Department of Electrical and Computer Systems Engineering

Technical Report MECSE-13-2005

A new boostrapping strategy for the AdaBoost-based face detector

T.-J. Chin and D. Suter

MONASH UNIVERSITY

A new boostrapping strategy for the AdaBoost-based face detector

Tat-Jun Chin*and David Suter

8th Mar 2005

Abstract

False alarms occur in the face detection process when non-face sub-images are classified as face images, and are a major problem in face detection. Usually practitioners perform "bootstrapping"to refine the face detector to account for the false alarms (or false positives). For AdaBoost-based face detection (specifically, the method propounded by Viola and Jones in [5]), two approaches of bootstrapping are used–incremental retraining of the entire face detector with the false positives as examples, or construct additional classifier stages to eliminate the false positives. While incremental retraining is costly in terms of training time and effort, additional stages will ultimately slow down the face detector, thus defeating the purpose of constructing the detector in a cascade manner for speed-ups. This technical report presents a novel bootstrapping strategy that although is similar in nature to the second approach, has some modifications with the aim of improving the efficiency of the bootstrapping procedure.

1 Introduction

Face detection is essentially a two class pattern classification problem i.e. given a feature vector, decide whether the object instance should be given label C_1 or C_2. As with the general problem of object detection in static images, face detection is difficult as the object detection system is required to distinguish a particular class of objects from all others [9], i.e. faces from the rest of the world. For a model which is obtained through learning algorithms, the classifier has to be trained on samples that provide good representations of the underlying distributions of the classes that are to be differentiated. Whereas the face class is well-defined (e.g. two eyes, a nose, a mouth), there are no typical examples of the negative class. Gathering enough samples that provide a good representation of

*Tat-Jun Chin is a holder of the Australia-Asia Awards conferred by the Department of Education, Science and Training (DEST) of the Government of Australia since Feb 2004.

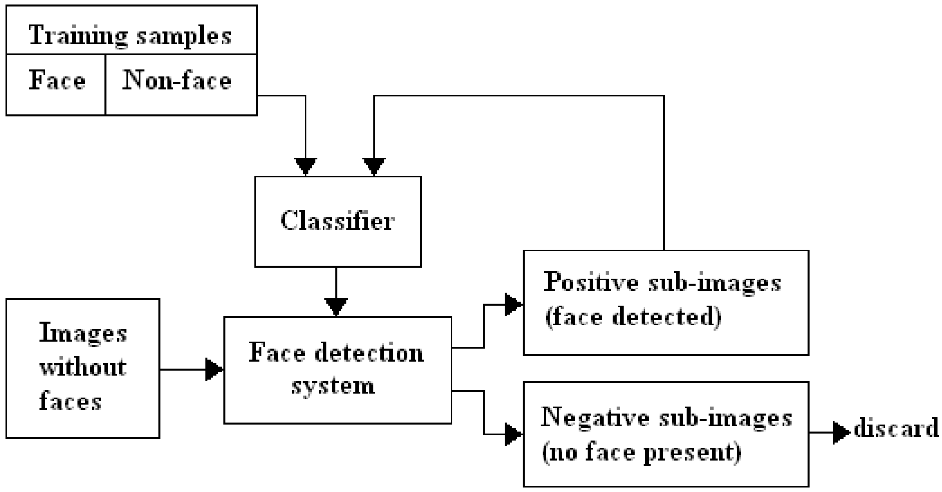

Figure 1: Incremental bootstrapping to account for false detections.

the negative class is non-trivial, as anything that does not look like a face belongs to this class!

Consequently, for learning-based face detection methods, false detection (non-face sub-images giving positive result for faces) is a major problem. Most researchers resort to "bootstrapping"to solve this problem (see [1, 10, 15]). In this training method, a trained classifier is subjected to arbitrary input images that do not contain faces. Any positive detections on these images are false detections and are collected for retraining of the classifier. This incremental refinement of the decision surface is iterated until satisfactory performance is achieved (refer Figure 1). Take note that each round of bootstrapping requires a re-training of the classifier.

While most face detection algorithms can adapt the bootstrapping approach shown in Figure 1 almost unmodified to account for false detections, due to the unique "cascade"structure of the AdaBoost face detector, there can be several interpretations as to how bootstrapping can be performed on a face detector of such nature. The cascade structure of the AdaBoost face detector allows it to implement the so-called "coarse-to-fine"search strategy in face detection, as well as achieving significant detection speed. With these advantages, retaining the cascade structure is certainly within our interest.

