# Face and Eye Detection on Hard Datasets

Jon Parris[1], Michael Wilber[1], Brian Heflin[2], Ham Rara[3], Ahmed El-barkouky[3], Aly Farag[3], Javier Movellan[4], Anonymous[5], Modesto Castrilón-Santana[6], Javier Lorenzo-Navarro[6], Mohammad Nayeem Teli[7], Sébastien Marcel[8], Cosmin Atanasoaei[8], and T.E. Boult[1,2]

[1] Vision and Technology Lab, UCCS, Colorado Springs, CO, 80918, USA
{jparris,mwilber}@vast.uccs.edu
[2] Securics Inc, Colorado Springs, CO, 80918, USA
[3] CVIP Laboratory, University of Louisville, KY, 40292, USA
[4] Machine Perception Laboratory, University of California San Diego, CA, 92093, USA
[5] A Commercial Submission from DEALTE, Saulėtekio al. 15, LT-10224 Vilnius, Lithuania
[6] SIANI, Universidad de Las Palmas de Gran Canaria, 35001, España
[7] Computer Vision Group, Colorado State University, Fort Collins, CO, 80523 USA
[8] Idiap Research Institute, Marconi 19, Martigny, Switzerland

## Abstract

*Face and eye detection algorithms are deployed in a wide variety of applications. Unfortunately, there has been no quantitative comparison of how these detectors perform under difficult circumstances. We created a dataset of low light and long distance images which possess some of the problems encountered by face and eye detectors solving real world problems. The dataset we created is composed of re-imaged images (photohead) and semi-synthetic heads imaged under varying conditions of low light, atmospheric blur, and distances of 3m, 50m, 80m, and 200m.*

*This paper analyzes the detection and localization performance of the participating face and eye algorithms compared with the Viola Jones detector and four leading commercial face detectors. Performance is characterized under the different conditions and parameterized by per-image brightness and contrast. In localization accuracy for eyes, the groups/companies focusing on long-range face detection outperform leading commercial applications.*

## 1 Introduction

Over the last several decades, face/eye detection has changed from being solely a topic for research to being commonplace in cheap point-and-shoot cameras. While this may lead one to believe that face detection is a solved problem, it is solved only for easy settings. Detection/localization in difficult settings is still an active field of research. Most researchers use controlled datasets such as FERET[14] and PIE[11], which are captured under controlled lighting and blur conditions. While these datasets are useful in the creation and testing of detectors, they give little indication of how these detectors will perform in difficult or uncontrolled circumstances.

In ongoing projects at UCCS and Securics addressing long-range and low-light biometrics, we found there were significant opportunities for improvement in the problems of face detection and localization. Face detection is just the first phase of a recognition pipeline and most recognition algorithms need to locate features, the most common being eyes. Until now, there has not been a quantitative comparison of how well eye detectors perform under difficult circumstances. This work created a dataset of low light and long distance images which possess some of the problems face detectors encounter in difficult circumstances. By challenging the community in this way, we have helped identify state-of-the-art algorithms suitable for real-world face and eye detection and localization and we show directions where future work is needed.

This paper discusses twelve algorithms. Participants include the Correlation-based Eye Detection algorithm (CBED), a submission from DEALTE, the Multi-Block Modified Census Transform algorithm (MBMCT), the Minimum Output Sum of Squared Error algorithm (MOSSE), the Robust Score Fusion-based Face Detection algorithm (RSFFD), SIANI, and a contribution from UCSD MPLab. In addition, we compare four leading commercial algorithms along with the Viola Jones implementation from OpenCV 2.1. In Table 1, algorithms are listed in alphabetical order with participants on the top section and our own contributions in the bottom.

## 2 Background

While many toolkits, datasets, and evaluation metrics exist for evaluating face recognition and identification systems, [14, 1] these are not designed for evaluating simple face/eye detection/localization measures. Overall there has been lit-

tle focus on difficult detection/localization, despite the obvious fact that a face not detected is a face not recognized – multiple papers show that eye localization has a significant impact on recognition rates [10, 7].

The Conference on Intelligent Systems Design and Applications [8] performed a face detection competition with two contestants in 2010. Their datasets included a law enforcement mugshot set of 845 images, controlled digital camera captures, uncontrolled captures, and a "tiny face" set intended to mimic captures from surveillance cameras. All except the mugshot database had generally good quality. In their conclusions, they state "Obviously, the biggest improvement opportunity lies in the surveillance area with tiny faces."

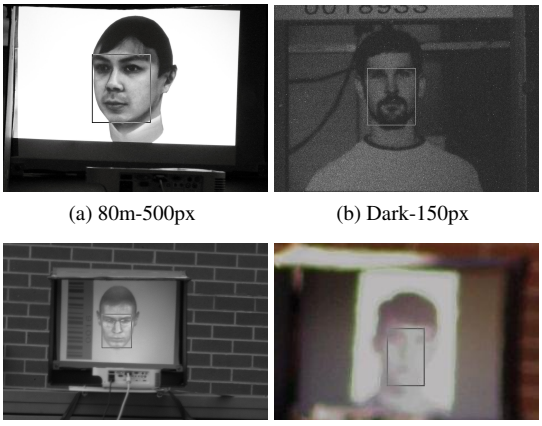
There have been a few good papers evaluating face detectors. For example, [35] uses a subset of data from LFW, and also considered elliptical models of the ideal face location. However, LFW is a dataset collected using automated face detection with refinement. Similarly, [3] leverages parts of existing data and spends much of their discussion about what is an ideal face model. The data in these is presented as being somewhat challenging but still most tested detectors did well. We note, however, that evaluating face detectors against an ideal model is not very appropriate, and in this paper we evaluate detectors with a much more accepting model of a detection – we consider a detection correct if the reported model overlaps the ground truth.

Many descriptions of face detection algorithms include a small evaluation of their performance, but they often evaluate only the effects of different changes within that algorithm. [37, 28] Comparisons to others are usually done in the context of proving that the discussed algorithm is better than the state-of-the-art. Because of the inconsistent metrics used, it is often impossible to compare the results of these kinds of evaluations across papers.

