

A comparison of two real-time face detection methods

David Cristinacce and Tim Cootes
Dept. Imaging Science and Biomedical Engineering
University of Manchester, Manchester, M13 9PT, U.K.
david.cristinacce@stud.man.ac.uk

Abstract

Recently two novel approaches to face detection have been devised which produce results comparable to well established methods, but allow near real-time image processing. The first method is due to Viola and Jones [11], the second is due to Fröba and Küllbeck [4]. This paper applies both methods to the FGNET video conference data set. The aim is to extract all the human faces from the video sequence, i.e. approximately 10,000 images, with a minimum number of false positives. The merits of both algorithms are discussed and the results compared.

1. Introduction

The concept of face detection is simple. Given an image, the computer is required to locate each human face in the image and indicate the location of each individual (e.g. by drawing a box around the face). It is desirable that no false detections occur, i.e. an object that is not a human face should not be selected along with the true faces. Therefore face detection is essentially a pattern classification problem. It is however an extreme example of classification as the number of false positives massively outweigh the number of true positives, because most image regions do not contain human faces.

The purest form of face detection uses static grey level images, as this assumes the least amount of information. This attempts to mimic the human ability to easily detect human faces in black and white photographs. Both methods discussed in this paper are of this type. They do not require colour images (to detect human skin tones) or video data to detect movement. Therefore both methods can be used in isolation or alongside cues from video, as discussed in section 4.2.

There have been many different approaches to face detection in grey-scale images (see [7] for a survey). The vast majority are template based, i.e. a varying sized window is passed over the image and all possible sub-regions within the image are evaluated for their likelihood to be human faces. Sub-regions passing a threshold are recorded as faces. Several authors have attempted to solve the problem using well known machine learning techniques. For

example Rowley *et al* [10] applies neural networks to partition the image sub-windows into face/non-face regions. Osuna *et al* [8] applied support vector machines to perform the classification. Roth *et al* [9] use a sparse network of window (SNoW) functions to discriminate faces from background.

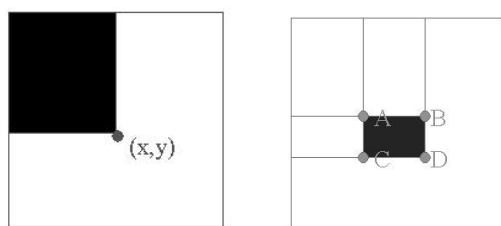
The main drawback of all these methods is speed. There are a large number of possible sub-rectangles within even a small image (say 320x240 pixels). To detect faces at a reasonable range of scales or at more than one possible orientation (i.e. not just vertical faces) a vast number must be evaluated. Therefore these methods can require anything up to a few minutes to analyse a single image. To analyse a large number of images, for example a video sequence, they are impractical.

The two face detection methods discussed in this paper, Viola-Jones Detector [11] and Fröba-Küllbeck Detector [4] are much faster algorithms. At the same time, both methods have reported results comparable to the more well established methods discussed above. They are both fast enough to allow real-time processing of small images on modest computer hardware. A summary of both methods is given below and then both algorithms are applied to the FGNET video data, as described in section 4.2. The overall results presented in section 5.

2. Viola-Jones Detector overview

The Viola-Jones Detector [11] [12] consists of three parts. The first is an efficient method of encoding the image data known as an "integral image". This allows the sum of pixel responses within a given sub-rectangle of an image to be computed quickly and is vital to the speed of the Viola-Jones Detector. The second element is the application of a boosting algorithm known as AdaBoost [3] to select appropriate features that can form a template to model human face variation. The third part is a cascade of templates that allows simple feature sets to quickly discard most of the uninteresting parts of the image and only apply more expensive complex template models to "interesting" regions, that pass the first few stages and are hence more likely to contain faces. This leads to an efficient and accurate face detector.

The integral image is constructed by replacing each image pixel with a value that corresponds to the pixel sum, above and to the left of the pixel, as shown in figure 1(a).



(a) Construction of an integral image

(b) The pixel sum in any region of the original image = $A+D-B-C$

Figure 1: The integral image

An integral image can be constructed in one raster scan of the image. This image structure is also known as an "area sum table" and has been used by [2] for computer graphics applications. The structure is useful, because it allows quick calculation of a pixel sum in any rectangular region, as described in figure 1(b).