This technical report attempts to discuss a novel approach conceived to efficiently bootstrap the AdaBoost face detector, as well as providing some initial implementation results of the proposed bootstrapping algorithm. As a summary of the report: Section 2 provides a brief introduction to the AdaBoost-based face detector proposed by Viola and Jones; Section 3 presents a concise description of the AdaBoost learning algorithm as well as the proposed bootstrapping approach; Some experimental results of the proposed method and the design of the experiments are presented in Section 4; Finally, the conclusion and some discussion are presented in Section 5.

1.1 Previous work

There has been little work in the literature that is dedicated to in-depth studies of bootstrapping strategies that can be used to complement the training of a face detector. Many researchers resort to bootstrapping to account for false positives, and the bootstrapping strategies used is similar to the one shown in Figure 1. It is worth noting that in many face detection literature, since false positives are unavoidable, the bootstrapping stage is incorporated into the training stage.

In [15], Sung and Poggio created their face detector by first partitioning the image space into several clusters for face and non-face clusters. Each cluster is then further decomposed into the PCA and null subspaces. The Bayesian estimation is then applied to obtain useful statistical features. With regards to accounting for false positives, Sung and Poggio suggested collecting all the non-face patterns that the current system wrongly classifies as faces, and adding these wrong results to the training database as new negative samples.

In [1], Henry Rowley et al. trained a face detector based on retinally connected MLP neural networks. Through a sliding window, the input images are examined after going through an extensive preprocessing stage. Although not achieving high detection speeds, the detection results obtained from their method would set the benchmark for comparisons of face detection algorithms. The bootstrapping strategy applied is no different from the one used in [15], i.e. "collect sub-images in which the network incorrectly identifies a face... apply the preprocessing steps and add them into the training set as negative examples".

In [11], Roth et al. used the SNoW (Sparse Networks of Winnows) learning algorithm for face detection. In their system, Boolean features that encode the positions and intensity values of pixels are used. A high detection rate (more than 90%) is reported in their paper. Furthermore, they acknowledged that it is extremely difficult to collect a representative set of non-face examples. To solve this problem, they use the similar bootstrap strategy that is to "include more non-face examples during training".

2 The AdaBoost face detector

Specifically, the face detector mentioned here is the one propounded by Viola and Jones in [5]. They built a simple and efficient classifier which was trained using the AdaBoost learning algorithm [4]. The algorithm selects a small number of critical visual features from a very large set of potential features. Their face detector was realized in a cascade structure which allows extremely rapid processing of input images. Their system yielded detection accuracy comparable to the best previous systems (e.g. [15, 1, 13]) but is far superior in terms of detection rates (speeds of up to 15 frames/s are reported). Recent surveys on face detection (e.g. in [8]) suggest that face detectors based on Viola and Jones' method have been the most effective to date.

Figure 2 illustrates the implementation of the face detection in a cascade structure. Each node in the structure is essentially a classifier that performs a labeling of the given

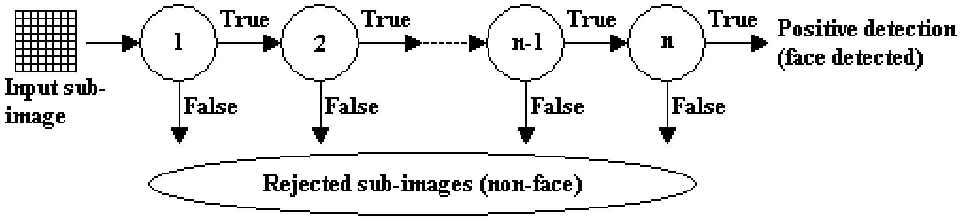

Figure 2: A face detector in a cascade structure.

input to the face class or non-face class, and these individual classifiers are trained with the AdaBoost algorithm. Once one of the nodes rejects an input for not containing a face, the face detection process stops and the next input sub-image is acquired for classification. Needless to say, a sub-image containing a face will be propagated through the whole cascade. This is an implementation of the so-called "coarse-to-fine"search strategy [2], where initial classifiers are simple but flawed, while subsequent classifiers are complex but highly discriminative. The face detection problem is a "rare-event detection problem", which means target samples occur at much lower frequencies than non-targets. Face detectors in a cascade structure will harness the huge speed-ups possible because non-face sub-images which constitute a major percentage of the total input image will be rejected early on without invoking a majority of the classifier nodes, while only rare face sub-images are allowed to progress till the end of the structure.

