Face 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 Terry 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] An Anonymous Commercial Submission from DEALTE, Saultekio 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 detection algorithms are deployed in a wide variety of applications. Unfortunately, there has been no quantitative comparison of how these detectors perform under uncontrolled circumstances. We created a dataset of low light and long distance images which possess some of the problems encountered by face detectors in the real world. We hope to advance and define the state of the art by challenging the computer vision community to compete on this dataset.

The dataset we created is composed of photographs and semi-synthetic heads photographed under varying conditions of low light, atmospheric blur, and a variety of distances: 3m, 80m, and 200m.

This paper describes the performance of the participants' face detectors against those of the Viola Jones detector and three leading commercial face detectors. We compared each detector's ability to both detect and localize faces and eyes.

1. Introduction

Over the last several decades, face detection has changed from being solely a topic for research to being commonplace in cheap point-and-shoot cameras[1]. While this may lead one to believe that face detection is a solved problem, it is still an active field of research. Most researchers use controlled datasets such as FERET[11] and PIE[9], 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 they will preform in uncontrolled circumstances.

Until now, there has not been a quantitative comparison of how detectors perform under difficult circumstances. To address this problem we created a dataset of low light and long distance images which possess some of the problems face detectors encounter in uncontrolled circumstances. We are challenging the computer vision community in this way in order to identify state-of-the-art algorithms suitable for real-world face and eye detection.

2. Background

While many toolkits, datasets, and evaluation metrics exist for evaluating face recognition and identification systems, [11, 1] these are not designed for evaluating simple face detection measures. Overall there has been little focus on detection.

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[26]. 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. Other formal competitions are focused in the domain of recognition [11, 7, 3]. Intending to solve a different problem, they do not report simple detection scores.

The Conference on Intelligent Systems Design and Applications [6] performed a face detection competition with two contestants in 2010. Their datasets included a law enforcement mugshot set of 845 images, controlled digital

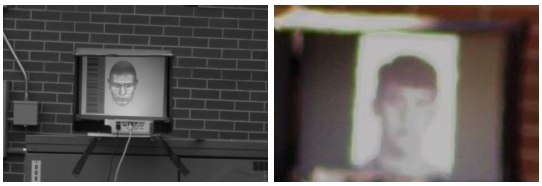
[1] At the time of writing the Canon PowerShot A495 contains face detection controlled autofocus and sells for under ninety dollars

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." We hope to help satisfy this opportunity by evaluating several algorithms in datasets with similar conditions such as our 200m-50px set.

3. Dataset

We set out to create a dataset which would pose some of the problems presented by unconstrained detection. To do this, we created four sub-sets, each of which presents a different problem in order to isolate how a detector performs on specific challenges. Our naming scheme generally follows *distance-width*, where *distance* is the capture distance and *width* is the approximate width of the face in pixels.

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



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

Figure 1: Samples from the dataset

3.1. 80m-500px

To create the first sub-set, we photographed semi-synthetic heads generated from PIE at 80 meters indoors using a Canon 5D mark II with a Sigma EX 800mm lens. See Figure 1a. These semi-synthetic heads were generated according to the protocols discussed in [4, 5]. This camera lens combination produced a controlled mid-distance dataset which provides a useful base line for the uncontrolled long distance sub-sets.

3.2. Dark-150px

For the second sub-section, we recaptured PIE[9] at close range, approximately 3m, in a low light environment. A sample of our dataset is provided in Figure 1b. We captured this set with a Salvador camera. While the Salvador can operate in extremely low light conditions, it produces a low resolution and high noise image. The noise coupled with low resolution was chosen to test the detectors' sensitivity to noise and darkness.

3.3. 200m-300px

For the third sub-set, we photographed the same semi-synthetic heads, this time from 200 meters outside. See Figure 1c. We used a Canon 5D mark II with a Sigma EX 800mm lens and a Canon EF 2x II Extender, resulting in an effective 1600mm lens. The captured faces suffered varying degrees of atmospheric blur.

3.4. 200m-50px

For the fourth sub-set, we rephotographed 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 heavy atmospheric blur and loss of contrast. We chose a subset of these images, distributed such that our configuration of Viola Jones correctly identified the face in 40% of the images. Of this group, we manually hand-picked only images that contained discernible detail around the eyes, nose, and mouth.

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.