The results of this competition show that there is room for improvement on larger, blurry, and dark faces, and especially so for smaller faces.

# 3 Dataset

We set out to create a dataset which would highlight some of the problems presented by somewhat realistic but difficult detection/localization scenarios. To do this, we created four sub-sets, each of which presents a different scenario in order to isolate how a detector performs on specific challenges. Our naming scheme generally follows *scenario-width*, where *scenario* is the capture conditions or distance and *width* is the approximate width of the face in pixels. Note that width alone is a very weak proxy for resolution and many of the images have significant blur within resulting in effective resolution sometimes being much lower. The experiments use the photohead approach for semi-synthetic data discussed in [4, 5] allowing control over the conditions and including many faces and poses.



(a) 80m-500px (b) Dark-150px

(c) 200m-300px (d) 200m-50px

Figure 1: Cropped samples from the dataset

## 3.1 80m-500px

The highest quality images, the 80m-500px sub-set, were obtained by imaging semi-synthetic head models generated from PIE. They are displayed on a projector and imaged at 80 meters indoors using a Canon 5D mark II with a Sigma EX 800mm lens; see Figure 1a. This camera lens combination produced a controlled mid-distance dataset with minimal atmospherics and provides a useful base line for the long distance sub-sets.

## 3.2 200m-300px

For the second sub-set, 200m-300px, we imaged the semi-synthetic PIE models, this time from 200 meters outside; see Figure 1c. We used a Canon 5D mark II with a Sigma EX 800mm lens with an added a Canon EFF 2x II Extender, resulting in an effective 1600mm lens. The captured faces suffered varying degrees of atmospheric blur.

## 3.3 200m-50px

For the third sub-set, we re-imaged FERET from 200 meters; see Figure 1d for a zoomed sample. We used a Canon 5D mark II with a Sigma EX 800mm lens. The resulting faces were approximately 50 pixels wide and suffered atmospheric blur and loss of contrast. We chose a subset of these images, first filtered such that our configuration of Viola Jones correctly detected the face in 40% of the images. We further filtered by hand-picking only images that contained discernible detail around the eyes, nose, and mouth.

## 3.4 Dark-150px

For the final sub-set, we captured displayed images (not models) from PIE[11] at close range, approximately 3m, in a low light environment, with an example in Figure 1b. We captured this set with a Salvador (now FLIR) EMCCD camera. While the Salvador can operate in extremely low light conditions, it produces a low resolution and high noise image. The noise and low resolution create challenging faces that simulate long-range low-light conditions.

### 3.5 Non-Face Images

To evaluate algorithm performance when given non-face images, we included a proportional number of images that did not contain faces. When evaluating the result, we also considered the false positives and true rejects of images in this non-face dataset. The "non-faces" were almost all natural scenes obtained from the web – most were very easily distinguished from faces.

### 3.6 Dataset Composition

Given these datasets, we randomly selected 50 images of each subset to create 4 labeled training datasets. The training sets also included the groundtruth for the face bounding box and eye coordinates. The purpose of this set was *not* to provide a dataset to train new algorithms; 50 images is far too few for that. Instead, it allowed the participants to internally validate that their algorithm could process the images and the protocol with some reasonable parameter selection.

For testing, we randomly selected 200 images of each subset to create the four testing sets. The location of the face within the image was randomized. An equal number of non-face images was added, and the order of images was then randomized.

## 4 Baseline Algorithms

Detailed descriptions of the contributors' algorithms are presented as appendices A through G.

We also benchmarked the standard Viola Jones Haar Classifier (hereafter VJ-OCV2.1), compiled with OpenCV 2.1 using the *frontalface_alt2* cascade, a scale of 1.1, 4 minimum neighbors, $20 \times 20$ minimum feature size, and canny edge detection enabled. These parameters were chosen by running a number of instances with varying parameters on training data. The choice was made to let Viola Jones have a high false positive rate with a correspondingly higher true positive rate. This choice was made due to the difficult nature of the dataset. Algorithms such as CBED use similar Viola Jones parameters. These parameters typically yield high performance in many scenarios[28].

For completeness, we compared the algorithms' performance against four leading commercial algorithms. Two of these (*"Commercial A (2005)"* and *"Commercial A (2011)"*) are versions from the same company from six years apart. Commercial A (2011) was also one of the best performers in [3].

We aimed to detect both face bounding boxes and eye coordinates. Because Commercial B only detects eye coordinates, we generate bounding boxes by using the ratios described in `csuPreprocessnormalize.c`, part of the *CSU Face Evaluation and Identification Toolkit* [1]. Similarly, we define a baseline VJ-based eye localization using the above Viola Jones face detector. Eyes are predefined ratios away from the midpoint of the bounding box along the X and Y axes. These ratios were the average of the groundtruth of the training data released to participants.

## 5 Evaluation metrics

We judged the contestants based on detection and localization of faces and the localization accuracy of eyes. To gather metrics, we compared each contestant's results with hand-created groundtruth.

For faces, we initially considered using a accuracy measure but found that these systems all have different face models and any face localization/size measurement would be highly biased. Thus our face detection evaluation metrics are comparatively straightforward. In Table 1, a contestant's bounding box is counted as a false positive if it does not overlap the groundtruth at all. Because all of the datasets (modulo the non-face set) have a face in each image, all images where the contestant reported no bounding box count as false rejects. Because some algorithms reported many false positives per image on the 200m-50px set, Table 1 lists the number of images which contain an incorrect box as column FP′ for this set. In the non-face set, only true rejects and false positives are relevant because those images contain no faces.

For these systems, eye detection rate is equal to face detection rate and is not reported separately. For eyes, localization is the critical measure. We associate a localization error score defined as the Euclidean distance between each groundtruth eye and the identified eye position. To present these scores, we use a "localization-error threshold" (LET) graph, which describes the performance of each algorithm in terms of the number of images that would be detected given a desired distance threshold. In Figure 2, we vary allowable error on the X axis and for each algorithm plot the percentage of eyes at or below this error threshold in the Y-axis.