2.1 Bootstrapping the AdaBoost face detector

Like other face detectors, an AdaBoost face detector will be plagued by false positives if the negative samples used during training do not provide a good representation of the actual distributions of the negative class. It has been mentioned that creating a training set for the negative class is non-trivial since it is not well-defined, as anything that does not look like a face can be included. Until now, a guideline that can be used to efficiently select the best negative samples for the training of a face detector (or general object detectors) cannot be found in the literature. For now, it seems that bootstrapping is the only solution for such a classification problem.

Two approaches can be used to bootstrap the AdaBoost face detector. The first approach is practically a "redo"– collect the false negatives and retrain the whole face detector. This can be intuitively understood by studying Figure 1. However, training a classifier from scratch using the AdaBoost algorithm is computationally costly. It is reported in [5] that the training duration was in the order of weeks. Naturally we seek alternatives to avoid this. It is worth noting that since complete re-training in other face detection algorithms might not be as prohibitively time consuming as re-training in the AdaBoost method, not much attention is paid to this area previously.

The second approach is less inherent. Mindful of the cascade structure of the detector, we can use the false positives to train additional stages of classifiers that will be attached to the end of the original cascade to eliminate these false positives completely. Training a few classifier stages is definitely more appealing than re-training the whole face detector, but this method has a drawback as well. Processing time increases with the addition of classifier stages. Since the negative class is almost infinitely large, there is no telling how many additional stages are needed before false detections are eliminated. The detection speed of the cascade could deteriorate badly if stages are continuously added.

While not immediately apparent in [5], the second bootstrapping method is actually implemented by Viola and Jones. During the course of training, a partially trained cascade is used to generate false positives by scanning images that do not contain faces. The false positive samples are used to train subsequent stages of the cascade. Hence, bootstrapping is considered as part of the training procedure.

3 The proposed bootstrapping approach

In consideration of the difficulties of training the AdaBoost face detector, a new bootstrapping strategy, an enhancement of the second bootstrapping approach, is devised and will be presented in the subsequent sections. Although derived heuristically, the initial results prove that the new algorithm is capable of providing a more efficient means of bootstrapping the AdaBoost face detector.

3.1 The AdaBoost algorithm

In order to understand the principles and motivation behind the new bootstrapping approach, it is crucial to gain a brief insight into the original AdaBoost face detection algorithm proposed by Viola and Jones [5]. The algorithm is reproduced in Table 1. Refer to Viola and Jones's paper for a detailed description of their face detection algorithm.

The AdaBoost face detection algorithm attempts to produce a very accurate face detector by combining rough and moderately inaccurate classifiers. These "weak"classifiers have a performance rate of slightly better than chance (more than 50% correct) and operate on simple 1-dimensional features. Each weak classifier performs a thresholding operation on a scalar feature selected from an overcomplete set of *Haar*-wavelet-like features. During the process of boosting using the AdaBoost algorithm, the weak classifiers are selected sequentially based on their performance– weak classifiers which are stronger relative to other weak classifiers are selected first– to form an ensemble that as a whole exhibit a highly non-linear joint decision function. Each classifier node in the cascade structure is a combination of these weak classifiers.

3.2 A new bootstrapping algorithm

With reference to Figure 2, the classifier nodes in the cascade structure are trained individually using the AdaBoost algorithm. Each node is actually an ensemble of weak classifiers and thus are face detectors in their own right. To exploit the speed advantage

Table 1: The *AdaBoost* algorithm for training a face detector.

1. Given example images $(x_1, y_1), ..., (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.

2. Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives positives respectively.

3. For $t = 1, ..., T$:

   (a) Normalize the weights by
$$w_{t,i} \leftarrow \frac{w_{t,1}}{\sum_{j=1}^{n} w_{t,j}}. \tag{1}$$

   (b) Select the best weak classifier with respect to the weighted error
$$\epsilon_t = \min_{f,p,\theta} \sum_i \omega_{t,i} \mid h(x_i, f, p, \theta) - y_i \mid . \tag{2}$$

   (c) Define $h_t(x) = h(x, f_t, p_t, \theta_t)$ where f_t,p_t and θ_t are the minimizers of ϵ_t.

   (d) Update the weights by
$$\omega_{t+1,i} = \omega_{t,i} \beta_t^{1-e_i}, \tag{3}$$
   where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

4. The final strong classifier is
$$C(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & otherwise. \end{cases} \tag{4}$$
   where $\alpha_t = \log \frac{1}{\beta_t}$.

of the cascade structure, the classification performance of the nodes are tuned to range from weak to strong.