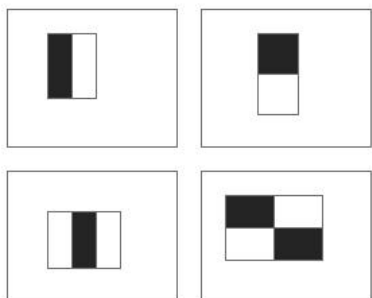


Figure 2: Different feature types

The second stage of the Viola-Jones Detector is the selection of image features to form a template. The initial feature set is chosen such that the response of each individual feature can be calculated easily using an integral image. The different types of feature are shown in figure 2. To form a model template of the human face many of these types of features must be selected and combined. The method used by Viola and Jones is to employ a boosting algorithm known as AdaBoost [3].

AdaBoost is a method of combining many "weak classifiers" into one effective "strong classifier". The individual weak classifiers may only give a very weak discrimination between two populations, e.g. only a little better than ran-

dom. However AdaBoost is able to select the more promising classifiers and by a process of re-weighting the training data, construct an effective classifier consisting of a large set of the appropriately weighted weak classifiers that complement each other and produce better discrimination than any individual weak classifier. The AdaBoost selection scheme is deliberately designed to have no knowledge of the underlying mechanism of each weak classifier, so can be applied to any large set of poorly performing classifiers.

In the Viola-Jones Detector the weak classifiers are based on features like those displayed in figure 2. The training set for each weak classifier is a large set of 24×24 pixel face images (~ 6000 in our implementation) and an equally large set of non-face images. Each feature type is allowed to occupy any sub-rectangle of the 24×24 region. This produces a vast number of possible features that AdaBoost may pick from. The selected features and weights then form the template that is used to scan the image and detect regions that resemble human faces. *

In our implementation the first 8 features chosen by AdaBoost are depicted in figure 3. The first feature exploits the fact that eyes are generally darker regions than the nose and cheeks. The other features have less obvious interpretation, but generally indicate that the outline of the head is modelled by the feature set. Note the first feature found in our implementation is the similar to the first feature selected by Viola and Jones [12], but the second feature is different. This difference is due to variation in the face/non-face training sets used.

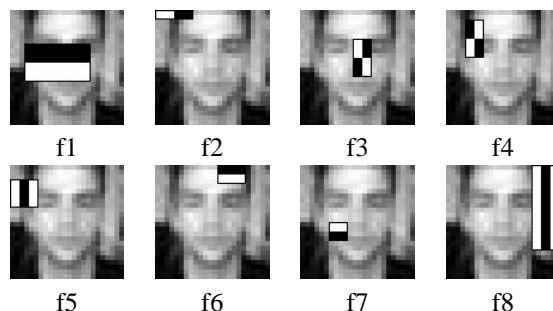


Figure 3: Features selected by AdaBoost, overlaid on an example from the training set

The third part of the Viola-Jones Detector is the use of a cascade of the feature templates described above. A cascade is employed to reduce image processing time by focusing attention on the more interesting regions of the image. For example the flat regions of an image, clearly do not contain faces and can be quickly discarded by use of a template

*The exact formulation of the individual feature classifiers is describe fully in [12]. Also described is a method of normalising the feature response from a given image patch, that is essential to the detection process. This normalisation involves computing a "squared integral image" in addition to the original integral image.

consisting of only a small number of features. The training of the cascade proceeds as follows.

1. Set the level number $i = 0$ and create an empty cascade model C_0 .
2. If $i = 0$ load from disk a set of face examples P_0 and a set of non-face examples N_0 .
3. If $i \neq 0$ form a new non-face examples set N_i by running the incomplete cascade C_{i-1} over a set of images, known not to contain human faces.
4. If the size of set N_i is less than the limit s_t then STOP.
5. Split the faces/non-faces sets P_0 & N_i into train/verification sets $P_{t_0}, P_{v_0}, N_{t_i}$ & N_{v_i} .
6. Build template model L_i from face set P_{t_0} and non-face examples N_{t_i} using n_i features.
7. Calculate a threshold t_i by applying L_i to the verification sets P_{v_0} & N_{v_i} , such that the false rejection rate is fr_i .
8. Create C_{i+1} by adding L_i with threshold t_i to the current cascade C_i .
9. $i = i + 1$, go to 2.

Here the training parameters that need to be set are s_t , n_i and fr_i . s_t is the threshold on the minimum size of the non-face training set N_i , n_i is the number of features used in each level of the cascade and fr_i is the false rejection rate at each level.