3.6. Dataset Composition

Given these datasets, we randomly selected 50 images of each subset to create the training dataset; each sub-set was placed into its own folder and distributed to the contestants prior to the competition for training purposes. The training set was also released with groundtruth for the face bounding box and eye coordinates. The purpose of this set was not to provide a dataset to train an algorithm, 50 images is to few for that, but for the participants to internally test and tune their algorithm.

We then randomly selected 200 images of each subset to create the testing set. 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

For completeness, we compared the algorithms' performance against three leading commercial algorithms. Two of these (*"Commercial A (2005)"* and *"Commercial A (2011)"*) are different versions of the same algorithm developed six years apart.

We also benchmarked the standard Viola Jones Haar Classifier, compiled with OpenCV 2.1 using the *frontalface_alt2* cascade, a scale of 1.1, 4 minimum neighbors,

20×20 minimum feature size, and canny edge detection enabled. These parameters where chosen by running a number viola jones instances with varying parameters. The choice was made to let Viola Jones have a high false positive rate compared in order to increase the true positive rate. This choice was made due to the nature of the dataset. Algorithms such as Correlation-based Eye Detection use similar Viola Jones parameters. These parameters typically yield high performance in many scenarios[26].

We aimed to detect both face bounding boxes and eye coordinates. Because Commercial B only detects eye coordinates, we generated bounding boxes by using the ratios described in `csuPreprocessnormalize.c`, part of the *CSU Face Evaluation and Identification Toolkit* [1]. Similarly, because we only used Viola Jones to detects faces and not eyes, we defined the eyes as points a predefined ratio away from the midpoint of the bounding box along the X and Y axes. These ratios were empirically determined because the CSU normalization ratios are designed to yield bounding boxes when given eye coordinates, not the other way around.

Detailed descriptions of the contributors' algorithms are presented as an appendix.

5. Evaluation metrics

We judged the contestants based on detection and localization of faces. To gather metrics, we compared each contestant's results with hand-created groundtruth. Because of the nature of our datasets, a simple "true positive" score does not necessarily reveal the true performance of a given algorithm. We represent the full spectrum between "true positive" and "false positive" by assigning a localization error score to each eye in the ground truth. This score is equal to the Euclidean distance between the groundtruth eye and the identified eye.

To present these scores in Figures 2, we vary the distance along the X axis and graph the percentage of eyes that satisfy this threshold in the Y-axis. These "localization-error threshold" (LET) graphs describe the performance of each algorithm in terms of how many images would be detected given a desired distance threshold.

The other 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. In the non-face set, only true rejects and false positives are relevant because those images contain no faces.