When false detections occur for the AdaBoost face detector, what happens under the hood is that the input sub-images in question successfully slipped through all the weak classifiers included in the whole cascade and appeared at the output of the face detector. In other words, all the weak classifiers classified the samples incorrectly. When this occurs, two conclusions can be made about the individual weak classifiers in the ensemble:

1. The feature type of the weak classifier is not discriminative enough to differentiate the false positives generated from the real positives.

2. The feature type of the weak classifier has enough discriminative power, but the decision boundary of the weak classifier is not optimum causing the negative input to be misclassified.

Figure 3 illustrates the situation above. The false positive samples provide extra information unknown before during training of the cascade. Naturally, we should make use of them to improve our cascade instead of straightforwardly training a new classifier stage to account for them.

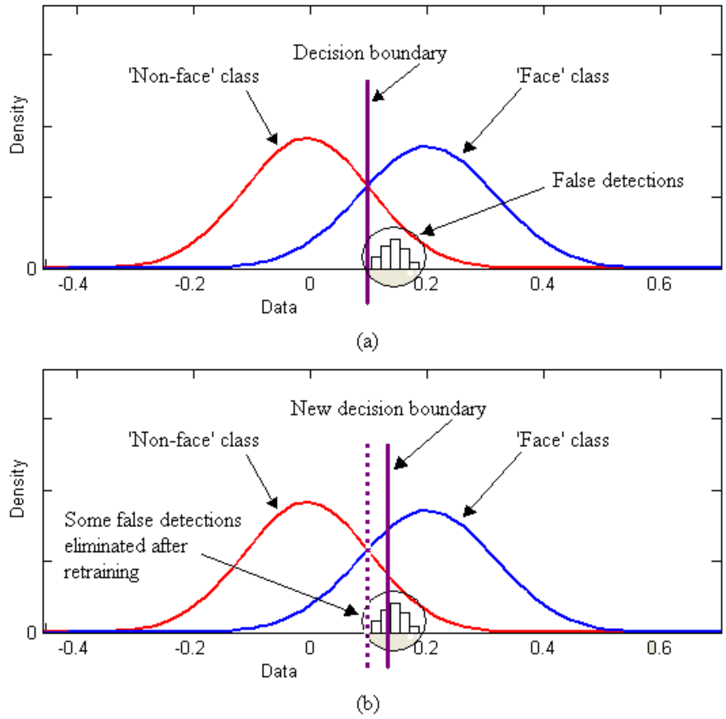

Figure 3: (a) False positives generated due to weak discriminative power of classifier or sub-optimum decision boundary. (b) Shifting of the decision boundary by retraining of the weak classifier with the extra information provided by the false positives.

The AdaBoost algorithm in Table 1 is only a sequential forward selection (SFS) algorithm (as opposed to sequential backward selection (SBS)), hence, there are no provisions for backtracking and removing previously added weak classifiers which are inefficient. The assumption of monotonicity of the error rate must be made in order to utilize the AdaBoost algorithm. Although there are other AdaBoost variants that incorporate SBS (such as the FloatBoost in [7]), the additional steps required for backtracking directly increase the cost of training and hence is unattractive. Since backtracking is not realizable, there is no way of changing the feature types in the weak classifiers added previously.

However, with the new information provided by the false positives, the decision boundary of the weak classifiers can at least be shifted in order to account for them. This is the basis on which the proposed bootstrapping algorithm is built on. The next problem would be what the new decision boundary should be. To achieve the most optimum boundary in view of the additional information, the weak classifiers must first be retrained with the addition of the false positives into the training dataset. If the new decision boundary of a specific classifier results in a lower error rate, then update the decision boundary of the classifier in question. The classifier should then be reweighted with regards to its new error rate. If a higher or equal error rate is achieved, no changes should be made.

The main concept behind the steps described previously is that the weak classifiers should not unlearn what they have been trained previously if the additional information do not result in a better performance. Certainly, the false positives might not be eliminated at all or they might not be eliminated completely since the overall response is the combination of individual results from the weak classifiers. In the former case, the effects of low discriminative power in the feature types cause the false samples inseparable from the face samples, and extra classifier stages are inevitable. In the latter case, even if the bootstrap process only results in a reduction of false positives, a lesser amount of false positives should result in simpler extra classifier stages. The experimental results will prove this point.

Table 2 illustrates the pseudo-code of this bootstrapping algorithm.