## 6 Results

The results of this competition are summarized in Table 1 and graphically presented as LET curves in Figure 2 as described above. To summarize results and rankings, we use the F-measure (also called F1-measure), defined as:

$$F(\text{precision}, \text{recall}) = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}, \quad (1)$$

where precision is $\frac{\text{TP}}{\text{TP+FP}}$ and recall is $\frac{\text{TP}}{\text{TP+FR}}$. TP is the number of correctly detected faces that overlap groundtruth, FP is the number of incorrect bounding boxes returned by the algorithm, FP′ is the number of images in which an incorrect bounding box was returned, and FR is number of faces the algorithm did not find. Here is a brief summary of our contestants' performance over each dataset.

| | | 80m-500px | | | | 200m-300px | | | | 200m-50px | | | | | Dark-150px | | | | Nonface | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | TP | FP | FR | F | TP | FP | FR | F | TP | FP | FP′ | FR | F | TP | FP | FR | F | TR | FP |
| Participants | CBED | 194 | 0 | 6 | 0.985 | 196 | 0 | 4 | 0.990 | 100 | 505 | 184 | 8 | 0.248 | 194 | 0 | 6 | 0.985 | 763 | 51 |
| | DEALTE FD 0.4.3 | 191 | 1 | 9 | 0.974 | 177 | 2 | 23 | 0.934 | 11 | 120 | 81 | 115 | 0.066 | 179 | 0 | 21 | 0.945 | 742 | 70 |
| | MBMCT | 192 | 1 | 8 | 0.977 | 191 | 7 | 9 | 0.960 | 1 | 45 | 31 | 168 | 0.008 | 178 | 0 | 22 | 0.942 | 789 | 13 |
| | MOSSE | 69 | 11 | 120 | 0.493 | 68 | 92 | 40 | 0.378 | 27 | 147 | 147 | 26 | 0.144 | 132 | 7 | 61 | 0.779 | 702 | 98 |
| | PittPatt | 198 | 0 | 2 | 0.995 | 194 | 0 | 6 | 0.985 | 0 | 0 | 0 | 200 | 0.000 | 191 | 0 | 9 | 0.977 | 800 | 0 |
| | RSFFD | 198 | 0 | 2 | 0.995 | 200 | 0 | 0 | 1.000 | 0 | 1 | 1 | 199 | 0.000 | 194 | 0 | 6 | 0.985 | 799 | 1 |
| | SIANI | 177 | 5 | 18 | 0.927 | 178 | 5 | 17 | 0.930 | 0 | 98 | 98 | 102 | 0.000 | 122 | 0 | 78 | 0.758 | 726 | 74 |
| | UCSD MPLab | 196 | 1 | 3 | 0.987 | 195 | 1 | 4 | 0.985 | 5 | 8 | 8 | 187 | 0.047 | 190 | 0 | 10 | 0.974 | 791 | 9 |
| Non-participants | Commercial A (2005) | 192 | 0 | 8 | 0.980 | 173 | 0 | 27 | 0.928 | 5 | 6 | 6 | 189 | 0.047 | 107 | 0 | 93 | 0.697 | 638 | 162 |
| | Commercial A (2011) | 144 | 0 | 56 | 0.837 | 187 | 0 | 13 | 0.966 | 0 | 0 | 0 | 200 | 0.000 | 105 | 0 | 95 | 0.689 | 800 | 0 |
| | Commercial B | 198 | 0 | 2 | 0.995 | 177 | 20 | 3 | 0.892 | 6 | 156 | 156 | 38 | 0.033 | 177 | 11 | 12 | 0.912 | 342 | 458 |
| | OpenCV 2.1 | 198 | 54 | 2 | 0.876 | 200 | 118 | 0 | 0.772 | 80 | 280 | 152 | 26 | 0.286 | 195 | 6 | 5 | 0.973 | 615 | 257 |

Table 1: Contestant results showing True Positives(TP), False Positives(FP), False Images(FP′), and False Rejects(FR) on face images. For Nonface, TR is no-face and FP is each incorrectly reported box. See Section 6 for details and discussion.

### 6.1 80m-500px

In this set, three algorithms tied for the highest F-score: RSFFD, PittPatt SDK, and Commercial B (F=0.995), missing faces in only two images. UCSD MPLab (F=0.987) secured the fourth-highest F-score. The lowest F-score belonged to MOSSE (F=0.49). The second lowest score was from Commercial A (2011) (F=0.837). Interestingly, the old version of Commercial A (2005) (F=0.980) outperformed the newer version with fewer false rejects.

While most algorithms did well in face detection, the top of Figure 2, we see that the LET graph clearly separates the different algorithms, with CBED doing much better at under 15 pixels error while RSFFD does second best and PittPatt SDK has higher percentage of eye localization when allowing errors between 18-25 pixels.

### 6.2 200m-300px

This dataset also had large size faces, but at a greater distance and slightly lower resolution the contestants performed very well overall. The algorithm with the highest F-score was RSFFD (F=1.00), who impressively found no false positives and no false rejects. A close second was CBED (F=0.990). MOSSE (F=0.378) had the lowest F-score by far, detecting about one third of the images in the dataset. Second worst was VJ-OCV2.1 (F=0.772), finding half as many false positives as it found true positives.

Again while most algorithms did well in face detection, the middle of Figure 2 clearly separates the different algorithms. CBED performed much better than the rest at under 15 pixels error and RSFFD performed second best. This time, PittPatt SDK is the 3rd best overall, among the best percentage of eye localization when allowing errors between 18-25 pixels. Surprisingly, the fixed ratio eye detector based on VJ-OCV2.1 does better than most algorithms including 3 commercial algorithms.