In our implementation $s_t = 100$ images, $fr_i = 0.01$ for all levels and n_i was chosen so that the first 16 levels were 10 – 10 – 10 – 20 – 20 – 50 – 50 – 100 – 100 – 100 – 100 – 100 – 200 – 200 – 200 – 200. The number of features n_i and the false rejection rate fr_i at each level are critical to the speed and the performance of the final cascade. However it is difficult to decide these values by anything other than trial and error. In our implementation the false rejection rate fr_i was set to 0.01 at each level i.e. 1.0% of faces may be rejected by each level. This false rejection rate was found to give good results with 16 cascade levels, comparable to the results presented in [12] on the CMU [10] test data.

However, there is no guarantee that other formulations will not provide better classification performance and be more efficient, e.g. by using a smaller number of features in the first level of the cascade. In fact, one of the main drawbacks of the Viola-Jones Detector method is the large amount of training time required to train a model. This extends from the fact that at each stage of the AdaBoost algorithm all possible features must be trained on the entire re-weighted training data. This means on a modern PC (e.g. a 2.0Ghz machine) training the whole cascade requires a few

days of computation time. Therefore testing different formulations of the cascade is a time consuming and difficult task.

The resulting cascaded classifier is however extremely efficient. The cascade described in this paper is capable of searching a 320*240 image in ~ 200 ms using modest hardware (i.e. a 500Mhz PentiumII processor).

Normally the cascade is scanned across an image (at multiple scales) and an image region is declared a face if it passes all levels of the cascade. However, in our final cascade the template response for each level of the cascade is summed and a threshold v_t applied. Only regions that pass the threshold v_t are accepted as face regions. This is similar to the original formulation as the template response from a given level is always positive and the image region must pass all the cascaded levels to obtain a score likely to pass the threshold v_t . The advantage of this scheme is it allows the threshold to be varied to aid in the construction of an FROC curve.

3. Fröba-Küllbeck Detector Overview

In comparison to the Viola-Jones Detector, the Fröba-Küllbeck Detector [4] [5] is much simpler. This method uses orientation maps to search images quickly and efficiently. The first step of the search is to form a multi-scale image pyramid of the original image [1]. Then an orientation map is formed from each pyramid level, using Sobel filter masks. This encodes the local orientation vector at each pixel, see [5] for exact details.

The set of edge strengths $s(x, y)$ and orientations $\theta(x, y)$ form an orientation map or "vector field". An orientation map of an unseen image can then be searched using an orientation map of an average face. If the similarity between the average face map and a sub-region of the image is small enough, then the region is accepted as face. This method is therefore very simple, but does have some inherent reliability due to the stability of orientation maps in general. i.e The orientation map of a image is stable under linear transformations of the underlying pixel values e.g. due to addition by a constant or scaling by a constant.

The Fröba-Küllbeck Detector is further speeded-up and improved by the addition of extra thresholds. Firstly, only the strong edges are utilised by the algorithm, i.e a threshold s_t is applied to all orientation vectors if $s(x, y) < s_t$ then the orientation vector is ignored at point (x, y) . Hence weak edges, in both the image orientation map and the template map are ignored, effectively weak edges are considered to be noise.

More formally, the distance metric between any two orientation vectors $v_1 = (s_1, \theta_1)$ and $v_2 = (s_2, \theta_2)$, defined by polar co-ordinates, is shown in equation 1.

$$d(v_1, v_2) = \begin{cases} \sin(\theta_1 - \theta_2) & \text{if } |s_1|, |s_2| > s_t \\ 1 & \text{else} \end{cases} \quad (1)$$

Hence the distance measure $d(v_1, v_2)$ between two orientation vectors with sufficient edge strength is just the sine of the angle between the two vectors. The polarity of edges in images being searched are therefore ignored because vectors that differ by 180° are considered equal.

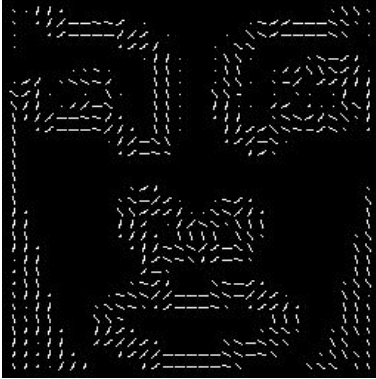


Figure 4: Orientation template

In our implementation the face template constructed is shown in figure 4. Here, only the orientation vectors with $s(x, y) > s_t$ are shown, because they are the only orientations used by the template when searching an image. The template response is the sum of distance measures $d(v_1, v_2)$ between the template and image region.

