rates are met

Received: February 25, 2021 Revised: March 30, 2021 Accepted: April 10, 2021
DOI: 10.5281/zenodo.4898039

(these rates are determined by testing the detector on a validation set). The stages are added until the thorough target for false positive and detection rate is matched.[1]

Conclusion

We have presented an approach for face detection which decrease computation time and achieving high detection accuracy. The approach was used to construct a face detection system which is approximately 15 fasters than any previous approach.

In this paper we have concluded that combination of Viola Jones algorithm and Principal component analysis gives result with fast detection and high accuracy. Some drawbacks of these techniques are high computation time when image size is large and resolution is high and if high dimensional data is used problem arises. Future work is to increase the accuracy of the face detection and recognition. Viola Jones Algorithm is a fast face detection technique but it has some false positive values for images with occluded faces, so future work is to try to reduce the false positive values.

References

[1] P. Viola and M. Jones, "Managing work role performance: Challenges for twenty-first century organizations and their employees.," *Rapid Object Detect. using a Boost. Cascade Simple Featur.*, pp. 511–518, 2001.

[2] Y.-Q. Wang, "An Analysis of the Viola-Jones Face Detection Algorithm," *Image Process. Line*, vol. 4, pp. 128–148, 2014, doi: 10.5201/ipol.2014.104.

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

[4] William T. Freeman and Edward H. Adelson. The design and use of steerable filters. IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(9):89 1-906, 1991.

[5] H. Greenspan, S. Belongie, R. Gooodman, P. Perona, S. Rakshit, and C. Anderson. Overcomplete steerable pyramid fil

[6] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory: Eurocolt '95*, pages 23–37. Springer-Verlag, 1995.

Received: February 25, 2021 Revised: March 30, 2021 Accepted: April 10, 2021

DOI: 10.5281/zenodo.4898039

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

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

[9] Edgar Osuna, Robert Freund, and Federico Girosi. Training support vector machines: an application to face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

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

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

[12] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. In *IEEE Patt. Anal. Mach. Intell.*, volume 20, pages 22–38, 1998.

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

[14] Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. In *Proceedings of the Fourteenth International Conference on Machine Learning*,1997.

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

[16] K. Sung and T. Poggio. Example-based learning for viewbased face detection. In *IEEE Patt. Anal. Mach. Intell.*, volume 20, pages 39–51, 1998.

[17] J.K. Tsotsos, S.M. Culhane,W.Y.K.Wai, Y.H. Lai, N. Davis, and F. Nuflo. Modeling visual-attention via selective tuning. *Artificial Intelligence Journal*, 78(1-2):507–545, October1995.

[18] Andrew Webb. *Statistical Pattern Recognition*. Oxford University Press, New York, 1999.

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

[20] F. Fleuret and D. Geman. Coarse-to-fine face detection. *Int.J. Computer Vision*, 2001.

Received: February 25, 2021 Revised: March 30, 2021 Accepted: April 10, 2021

DOI: 10.5281/zenodo.4898039