### 6.3 200m-50px

This dataset had the lowest resolution and most algorithms performed very poorly. RSFFD, SIANI, PittPatt SDK, and Commercial A (2011) (F=0.00) found no faces at all and MBMCT (F=0.01) found one face. Commercial A (2005) (F=0.05) outperformed its newer version (F=0.00) again. A few algorithms did better, but still not near as well as on other datasets. While CBED (F=0.248) found more true positives than VJ-OCV2.1 (F=0.286), CBED found 505 false faces in this dataset of 200 images, whereas VJ-OCV2.1 reported 280 false positives. MOSSE (F=0.144) had the third-highest F-score and the third most true positives. Because it returned at most one box per face, it is likely the most pragmatic contestant for this set. The submission from DEALTE (F=0.066) had the fourth-highest F-score. With such poor detection, eye localization is not computable or very poor for most algorithms. Only CBED and VJ-OCV2.1 had measurable eye localization (not shown). While they have high false detect rates on the faces, the eye localization could allow subsequent face recognition to determine if detected faces/eyes are really valid faces.

### 6.4 Dark-150pix

This dataset was composed of low light but good resolution images, and many algorithms did well during detection. CBED and RSFFD (F=0.985) tied for highest F-score, both missing six faces. PittPatt SDK (F=0.977) had third-highest. The algorithms with the lowest F-scores were Commercial A (2011) (F=0.689) and Commercial A (2005) (F=0.697). As usual, the old version of this commercial algorithm outperformed the new version; both detected just over half of the images in the set.

In the dark data, the eye localization of CBED, PittPatt SDK, RSFFD and UCSD MPLab all did well. Again, VJ-OCV2.1 outperformed many other algorithms including two commercial algorithms.

### 6.5 Nonface

Normal metrics such as "true positives," "false rejects," and "F-score" do not apply in this set because this set contains no faces. Its purpose is to measure false positive and true reject rates. PittPatt SDK and Commercial A (2011) (TR: 800) both achieved perfect accuracy. RSFFD (TR: 799) falsely detected one image, and UCSD MPLab (TR: 791) falsely detected only nine. The algorithms that reported the most false positives were Commercial B (TR: 342), VJ-OCV2.1 (TR: 615), and Commercial A (2005) (TR: 638).

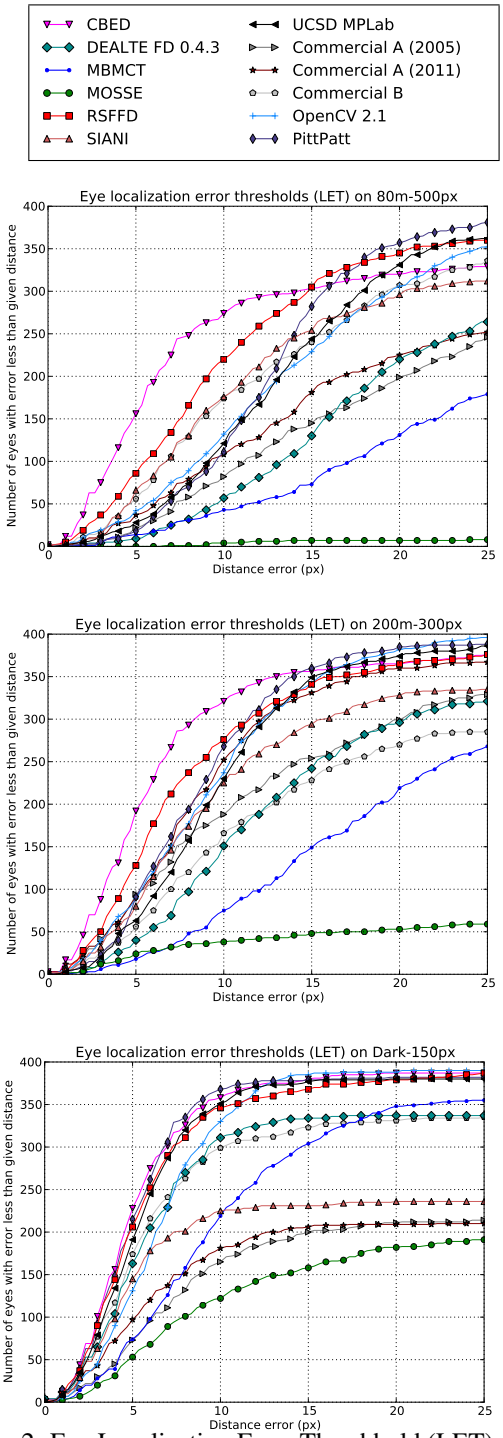



Figure 2: Eye Localization Error Threshhold (LET) curves. See Section 5 for details.

For our other datasets, contestants could use the assumption that there is one face per image to their advantage by setting a very low detection threshold and returning the most confident face. However, in a real-world setting, thresholds must be set to a useful value to reduce false positives. This was not always the case; for example, the submission from DEALTE found 70 false positives in the Non-face set but only 3 total false positives in the 80m-50px, Dark-150px, and 200m-300px sets.

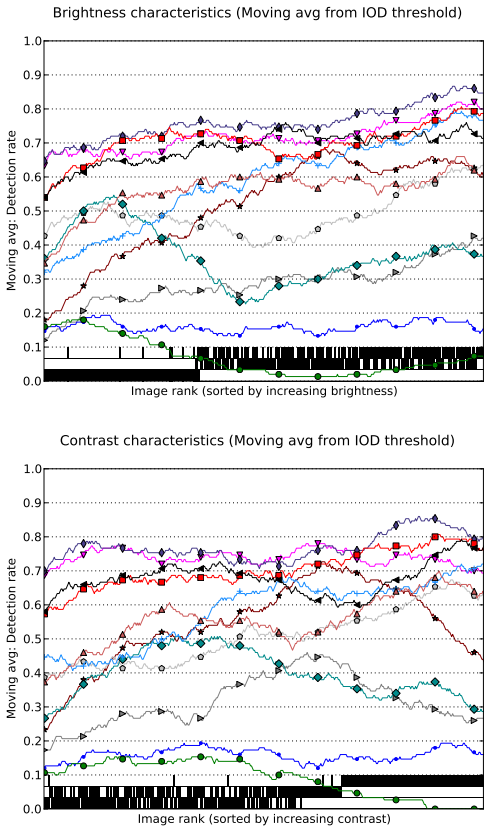



Figure 3: Detection Characteristic Curve. Measures how detection rate changes with image brightness and contrast. See Section 6.6 for a detailed description.