The number of computer operations required to search an image linearly increases with the number of orientation vectors stored in the template, so the total number of vectors need to be kept to a minimum. Our implementations uses 453 orientations in a 31 by 31 template.

A second speed-up of the Fröba-Küllbeck Detector method is to only search each pyramid level on a coarse scale initially and then perform a refined search if a strong enough match is found. In practice this scheme involves the following search algorithm.

1. Search every 6th pixel position. Get template response r_1
2. If $r_1 < T_1$ then test all 8 possible positions that are 3 pixel positions away. Get template response r_2 , for each location.
3. If $r_2 < T_2$ then test all 8 possible positions that are 1 pixel position away. Get template response r_3 , for each location.
4. if $r_3 < T_3$ then accept as a face candidate.

Note searching every 6th position instead of every possible position in the image reduces the computation time by a factor of approximately 2.8% ($= (1/6) * (1/6)$). But note that this scheme can result in the same candidate being returned more than once! To enable efficient computation appropriate thresholds values $T_1, T_2 \& T_3$ are essential. A large value of T_1 produces a very slow detector!

This coarse to fine method takes advantage of the fact that a face template often matches at many locations around and near the face, a phenomenon noted by several researcher e.g. [10][11]. However, the three thresholds $T_1, T_2 \& T_3$ can only be selected by empirical testing on a validation set and may not be optimal for all possible images. Our implementation gives results similar to those presented in [4] and searches $320 * 240$ pixel images in ~ 400 ms. However, the speed of the detector could probably be improved by further optimisation.

4. Experiments

The main focus of this paper is the empirical evaluation of both the Viola-Jones Detector and the Fröba-Küllbeck Detector for the purpose of extracting face images from a large set of video images. The exact set of images used is described in section 4.1, the methodology used to analyse the search results described in section 4.2 and a simple background differencing scheme described in section 4.3.

4.1 FGNET Data

The FGNET video data consists of three static cameras placed in a conference room. The first camera (cam1) records the view across the desk where the participants in the meeting are sitting. The second camera (cam2) records the opposing view from the other side of the desk. The third camera (cam3) records a 360° view of the whole room from the centre of the desk. The face images in this third camera are highly distorted, so cam3 is not used in this paper.

The total number of images (ie frames of the video) across both cameras and the scenarios which have labelled data, namely A,B & D, is over 10,000 images. Whilst it is feasible to search every one of these images using either the Viola-Jones Detector or Fröba-Küllbeck Detector, it is infeasible to analyse this number of images with a variety of different parameter settings. Each different set of parameters requires the whole data set to be searched, therefore a restricted data set has been obtained, by including only images where the frame number is divisible by 100. This produces three test sets as shown in table 1.

[†]Note frames 21900-22300 are excluded from the "cam1" image set, because not all faces are marked in these images. This appears to be due to an error that occurred in the frame numbering during marking up.

Camera	No. Images	No. Faces
cam1†	193	459
cam2	212	536
cam1+cam2	405	995

Table 1: The FGNET video data subset of scenarios A,B & D

4.2 Calculating the FROC graphs

To analyse the data sets, i.e. cam1 and cam2, all images are searched and a set of candidate regions returned by the face detector. The regions that are close enough to a true face in the image are declared true positives. All other regions are declared false positives. Each possible configuration of a face detector can then be used to plot a point on the FROC curve, showing the trade off between the proportion of faces found (i.e. true positive rate) and the number of false positives return across the data set. The FROC curves produced are shown in figure 8 and figure 9.

There are some refinements to the list of candidates returned by the detector. For example, candidates that are close together are merged. This reflects the fact that both the Viola-Jones Detector and Fröba-Küllbeck Detector produce multiple candidates around a face region, i.e. the face is detected multiple times. This behaviour is also true of some false positive regions. Therefore to produce one candidate per face the candidates in close proximity are merged. The distance metric between any two candidates is described in figure 5.

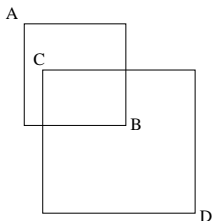


Figure 5: The distance metric between a candidate defined by corners (A,B) and a candidate defined by corners (C,D) $d_{cands} = (d_{AB}^2 + d_{CD}^2) \div (d_{AC}^2 + d_{BD}^2)$, where d_{AB} is the distance between points A & B.