4 Some experimental results

In the first experiment, a face detector with a base resolution of 20×20 pixels was trained using the AdaBoost algorithm with a training set of 1521 face sub-images obtained from the *BioID face database* (http://www.humanscan.de/) and 10564 non-face sub-images (obtained from a random crawl on the internet). The resulting face detector was built in a cascade structure.

The face detector was then subjected to 2 sets of images which contain no faces. Both sets contain 50 images in total of average size 400×300 pixels. For the first set of images, the face detector accumulated 258 false positive sub-images. A total of 339 false positives were accumulated for the second set. Using the false positives generated from both sets, the face detector was bootstrapped using the algorithm in Table 2.

The resulting two bootstrapped face detectors were then subjected back to the 2 sets of images. Without time-consuming retraining or additional classifier stages, the face detector registered less false positives. For the first set of 50 images without faces, a total of 127 false positive sub-images were accumulated, a decline of 51%. For the second set, a total of 186 false positive sub-images were accumulated which indicates a decrease of 45%. Figure 4 depicts the design of the experiment.

In the second experiment, the trained face detector was subjected to an image which contains a face (see Figure 5). Due to the fact that the training set of the classifier was less than adequate (only 10564 non-face sub-images were used) and that the input image contained rich details, as many as 80 false positives were accumulated from this image alone. The false positive sub-images and the detector are then subjected to the bootstrapping algorithm.

As expected, the modified face detector generated less false positives– a total of 21 were generated which indicates a reduction of 74%. An additional stage of classifier was then trained on the remaining 21 false positives to prevent them from recurring. In this new stage of the cascade, a total of 15 weak classifiers were needed to achieve 0% false positive rate. As a comparison, a second additional stage of classifier was added to the original face detector as well. The second extra stage was trained on the 80 false

Table 2: The bootstrapping algorithm.

1. Given a previously trained boosted classifier $\mathcal{H} = \{(h_1, \alpha_1), ..., (h_T, \alpha_T)\}$ where α_t is the associated weight for weak classifier h_t with detection rate $D_{\mathcal{H}}$ false positive rate $F_{\mathcal{H}}$.

2. Given the set of training images $\mathcal{S} = \{(x_1, y_1), ..., (x_n, y_n)\}$ where $y_i = 0, 1$ for negative and positive examples respectively.

3. Given the set of collected false positive images $\mathcal{B} = \{(p_1, q_1), ..., (p_n, q_n)\}$ where $q_i = 0, 1$ for negative and positive examples respectively.

4. Initialize weights for both $\mathcal{S}$ and $\mathcal{B}$ samples $\omega_{1,i} = \frac{1}{2(m_1+m_2)}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively and for $q_i = 0, 1$ respectively, where m_1 and m_2 are the number of negative samples of $\mathcal{S}$ and $\mathcal{B}$ respectively, and l is the number of positive samples of $\mathcal{S}$.

5. For $t = 1, ..., T$:

   (a) Normalize the weights by
   $$w_{t,i} \leftarrow \frac{w_{t,1}}{\sum_{j=1}^{n} w_{t,j}}. \tag{5}$$

   (b) Duplicate sets by setting $\mathcal{B}^* = \mathcal{B}$ and $\mathcal{S}^* = \mathcal{S}$.

   (c) For current weak classifier h_t of feature type f_t with threshold θ_t and polarity p_t, $\forall (p, q) \in \mathcal{B}$, if $[q \times h_t(p, f_t, \theta_t, p_t)] > 0$,$\mathcal{B}^* = \mathcal{B}^* - p$ and $\mathcal{S}^* = \mathcal{S}^* + p$.

   (d) Get error rate ϵ_1 of h_t on $\mathcal{S}^*$.

   (e) Retrain h_t with the same feature type f_t on the sets $\mathcal{S}^*$ and $\mathcal{B}^*$ to obtain a new weak classifier h_t^*.

   (f) Get error rate ϵ_2 of h_t^* on $\mathcal{S}^*$ and $\mathcal{S}^*$.

   (g) If $\epsilon_2 < \epsilon_1$, $h_t \leftarrow h_t^*$.

   (h) Get weighted error of current weak classifier by
   $$\epsilon_t = \sum_i \omega_{t,i} \mid h_t(x_i, f_t, p_t, \theta_t) - y_i \mid + \sum_i \omega_{t,i} \mid h_t(p_i, f_t, p_t, \theta_t) - q_i \mid . \tag{6}$$

   (i) Update the weights by
   $$\omega_{t+1,i} = \omega_{t,i} \beta_t^{1-e_i}, \tag{7}$$
   where $e_i = 0$ if example x_i or p_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

   (j) Set new weight for current classifier as $\alpha_t = \log \frac{1}{\beta_t}$.