### 6.6 Detection Characteristic Curves

The above metrics tell us how the algorithms compare on different datasets, but why did they fail on certain images? We cannot answer definitively, but we can examine what image qualities make a face less likely to be detected. We examined this question along the dimensions of image brightness and image contrast by drawing "Detection Characteristic Curves (DCC)" as seen in Figure 3.

The X-axis of a DCC curve is image rank for the particular characteristics; where images are sorted by brightness (mean) or contrast (standard deviation). The Y-axis is a moving average of the face detection rates where a true positive counts as 1.0 and a false reject counts as 0. For this graph, we only count a detection as a true positive if both eyes are within $\frac{1}{10}$ of the average inter-ocular distance for that dataset. By graphing these metrics this way, we can present a rough sense of how detection varies as a function of brightness or contrast. Because these graphs are not balanced (for example, Dark-150px contains most of the darkest images), we plot the source for each image as a small

bar within a strip at the bottom of the graph to gain a better view of the characteristic composition. From top to bottom, these dataset strips are 80m-500px, 200m-300px, and Dark-150px. Images from 200m-50px are not included due to the poor performance.

The brightness DCC reveals that detection generally increases with increasing brightness. MOSSE and the submission from DEALTE have lowest detection rates in images of mid-brightness, but Commercial A (2011) peaks at mid-brightness.

For the contrast DCC, most of the algorithms were very clearly separated. With some algorithms (VJ-OCV2.1, Commercial B), detection rates increased with contrast. Other algorithms (the submission from DEALTE, MOSSE, SIANI, and UCSD MPLab) had a local maximum of detection rates in mid-contrast images. Some algorithms (SIANI, UCSD MPLab, and PittPatt SDK) exhibited a drop in performance on images of mid-high contrast before improving on the high-contrast images in the 80m-500px set. Others (Commercial A (2011)) exhibited the opposite trend. These results suggest that researchers should focus on improving detection rates in images of low brightness and low contrast.

## 7 Conclusions

This paper presented a performance evaluation of face detection algorithms on a variety of hard datasets. Twelve different detection algorithms from academic and commercial institutions participated.

The performance of our contestants' algorithms ranged from exceptional to experimental. Many classes of algorithms behaved differently on different datasets; for example, MOSSE had the worst F-score on 80m and 200m-300px and the third highest F-score on 200m-50px. None of the contestants did particularly well on the small, distorted faces in the 200m-50px set; this is a possible area for researchers to focus on.

There are many opportunities for future improvements on our competition model. For example, future competitions may wish to provide a more in-depth analysis of image characteristics, perhaps comparing detection rates on images of varying blur, in-plane and out-of-plane rotation, scale, compression artifacts, and noise levels. This will give researchers a better idea of why their algorithms might fail.

## Acknowledgments

We thank Pittsburgh Pattern Recognition, Inc. for contributing a set of results from their PittPatt SDK at late notice.

## References

[1] D. Bolme, , R. J. Beveridge, M. Teixeira, and B. Draper. The csu face identification evaluation system: Its purpose, features, and structure. In *Computer Vision Systems*, vol. 2626 of *LNCS*, 304–313. Springer, 2003.

[2] M. Castrillón, O. Deniz-Suarez, L. Anton-Canalis, and J. Lorenzo-Navarro. Face and facial feature detection evaluation-performance evaluation of public domain haar detectors for face and facial feature detection. In *Int. Conf. on Computer Vision Theory and Applications*, 2008.

[3] N. Degtyarev. and O. Seredin. Comparative testing of face detection algorithms. *Image and Signal Processing*, 200–209, 2010.

[4] V. Iyer, S. Kirkbride, B. Parks, W. Scheirer, and T. Boult. A taxonomy of face-models for system evaluation. In *Computer Vision and Pattern Recognition Workshops (CVPRW),*, 63–70, 2010.

[5] V. Iyer, W. Scheirer, and T. Boult. Face system evaluation toolkit: Recognition is harder than it seems. In *IEEE Biometrics: Theory Applications and Systems (BTAS)*, 2010.

[6] V. Jain and E. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, Univ. of Massachusetts, Amherst, 2010.

[7] B. Kroon, A. Hanjalic, and S. Maas. Eye localization for face matching: is it always useful and under what conditions? In *Conf. Content-based Image and Video Retrieval*, 379–388. ACM, 2008.

[8] M. Moustafa and H. Mahdi. A simple evaluation of face detection algorithms using unpublished static images. In *Int. Conf. on Intelligent Systems Design and Applications (ISDA),*, 2010.

[9] P. Phillips, H. Moon, P. Rauss, and S. Rizvi. The feret evaluation methodology for face recognition algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(10), 2000.

[10] T. Riopka. and T. Boult. The eyes have it. In *Proc. 2003 ACM SIGMM Wksp on Biometrics methods and applications*, 9–16. ACM, 2003.

[11] T. Sim, S. Baker, and M. Bsat. The CMU Pose, Illumination, and Expression (PIE) database. In *IEEE Auto. Face and Gesture Rec.*, 46–51, 2002.

[12] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2001.

# Appendices: Participants Algorithms

## A CBED

BRIAN HEFLIN
Securics Inc, Colorado Springs, CO

It can be argued that face detection is one of the most complex and challenging problems in the field of computer vision due to the large intra-class variations caused by the changes in facial appearance, expression, and lighting. These variations cause the face distribution to be highly nonlinear and complex in any space which is linear to the original image space. Additionally, in applications such as surveillance, the camera limitations and pose variations make the distribution of human faces in feature space more

dispersed and complicated than that of frontal faces. This further complicates the problem of robust face detection.