Two candidates are merged if $d_{cands} < 0.3$. The two candidates are then merged by averaging the top left corners $A&C$ and the bottom right corners $B&D$ to form a new candidate. Note the ordering of the merging process affects the final result slightly if more than two candidates overlap within a region. However, unless there are an unusually large number of candidates the effect is not significant.

The set of merged candidates is then compared with the hand labelled ground truth data. The eye points are labelled for each approximately frontal face in the data set, i.e. all faces where both eyes are visible. The face candidate regions cannot be directly compared with the labelled

eye points. Therefore the average eye point locations within the template region are learnt during training. The distance metric between predicted candidate eye points and true eye points is described in figure 6.

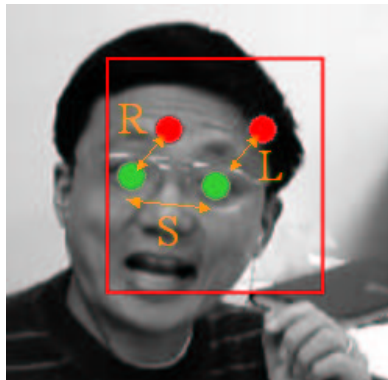


Figure 6: The distance between the eye points predicted by the candidate region and the true eye points $d_{eyes} = \frac{R+L}{2S}$

When constructing the FROC graph a candidate is considered to be a true positive if $d_{eyes} < 0.5$ for one pair of labelled eye points. A candidate is considered a false positive if $d_{eyes} > 1.0$ and ambiguous if $0.5 < d_{eyes} < 1.0$ when it is ignored by the FROC curve. The true positive threshold of 0.5 is quite high because when the face is turned to the side the actual eye separation S is reduced, increasing the d_{eyes} metric, for non-frontal faces.

The points on the FROC curve are determined by the configuration of the Viola-Jones Detector and Fröba-Küllbeck Detector. The parameters chosen affect the trade off between finding as many true positives as possible and minimising the number of false positives. The parameters for the two detectors are varied as follows.

- Viola-Jones Detector - The threshold v_t is varied and each different value defines a different point on the FROC curve, all other parameters are kept constant.
- Fröba-Küllbeck Detector - A constant k is added to each threshold T_1, T_2 & T_3 . Each value of k defines a different point on the FROC curve, all other parameters are kept constant.

4.3 Background Differencing

The FGNET data is collected in a constrained environment. It is therefore possible to use image differencing to exploit the fact that the background is constant. The first few frames from scenarios A,B & D are used to create separate average background images for both cam1 and cam2. These first few frames don't contain any humans, however there is some movement of the chairs between scenarios and the viewing angle of cam1 is shifted down slightly in Scenario

D relative to scenarios A&B. Typical images are shown in figure 7.

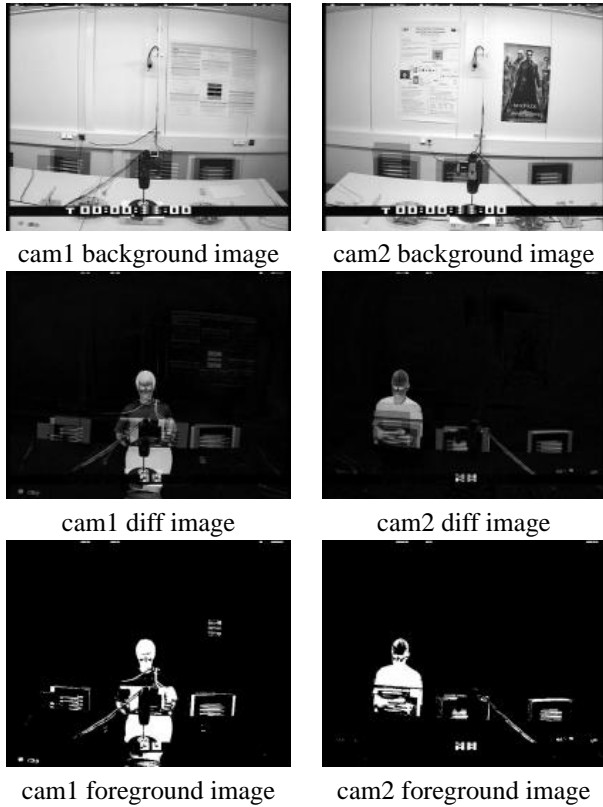


Figure 7: Results of difference imaging on cam1 and cam2 datasets

The difference image is the absolute difference in pixel values between each frame and the background image. The foreground image is computed by classifying pixel differences greater than 60 as foreground and all other pixels as background, here the pixel value range is 0-255. As the images in the third row of figure 7 show, the difference imaging does exclude a large amount of the background (shown in black) and enables candidates to be excluded that occur outside of the foreground region (shown in white). This should help to reduce the number of false positives that are detected by both the Viola-Jones Detector and Fröba-Küllbeck Detector .