6. For the modified overall boosted classifier $\mathcal{H}^*$ with detection rate $D_{\mathcal{H}^*}$ and false positive rate $F_{\mathcal{H}^*}$, while $D_{\mathcal{H}^*} < D_{\mathcal{H}}$ or $H_{\mathcal{H}^*} > H_{\mathcal{H}}$, add more weak classifiers with the boosting algorithm.

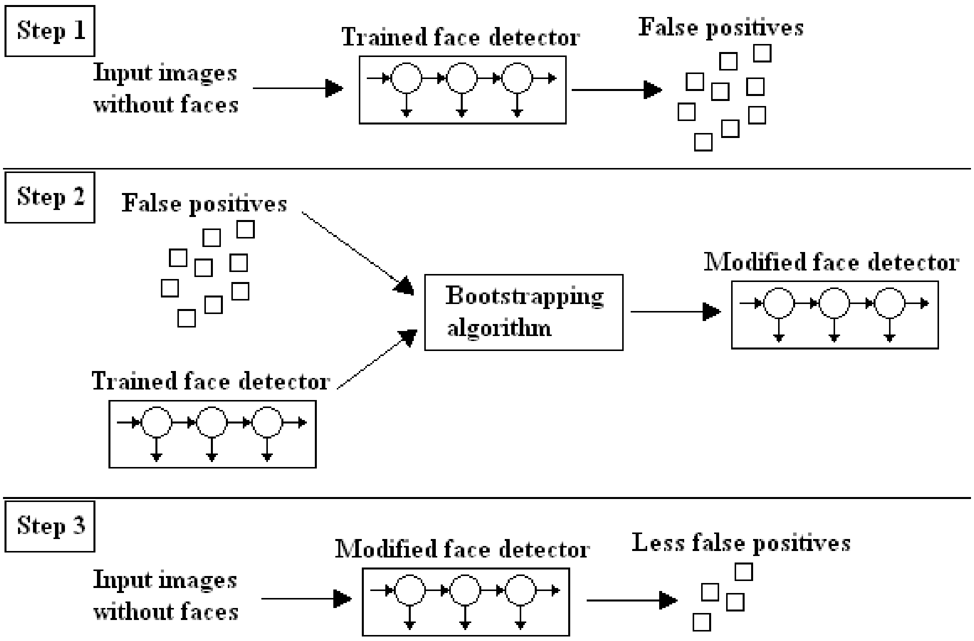

Figure 4: The design of experiment 1.

Figure 5: The input image for the second experiment.

positives and a total of 24 weak classifiers were needed to achieve 0% false positive rate. Figure 6 illustrates the results of the experiment, while Figure 7 depicts the design of the experiment.

5 Discussion and conclusion

The initial results show that the algorithm is capable of reducing false positive instances without additional weak classifiers and classifier stages. While the false positives were not eliminated completely, fewer false positives indicates that fewer weak classifiers will be needed to construct an additional stage using the AdaBoost algorithm for a classifier cascade to account for the generated false positives.

Nevertheless, more empirical evidence of the effectiveness of the algorithm should be