To detect faces on the two datasets for this competition, we selected the Viola-Jones face detector [37]. The Haar classifier used for both datasets was the *haarcascade-frontalFace-alt2.xml*. The scale factor was set at 1.1 and the "minimum neighbors" parameter was set at 2. The Canny edge detector was not used. The minimum size for the first dataset was (90,90) by default and (20,20) for 200m-50px.

### A.1 Correlation Filter Approach for Eye Detection

The correlation based eye detector is based on the Unconstrained Minimum Average Correlation Energy (UMACE) filter [16]. The UMACE filter was synthesized with 3000 eye images. One advantage of the UMACE filter over other types of correlation filters such as the Minimum Average Correlation Energy (MACE) filter [13] is that over-fitting of the training data is avoided by averaging the training images. Because eyes are symmetric, we use one filter to detect both eyes by horizontally flipping the image after finding the left eye. To find the location of the eye, a 2D correlation operation is performed between the UMACE filter and the cropped face image. The global maximum is the detected eye location. One issue of correlation based eye detectors is that they will show a high response to eyebrows, nostrils, dark rimmed glasses, and strong lighting such as glare from eye glasses [17]. Therefore, we modified our eye detection algorithm to search for multiple correlation peaks on each side of the face and to determine which correlation peak is the true location of the eye. This process is called "eye perturbation" and it consists of two distinct steps: First, to eliminate all but the salient structures in the correlation output, the initial correlation output is thresholded at 80% of the maximum value. Next, a unique label is assigned to each structure using connected component labeling [18]. The location of the maximum peak within each label is located and returned as a possible eye location. This process is then repeated for both sides of the face. Next, geometric normalization is performed using all of the potential eye coordinates. All of the geometrically normalized images are then compared against an UMACE based "average" face filter using frequency based cross correlation. This "average" is the geometric normalization of all of the faces from the FERET data set [14]. A UMACE filter was then synthesized from all of the normalized images. After the cross correlation operation is performed, only a small region around the center of the image is searched for a global maximum. The top two left and right $(x, y)$ eye coordinates corresponding to the image with the highest similarity are returned as potential eye coordinates and sent to the facial alignment test.

### A.2 Facial alignment

Once the eye perturbation algorithm finishes, the top two images will be returned as input into the facial alignment test. The purpose of this test is to eliminate slightly rotated face images. The first step in the eye perturbation algorithm will usually return the un-rotated face, but it is possible to receive a greater correlation score between the rotated image and the average face UMACE filter. The facial image is preprocessed by the GRAB normalization operator [15]. Next, the face image is split in half along the vertical axis and the right half is flipped. Normalized cross-correlation is then performed between the halves. A small window around the center is searched and the image with the greatest peak-to-sidelobe ratio (PSR) is then chosen as the image with the true eye coordinates.

## References


[13] A. Mahalanobis, B. Kumar, and D. Casasent. Minimum average correlation energy filters. *Appl. Opt.*, 26(17):3633–3640, 1987.

[14] P. Phillips, H. Moon, P. Rauss, and S. Rizvi. The feret evaluation methodology for face recognition algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(10), 2000.

[15] A. Sapkota, B. Parks, W. Scheirer, and T. Boult. Face-grab: Face recognition with general region assigned to binary operator. In *IEEE CVPR Wksp. on Biometrics*, vol. 1, 82–89, Los Alamitos, CA, USA, 2010. IEEE Computer Society.

[16] M. Savvides and B. Kumar. Efficient design of advanced correlation filters for robust distortion-tolerant face recognition. In *IEEE Advanced Video and Signal Based Surveillance*, 45, 2003.

[17] W. Scheirer, A. Rocha, B. Heflin, and T. Boult. Difficult detection: A comparison of two different approaches to eye detection for unconstrained environments. In *IEEE Biometrics Theory, Applications and Systems (BTAS)*, 2009.

[18] L. Shapiro and G. Stockman. *Computer Vision*. Prentice Hall, Englewood-Cliffs NJ, 2001.


## B DEALTE FD 0.4.3


A COMMERCIAL SUBMISSION FROM DEALTE
DEALTE, Saulėtekio al. 15, LT-10224 Vilnius, Lithuania


This face detector uses a variation of RealAdaBoost with weak classifiers built using trees with modified LBP-like elements of features. It scans input images in all scales and positions. To speed-up detection, we use:

- Feature-centric weak classifiers at the initial stage of the detector
- Estimation of face presence probability in somewhat bigger windows at the second stage and a deeper scanning of these bigger windows at the last stage

The algorithm analyzes and accepts/rejects samples when they exceed a predefined threshold of probability to be a face or non-face.

## C MBMCT

SÉBASTIEN MARCEL AND COSMIN ATANASOAEI
Idiap Research Institute, Marconi 19, Martigny, Switzerland

Our face detector uses a new feature – the Multi-Block Modified Census Transform (MBMCT) – that combines the multi-block idea proposed in [20] and the MCT features proposed in [19]. The MBMCT features are parametrized by the top-left coordinate $(x, y)$ and the size $w \times h$ of the rectangular cells in the $3 \times 3$ neighborhood. This gives a region of $3w \times 3h$ pixels to compute the 9-bit MBMCT:

$$MBMCT(x, y, w, h) = \sum_{i=0:8} \delta(p_i \geq \bar{p}) * 2^i, \quad (2)$$

where $\delta$ is the Kronecker delta function, $\bar{p}$ is the average pixel intensity in the $3 \times 3$ region and $p_i$ is the average pixel intensity in the cell $i$. The feature is computed in constant time for any parameterization using the integral image. Various patterns at multiple scales and aspect ratios can be obtained by varying the parameters $w$ and $h$.