It may also be possible to do some pre-selection of likely looking "blob" shapes to further speed up processing time. However in this paper the foreground images are merely used to prune candidates after a normal search has taken place. The method is to calculate an integral image of the foreground image, where a white foreground pixel has value 1 and a black background pixel has value 0. The proportion of foreground to background p within a candidate region can then be computed efficiently with 4 summations, as shown in figure 1(b). A face candidate is only accepted

if $p > p_t$ for some threshold p_t .

The value of p_t is slightly different for the Viola-Jones Detector and Fröba-Küllbeck Detector methods. The thresholds used are as follows.

- Viola-Jones Detector - $p_t = 0.4$
- Fröba-Küllbeck Detector - $p_t = 0.6$

These values differ to take account of the fact that the Viola-Jones Detector models the background as part of the face template, see figure 3, so requires a threshold that allows more background to occur within the candidate region. The Fröba-Küllbeck Detector only models the interior of the face, see the orientation map depicted in figure 4, so a higher threshold is required.

5. Results

The FROC curves obtained by applying both the Viola-Jones Detector and the Fröba-Küllbeck Detector to the combined cam1 & cam2 dataset are shown in figure 8.

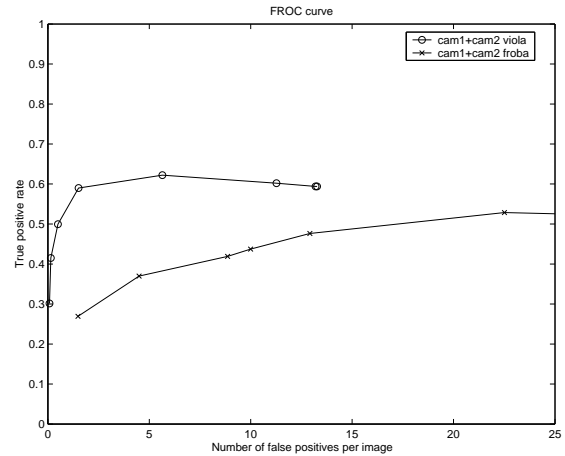


Figure 8: Face detection results *without* using background differencing on the cam1+cam2 data set

Without using background differencing the Viola-Jones Detector clearly outperforms the Fröba-Küllbeck Detector . The Viola-Jones Detector can detect 50.0% of faces and only produces on average 0.50 false positives per image, see the 3rd FROC point in table 2(a). To find 51.5% of the faces with the Fröba-Küllbeck Detector produces around 22.5 false positives per image, a significantly worse result, see the 6th FROC point in table 2(b) or both points plotted in figure 8.

However, the detection performance of the Viola-Jones Detector is still poor, detecting only 50.0% of the 995 faces and producing 202 false positives over the 405 images in the test set. The number of detections can be increased, but only at the expense of further increasing the number of false positives , for example 62.2% of true positives costs 2287 false positives , i.e. 5.65 false positives per images.

Pt	TPR	FPs	Pt	TPR	FPs
1	0.3015	0.0815	1	0.2693	1.4864
2	0.4151	0.1457	2	0.3698	4.4963
3	0.4995	0.4988	3	0.4191	8.8593
4	0.5899	1.5136	4	0.4372	9.9975
5	0.6221	5.6469	5	0.4764	12.9185
6	0.6020	11.2691	6	0.5286	22.5086
7	0.5940	13.2000	7	0.5116	36.8568
8	0.5940	13.2741	8	0.4090	52.8346

(a) Viola-Jones Detector

(b) Fröba-Küllbeck Detector

Pt	TPR	FPs	Pt	TPR	FPs
1	0.2985	0.0667	1	0.2412	0.0543
2	0.4131	0.0988	2	0.3296	0.0889
3	0.4965	0.2988	3	0.3869	0.1309
4	0.5849	0.7284	4	0.4030	0.1630
5	0.6201	1.2765	5	0.4503	0.3235
6	0.6000	1.8988	6	0.5146	0.8543
7	0.5940	2.0914	7	0.5216	2.0790
8	0.5940	2.1012	8	0.4221	3.5556