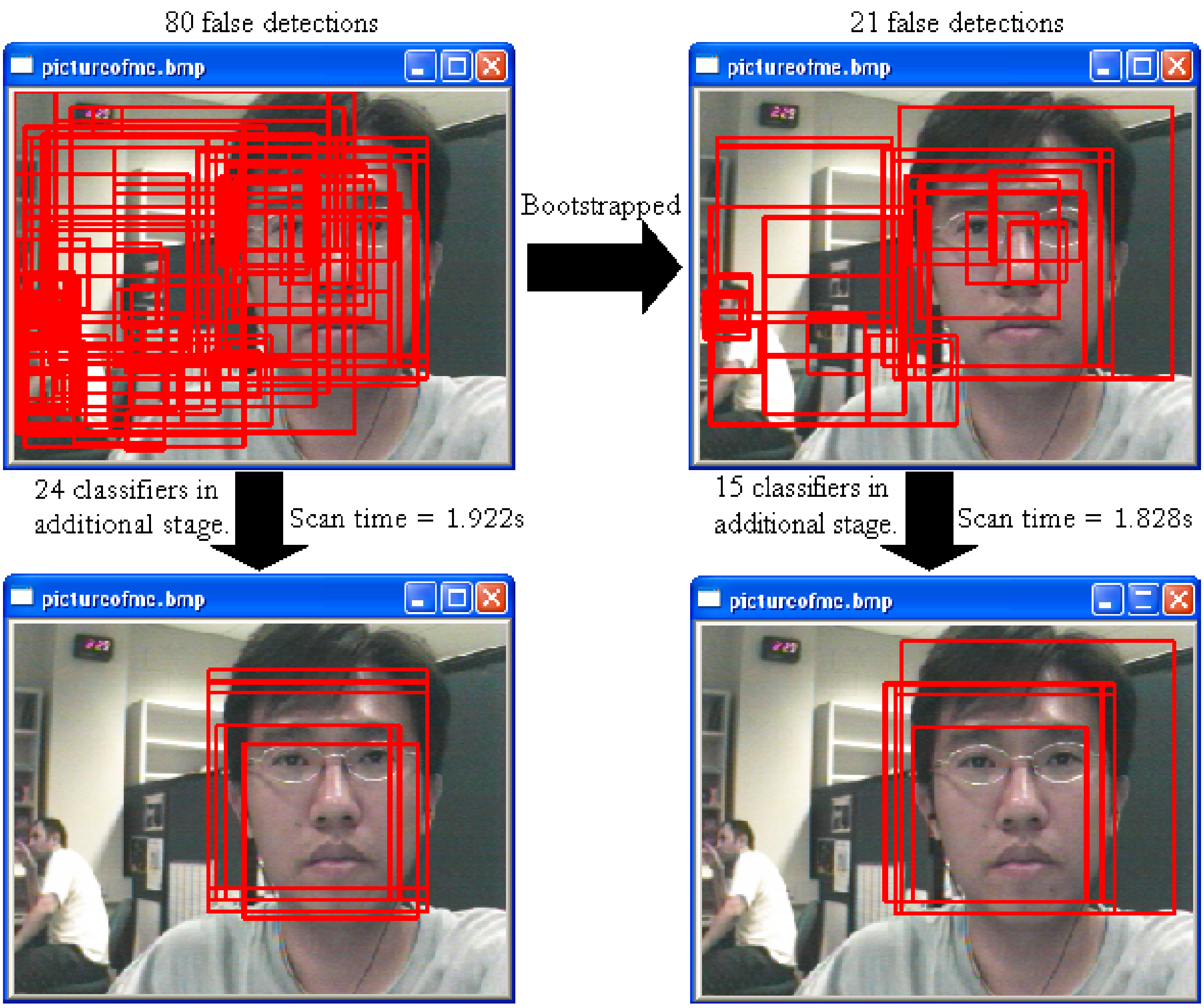

Figure 6: Results of the second experiment.

obtained. The proposed bootstrap algorithm should be tested on face detectors which are trained on more extensive or more standard datasets to allow the results to be comparable with other researches. Since the algorithm was derived heuristically, efforts should be made to explain how the proposed algorithm affects the AdaBoost classifier from the standpoint of learning theory.

To kick start future works, a bigger dataset of training images has been collected. Besides the *BioID face database* used previously, the *CBCL Face Database Number 1* from the *MIT Center for Biological and Computation Learning* (http://www.ai.mit.edu/projects/cbcl) and the *CMU PIE* database [14] have been added to the collection of face images. The current total number of face images stands at 5446 images. The total number of non-face images will be increased as well.

On another note, it is interesting to see if the the algorithm is applicable to other variants of AdaBoost-based face detection approaches. Examples of the variants include the *RealBoost* face detection [12], the *Gentle AdaBoost* face detection [6], the *FloatBoost* face detection [7] and etc.