The MBMCT feature values are non-metric codes and this restricts the type of weak learner to boost. We use the multi-branch decision tree (look-up-table) proposed in [20] as weak learner. This weak learner is parameterized by a feature index (e.g. dimension in the feature space) and a set of fixed outputs, one for each distinct feature value. More formally, the weak learner $g$ is computed for a sample $x$ and a feature $d$ with:

$$g(x) = g(x; d, \boldsymbol{a}) = \boldsymbol{a}[u = x^d], \quad (3)$$

where $\boldsymbol{a}$ is a look-up table with 512 entries $a_u$ (because there are 512 distinct MCT codes) and $d$ indexes the space of $x, y, w, h$ possible MBMCT parameterizations. The goal of the boosting algorithm is then to compute the optimum feature $d$ and $a_u$ entries using a training set of face and non-face images.

## Acknowledgments

The Idiap Research Institute would like to thank the Swiss Hasler Foundation (CONTEXT project) and the FP7 European TABULA RASA Project (257289) for their financial support.

## References

[19] B. Froba and A. Ernst. Face detection with the modified census transform. *IEEE Automatic Face and Gesture Recognition,*, 0:91–99, 2004.

[20] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Z. Li. Face detection based on multi-block lbp representation. In *Int. Conf. on Biometrics*, 11–18. Springer, 2007.

## D MOSSE

MOHAMMAD NAYEEM TELI
Computer Vision Group, Colorado State University, Fort Collins, CO

This face detector is based on the Minimum Output Sum of Squared Error (MOSSE) [21]. It is a correlation based approach in the frequency domain. MOSSE works by identifying a point in the image that correlates to a face. To train we created a Gaussian filter for each image, centered at a point between the eyes. Then, we took the element-wise product of the Fast Fourier Transform (FFT) of each image and its Gaussian filter to give a resulting correlation surface. The peak of the correlation surface identifies the targeted face in the image.

A MOSSE filter is constructed such that the output sum of squared error is minimized. The pairs $f_i$, $g_i$ are the training images and the desired correlation output respectively. This desired output image $g_i$ is synthetically generated such that the point between the eyes in the training image $f_i$ has the largest value and the rest of pixels have very small values. More specifically, $g_i$ is generated using a 2D Gaussian. The construction of the filter requires transformation of the input images and the Gaussian images into the Fourier domain in order to take advantage of the simple element-wise relationship between the input and the output. Let $F_i$, $G_i$ be the Fourier transform of the lower case counterparts. The exact filter $H_i$ is defined as

$$H_i^* = \frac{G_i}{F_i}, \quad (4)$$

where the division is performed element-wise. The exact filters, like the one defined in Equation 4, are specific to their corresponding image. In order to find a filter that generalizes across the dataset, we generate the MOSSE filter $H$ such that it minimizes the sum of squared error between the actual output and the desired output of the convolution. The minimization problem is represented as:

$$min_{H^*} \sum_i |F_i \odot H^* - G_i|^2, \quad (5)$$

where $F_i$ and $G_i$ are the input images and the corresponding desired outputs in the Fourier domain. This equation can be solved to get a closed form solution for the final filter $H$. Since the operation involves element-wise multiplication, each element of the filter $H$ can be optimized independently. In order to optimize each element of $H$ independently we can rewrite equation 5 as

$$H_{wv} = min_{H_{we}} \sum_i |F_{iwv} \odot H_{wv}^* - G_i|^2, \quad (6)$$

where $w$ and $v$ index the elements of $H$. This function is real valued, positive, and convex which implies the presence of a single optima. This optima is obtained by taking the partial derivative of $H_{wv}$ w.r.t. $H_{wv}^*$ and setting it to 0. By

solving for $H^*$, we obtain a closed form expression for the MOSSE filter to be

$$H^* = \frac{\sum_i G_i \odot F_i^*}{\sum_i F_i \odot F_i^*} \quad (7)$$

where $H^*$ is the complex conjugate of the final filter $H$ in the Fourier domain. A complete derivation of this expression is in the appendix of the MOSSE paper [21].

## References


[21] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui. Visual object tracking using adaptive correlation filters. *IEEE Computer Vision and Pattern Recognition(CVPR)*, 0:2544–2550, 2010.


# E RSFFD


HAM RARA, AHMED EL-BARKOUKY, AND ALY FARAG
CVIP Laboratory, University of Louisville, KY


This face detector starts by identifying the possible facial regions in the input image using the OpenCV implementation [37] of the Viola-Jones (VJ) object detection algorithm [29]. By itself, the VJ OpenCV implementation suffers from false positive errors as well as occasional false negative results when directly applied to the input image. Jun and Kim [24] proposed the concept of face certainty maps (FCM) to reduce false positive results. We use FCM to help reduce the occurrence of non-face detected regions.

The following sections describe the steps of our face detection algorithm, based on the detection module of [26].

### E.1 Preprocessing

First, each image's brightness is adjusted according to a power law (Gamma) transformation. The images are then denoised using a median filter. Smaller images are further denoised with the stationary wavelet transform (SWT) approach [23]; SWT denoising is not applied to the larger images because of processing time concerns.

Face detection is then performed at different scales. At each scale, there are some residual detected rectangular regions. These regions (for all scales) are transformed to a common reference frame. The overlapped rectangles from different scales are combined into a single rectangle. A score that represents the number of combined rectangles is generated and assigned to each combined rectangle.

### E.2 Facial Features Detection

After a facial region is detected, the next step is to locate some facial features (two eyes and mouth) using the same OpenCV VJ object detection approach but with a different cascade XML file. Every facial feature has its own training XML file acquired from various sources [37, 28]. The geometric structure of the face (i.e., expected facial feature locations) is taken into consideration to constrain the search space. The FCM concept above is again used to remove false positives and negatives. Each candidate rectangle is given another score that corresponds to the number of facial features detected inside.

### E.3 Final Decision

Every candidate face is assigned two scores that are combined into a single score, representing the sum of the number of overlapped rectangles plus the number of facial features detected. Candidates with scores above a certain threshold are considered as faces; if all candidates scores are below the threshold, the image has no faces.

## References


[22] M. Castrillón, O. Deniz-Suarez, L. Anton-Canalis, and J. Lorenzo-Navarro. Face and facial feature detection evaluation-performance evaluation of public domain haar detectors for face and facial feature detection. In *Int. Conf. on Computer Vision Theory and Applications*, 2008.