(a) Viola-Jones Detector

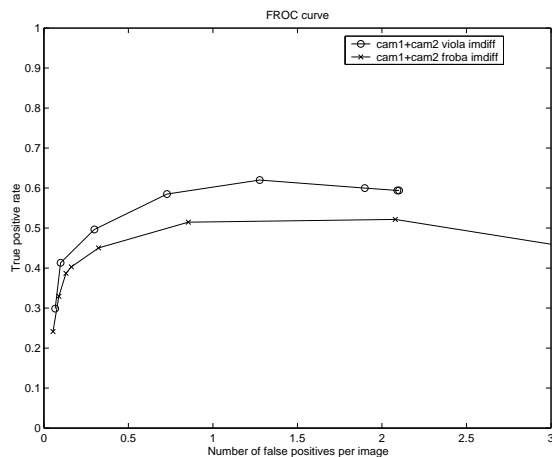
(b) Fröba-Küllbeck Detector

Table 2: FROC points *without* using image differencing on cam1+cam2 dataset

The situation is worse for the Fröba-Küllbeck Detector, see table 2(b).

Note, in figure 8, the true positive rate increases as the number of false positives increases. However, after FROC point 5, the true positive rate starts to decrease. This is due to the merging of candidates. If the threshold of either detector is set too high then too many spurious false candidates are produced and many true candidates are "dissolved" in the merging process. This produces a slight decrease in the true positive rate.

The results can be improved by increasing the threshold of both detectors and using background differencing to restrict the number false positives, as described in section 4.3. The new FROC curve using image differencing is shown in figure 9 and table 3.

Figure 9: Face detection results *using* background differencing on the cam1+cam2 dataset

When using image differencing the Viola-Jones Detector still outperforms the Fröba-Küllbeck Detector. The Viola-Jones Detector can now detect 50.0% of faces at a cost of

Table 3: FROC curve points using image differencing

0.30 false positives per image, see the 3rd FROC point in table 3(a), compared to 0.50 false positives per image without using background differencing. The performance of the Fröba-Küllbeck Detector is worse than the Viola-Jones Detector, i.e. an average of 0.85 false positives are found per image, at a detection rate of 51.5%, see the 6th FROC point in table 3(b).

One interesting point to note is that image differencing has a dramatic effect on reducing the number of false positives that occur with the Fröba-Küllbeck Detector (e.g. 22.5 \rightarrow 0.85 with the 6th FROC point), but a more modest effect when searching with the Viola-Jones Detector (e.g. 0.5 \rightarrow 0.3 with the 3rd FROC point).

This is mainly because the Fröba-Küllbeck Detector is an inferior detector and produces many spurious false positives, when forced to obtain the same detection rate as the Viola-Jones Detector, therefore image differencing removes a large number of false positives and improves results significantly. The other factor is that the Fröba-Küllbeck Detector is particularly prone to detecting certain features in the background. The Fröba-Küllbeck Detector often falsely detects regions with horizontal edge structure as faces. This is because the face template (see figure 4) and faces in general, contain horizontal structure, eg. eyes, eyebrows and mouth. For example in the background image for cam1, shown in figure 7, the centre of the poster on the back wall and the grills on the backs of the chairs often result in false positives. Note that the minimum scale of both detector methods is such that the face of Keanu Reeves in the Matrix poster, shown in cam2 of figure 7 is never detected.

The small improvement of the Viola-Jones Detector when applying background differencing reflects the fact that the Viola-Jones Detector is rarely distracted by the background anyway. Most of the interesting image structure occurs within the foreground region indicated by the differencing image process, i.e. the faces, clothes and desk clutter shown in figure 7. However, using the background