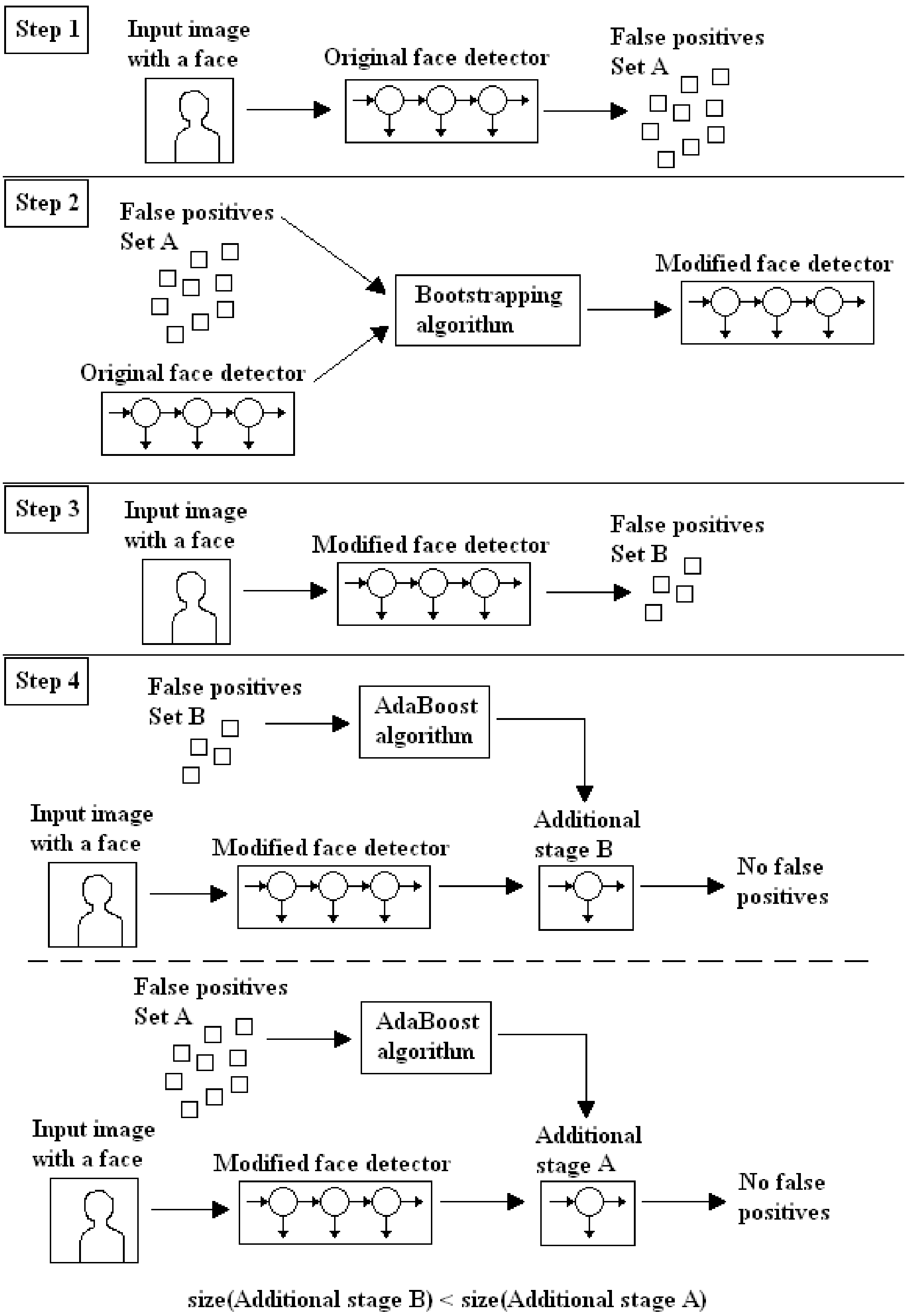

Figure 7: The design of experiment 2.

References

[1] S. Baluja, H.A. Rowley, and T. Kanade. Neural network-based face detection. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.

[2] F. Fleuret and D. Geman. Coarse-to-fine face detection. *International Journal of Computer Vision*, 41:85–107, 2001.

[3] Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, 1995.

[4] Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):325–332, 1997.

[5] M.J. Jones and P. Viola. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[6] A. Kuranov, R. Lienhart, and V. Pisarevsky. Mrl technical report: Empirical analysis of detection cascades of boosted classifiers for rapid object detection. Technical report, Intel Labs, 2002.

[7] S.Z. Li and Z. Zhang. Floatboost learning and statistical face detection. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 26(9):1–12, 2004.

[8] G. Medioni and S.-B. Kang. *Emerging Topics in Computer Vision*, chapter 9. IMSC Press Multimedia Series. Prentice Hall Professional Technical Reference, 2005.

[9] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 23(4):349–361, 2001.

[10] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *IEEE Conference in Computer Vision and Pattern Recognition*, 1997.

[11] D. Roth, M.-H. Yang, and N. Ahuja. A SNoW-based face detector. In *Advances in Neural Information Processing Systems 16*, 1999.

[12] R.E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. In *Proceedings of the Eleventh Annual Conference in Computational Learning Theory*, 1998.

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

[14] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression (PIE) database. In *Proceedings of the 5th International Conference on Automatic Face and Gesture Recognition*, 2002.

[15] K.-K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 20(1):39–51, 1998.