[23] R. R. Coifman and D. L. Donoho. Translation-invariant denoising. *Lecture Notes in Statistics*, 1995.

[24] B. Jun and D. Kim. Robust real-time face detection using face certainty map. In S.-W. Lee and S. Li, editors, *Advances in Biometrics*, vol. 4642 of *LNCS*, 29–38. Springer, 2007.

[25] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Int. Conf. on Image Processing*, 900–903, 2002.

[26] H. Rara, A. Farag, S. Elhabian, A. Ali, W. Miller, T. Starr, and T. Davis. In *IEEE Biometrics: Theory Applications and Systems (BTAS)*, 2010.

[27] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2001.


# F SIANI


MODESTO CASTRILÓN-SANTANA AND JAVIER LORENZO-NAVARRO
SIANI, University of Las Palmas de Gran Canaria, 35001, Spain


As an experiment, this approach combines detectors and evidence accumulation. To ease repeatability, we selected the Viola Jones [37] general object detection framework via its implementation in OpenCV [29] but these ideas could easily be applied with other detection frameworks.

Our hypothesis is that we can get better performance by introducing different heuristics in the face search. In this sense, we used the set of detectors available in the latest OpenCV release for frontal face detection ($frontalface\_default$ (FD), $frontalface\_alt$ (FA) and $frontalface\_alt2$ (FA2)), and for facial feature detection, we used $mcs\_lefteye$, $mcs\_righteye$, $mcs\_nose$ and $mcs\_mouth$ [28]).

The evidence accumulation is based on the simultaneous face and facial elements detection, or if the face is not located, in the simultaneous co-occurrence of facial feature detections. The simultaneous use of different detectors (face and/or multiple facial features) effectively reduces the influence of false alarms. These elements include the left and right eyes, nose, and mouth.

The approach is described algorithmically as follows:

$nofacefound \leftarrow false$
$nofacefound \leftarrow FaceDetectionandFFsInside()$
**if** $!nofacefound$ **then**
  $nofacefound \leftarrow FaceDetectionbyFFs()$
**end if**
**if** $nofacefound$ **then**
  $SelectBestCandidate()$
**end if**

According to the competition, the images have at most one face per image. A summarized description of each module:

- *FaceDetectionandFFsInside()*: Face detection is performed using *FA2*, *FA* and *FD* classifiers until a face candidate with more than two facial features is detected. The facial feature detection is applied within their respective expected Region of Interest (ROI) where a face container is provided. Each ROI is scaled up before searching the element. The different ROIs (format left upper corner and dimensions), considering that $sx$ and $sy$ are the face container dimensions (width and height respectively), are:
  - Left eye: $(0, 0)$ $(sx * 0.6, sy * 0.6)$.
  - Right eye: $(sx * 0.4, 0)$ $(sx * 0.6, sy * 0.6)$.
  - Nose: $(sx * 0.2, sy * 0.25)$ $(sx * 0.6, sy * 0.6)$.
  - Mouth: $(sx * 0.1, sy * 0.4)$ $(sx * 0.8, sy * 0.6)$.
- *FaceDetectionbyFFs()*: If there is no face candidate, facial feature detection is applied in the whole image. The co-occurrence of at least three geometrically coherent facial features provides evidence of a face presence. The summarized geometric rules are: The mouth must be below any other facial feature; the nose must be below both eyes but above the mouth; the centroid of the left eye must be to the left of any other facial feature and above the nose and the mouth; the centroid of the right eye must be to the right of any other facial feature and above the nose and the mouth; and the separation distance between two facial features must be coherent with the element size.
- *SelectBestCandidate()*: Because no more than one face is accepted per image, the best candidate is preferred attending the number of facial features.

The described approach could successfully detect the faces contained in the training set by considering just two inner facial features (at least one eye). To ensure our algorithm performed well on the non-face set, the minimum number of facial features required was fixed to 3. This approach worked well on all datasets except 200m-50px.

## Acknowledgments

The SIANI Institute would like to thank the Spanish Ministry of Science and Innovation funds (TIN 2008-06068)

## References

[28] M. Castrillón, O. Deniz-Suarez, L. Anton-Canalis, and J. Lorenzo-Navarro. Face and facial feature detection evaluation-performance evaluation of public domain haar detectors for face and facial feature detection. In *Int. Conf. on Computer Vision Theory and Applications*, 2008.

[29] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Int. Conf. on Image Processing*, 900–903, 2002.

[30] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2001.

# G UCSD MPLab

JAVIER MOVELLAN
Machine Perception Laboratory, University of California San Diego, CA

We used the facial feature detection architecture described in [33]. Briefly, the face finder is a Viola Jones style cascaded detector [37]. The features used were Haar wavelets that were variance-normalized. The classifier was GentleBoost [34] with cascade thresholds set by the WaldBoost algorithm [36].

No FDHD images were used in training. Instead, a custom combined dataset of about 10,000 faces was used. The sources included publicly available databases such as FDDB, GEMEP-FERA, and GENKI-SZSL [35, 31, 32] along with custom sources such as TV shows, movies, and movie trailers.

## References

[31] T. Bänziger. and K. Scherer. Introducing the geneva multimodal emotion portrayal (gemep) corpus. *Blueprint for Affective Computing: A Sourcebook*, 271–294, 2010.

[32] N. J. Butko and J. R. Movellan. Optimal scanning for faster object detection. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2751–2758, 2009.

[33] M. Eckhardt, I. Fasel, and J. Movellan. Towards practical facial feature detection. *Int. J. of Pattern Recognition and Artificial Intelligence*, 23(3):379–400, 2009.

[34] I. R. Fasel. *Learning to Detect Objects in Real-Time: Probabilistic Generative Approaches*. PhD thesis, Univ. of California at San Diego, 2006.

[35] V. Jain and E. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, Univ. of Massachusetts, Amherst, 2010.

[36] J. Sochman and J. Matas. Waldboost-learning for time constrained sequential detection. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 150–156, 2005.

[37] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2001.