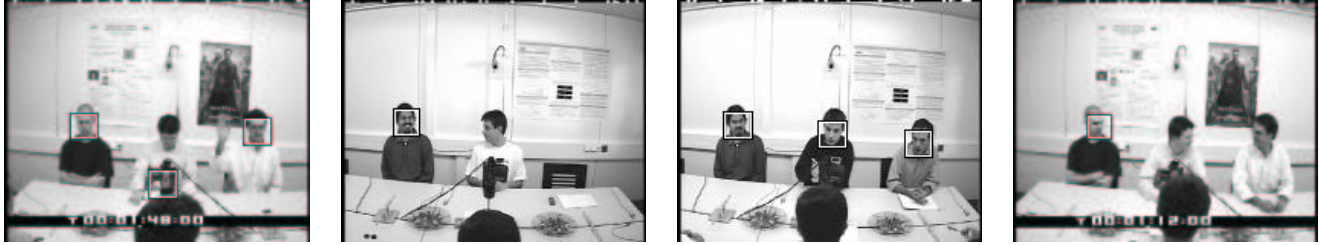


Figure 10: Some example searches using the Viola-Jones Detector and background subtraction

differencing does allow the threshold to be raised, so that more faces are detected for a reasonable number of false positives. For example, the threshold used for FROC point 5 in table 3(a) results in a detection rate of 62.0% faces, at a cost of 1.28 false positives per image, without image differencing the cost would be 5.65 false positives per image.

Some typical search results when using the Viola-Jones Detector method and applying background differencing are shown in figure 10. Faces are generally detected when the subject is looking directly at the camera, but are missed more often when an individual is looking down or away from the camera. Also a small number of false positives occur when items of clothing, folds in a jumper, items on the desk, or hands, are mistakenly classified as faces.

6. Summary and Conclusions

The maximum possible detection rate, using the Viola-Jones Detector method is 62.0% of the 995 faces distributed across 405 images. This is perhaps a little disappointing, but partly reflects the fact that both detectors are only designed to detect frontal faces and in many frames of the video sequence the faces are at oblique angles. In the FGNET database a face is defined as "frontal" if both eyes are visible. However this allows more variation than is exhibited by the Viola-Jones Detector training set and certainly more variation than the Fröba-Küllbeck Detector training set, which constructs the template from an average frontal face. Therefore, viewed in this context, the detection rate is reasonable. Using Viola-Jones Detector and background differencing results in 617 correct face detections being returned and 517 false positives. A tighter threshold could return 494 faces and 121 false positives. Therefore a subset of faces can be extracted from the video sequence for a modest number of false positives.

The main conclusion of this paper is that the Viola-Jones Detector is superior to the Fröba-Küllbeck Detector method. The difference in performance is marginal when using background differencing to discard non-foreground candidates, see figure 9, but the Viola-Jones Detector method is clearly better when image differencing is not used, see figure 8. The Viola-Jones Detector gives better results as it is a more sophisticated non-linear classifier, whilst the Fröba-Küllbeck Detector relies on a rigid template to

match to image regions and uses a simple sum of differences metric. However, the Fröba-Küllbeck Detector is still a fast method, so can be used as a pre-filter before applying more expensive methods, for example [6], which uses the SNoW method [9] to verify the face candidates found by the Fröba-Küllbeck Detector. The Viola-Jones Detector is robust enough to work well on its own.

References

- [1] C.H. Ballard and C.M. Brown. *Computer Vision*. Prentice-Hall, 1982.
- [2] F. C. Crow. Summed-area tables for texture mapping. In *Proceedings of SIGGRAPH '84*, pages 207–212, 1984.
- [3] Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *2nd European Conference on Computational Learning Theory*, 1995.
- [4] B. Fröba and C. Küllbeck. Orientation template matching for face localization in complex visual scenes. In *International Conference on Image Processing ICIP2000*, pages 251–254, 2000.
- [5] B. Fröba and C. Küllbeck. Face detection and tracking using edge orientation information. In *SPIE Visual Communications and Image Processing*, pages 583–594, 2001.
- [6] B. Fröba and C. Küllbeck. Robust face detection at video frame rate based on edge orientation features. In *5th International Conference on Automatic Face and Gesture Recognition 2002*, pages 342–347, 2002.
- [7] E. Hjelmas and B.Low. Face detection: A survey. *Computer Vision and Image Understanding*, 83:235–274, 2001.
- [8] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. In *Computer Vision and Pattern Recognition Conference 1997*, 1997.
- [9] D. Roth, M.Yang, and N.Ahuja. A snowbased face detector. (12), 2000.
- [10] H.A. Rowley, S.Baluja, and T. Kanade. Neural network-based face detection. 20(1):23–38, 1998.
- [11] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition Conference 2001*, volume 1, pages 511–518, Kauai, Hawaii, 2001.
- [12] Paul Viola and Michael Jones. Robust real-time object detection. *International Journal of Computer Vision - to appear*, 2002.