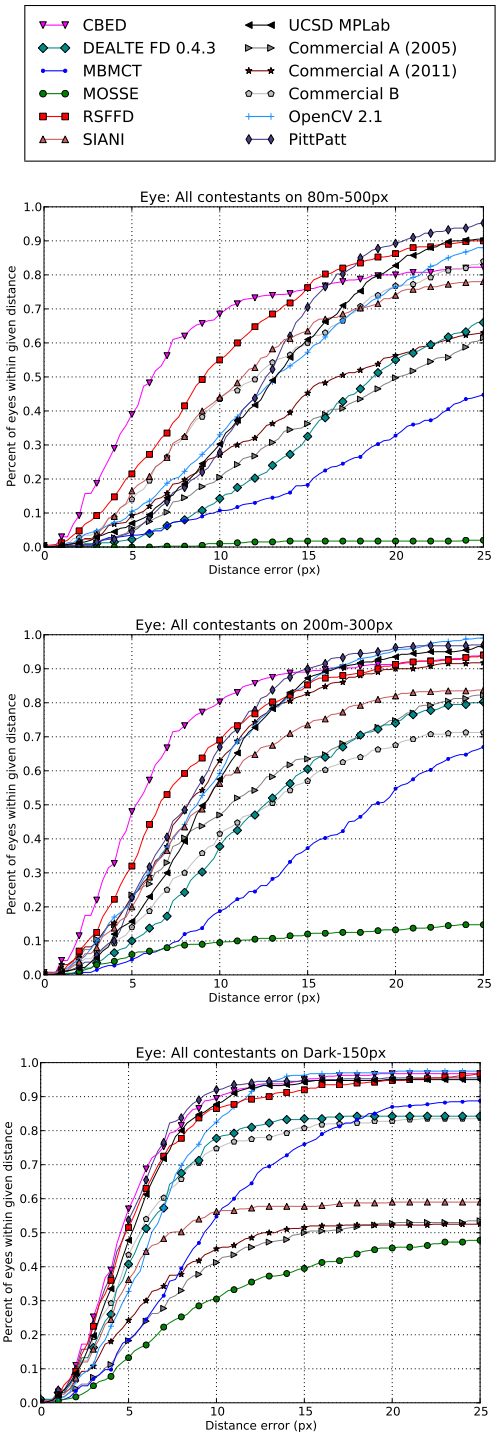


Figure 2: Eye Localization Error Threshhold (LET) curves. See Section 5 for details.

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 rank contestants, we use the F-measure,

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

Table 1: Contestant results. See Section 6 for a description of TP, FP, and FR. For the Nonface set, TR is the number of images where the algorithm reported no face and FP is the number of images where the algorithm found a face.

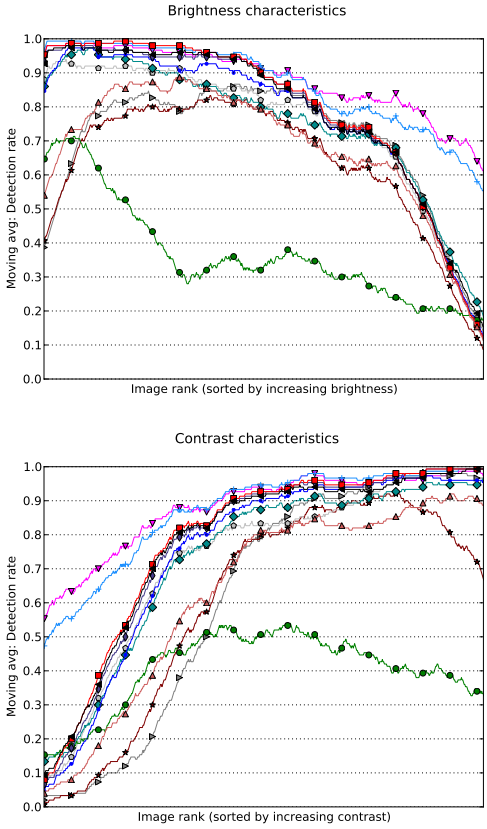


Figure 3: Detection Characteristic Curve. Measures how detection rate changes with image brightness and contrast. See Section 6.6 for a detailed description.

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 $\frac{\text{TP}}{\text{Total}}$. TP is the number of correctly detected faces that overlap groundtruth, FP is the number of incorrect bounding boxes returned by the algorithm, and FR is number of faces the algorithm did not find. Here is a brief summary of our contestants' performance over each dataset.

6.1. 80m-500px

In this set, three algorithms tied for the highest F-score: Robust Score Fusion-based Face Detection, 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.

6.2. Dark-150pix

Correlation-based Eye Detection and Robust Score Fusion-based Face Detection (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). This is ironic because the old version of this commercial algorithm outperformed the new version. Both detected just over half of the images in the set. SIANI (F=0.758) was third-worst.

6.3. 200m-300px

This was one of our most sensible datasets. As such, the contestants performed very well overall. The algorithm with the highest F-score was Robust Score Fusion-based Face Detection (F=1.00), who impressively found no false positives and no false rejects. A close second was Correlation-based Eye Detection (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 Viola Jones (OpenCV 2.1) (F=0.772), finding almost as many false positives as it found true positives.

6.4. 200m-50px

This dataset was the most brutal set we evaluated. Though Correlation-based Eye Detection (F=0.248) found more true positives than Viola Jones (OpenCV 2.1) (F=0.286), Correlation-based Eye Detection found 505

false faces in this dataset of 200 images, whereas Viola Jones (OpenCV 2.1) reported 280 false positives. This likely makes Viola Jones (OpenCV 2.1) slightly better suited for real-world detection scenarios. 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.

Most algorithms performed very poorly. Robust Score Fusion-based Face Detection, SIANI, PittPatt SDK, and Commercial A (2005) (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.

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. Robust Score Fusion-based Face Detection (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), Viola Jones (OpenCV 2.1) (TR: 615), and Commercial A (2005) (TR: 638).

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? Put more formally, what image qualities make a probe less likely to be detected? We examined this question along the dimensions of image brightness and image contrast by drawing "Detection Characteristic Curves" as seen in Figure 3.

The X-axis of a DCC curve is image rank, where images are sorted by brightness (mean) or contrast (standard deviation). The Y-axis is a moving average of the detection rates where a true positive counts as 1.0 and a false reject counts as 0. By graphing these metrics this way, we can present a rough sense of how detection varies as a function of brightness or contrast.

7. Discussion

After processing all the results, some interesting trends emerged.

- **Bounding boxes:** Since we did not define a standardized bounding box each group returns a slightly different size. Because of this, different algorithms returned boxes that were consistently tighter or smaller than our groundtruth. This does not affect the number of detected faces, but it manifests itself on the LET curves as a shift in the X-axis where the localization accuracy begins to plateau.

- **False positives:** In the 200m-50px sub-set, Viola-Jones based detectors were the only ones that consistently detected faces, but this came at the cost of returning large numbers of false positives.

  The Nonface sub-set exposed weaknesses in certain detectors. Commercial B has 458 false positives and OpenCV 2.1 has 257 false positives. Both these numbers are unacceptable. Commercial A has improved between versions, dropping from 162 false positive to 0.

- **Image characteristics:** Image brightness and contrast played an interesting role in our experiments. For example, as can be seen from the brightness detection characteristic curve in Figure 3, the algorithms generally detected more dark images than bright ones. As the images get brighter, the performance begins to drop off and falls sharply on very bright images. In particular, the majority of images detected by MOSSE were very dark, and its detection rate quickly plateaued as brightness increased.

  As for contrast, we found that better detection generally correlates with increasing contrast. The exceptions to this rule included MOSSE where detection peaked on images of mid-level contrast, and Commercial A (2011) where detection rates dropped sharply on images with very high contrast. Interestingly, Commercial A (2005) did not exhibit this trend.

8. 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 would also like to thank Pittsburgh Pattern Recognition, Inc. for contributing a set of results from their PittPatt SDK at late notice.

References

[1] D. Bolme, J. Ross Beveridge, M. Teixeira, and B. Draper. The csu face identification evaluation system: Its purpose, features, and structure. In J. Crowley, J. Piater, M. Vincze, and L. Paletta, editors, *Computer Vision Systems*, volume 2626 of *Lecture Notes in Computer Science*, pages 304–313. Springer Berlin / Heidelberg, 2003. 10.1007/3-540-36592-3_29.

[2] M. Castrilln-santana, O. Dniz-surez, L. Antn-canals, and J. Lorenzo-navarro. Face and facial feature detection evaluation performance evaluation of public domain haar detectors for face and facial feature detection. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 2, pages 167 – 172, 2008.

[3] P. J. Grother, G. W. Quinn, and P. J. Phillips. Report on the evaluation of 2d still-image face recognition algorithms - nist interagency report 7709. *National Institute of Standards and Technology*, 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), 2010 IEEE Computer Society Conference on*, pages 63 –70, june 2010.

[5] V. Iyer, W. Scheirer, and T. Boult. Face system evaluation toolkit: Recognition is harder than it seems. In *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*, pages 1 –8, sept. 2010.

[6] M. Moustafa and H. Mahdi. A simple evaluation of face detection algorithms using unpublished static images. In *Intelligent Systems Design and Applications (ISDA), 2010 10th International Conference on*, pages 1 –5, 29 2010-dec. 1 2010.

[7] P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 947 – 954 vol. 1, june 2005.

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

[9] T. Sim, S. Baker, and M. Bsat. The cmu pose, illumination, and expression (pie) database. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 46 –51, may 2002.

Appendix: Descriptions of Participants Algorithms

A. Correlation-based Eye Detection

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 [28]. 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 [13]. 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 [10] 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 [14]. 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 [15]. 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 [11]. 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 [12]. 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

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

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

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

[13] M. Savvides and B. V. Kumar. Efficient design of advanced correlation filters for robust distortion-tolerant face recognition. In *IEEE Intl. Conference on Advanced Video and Signal Based Surveillance*, page 45, Los Alamitos, CA, USA, 2003. IEEE Computer Society.

[14] 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 Intl. Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2009.

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

[16] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):151–173, May 2004.

B. DEALTE FD 0.4.3

AN ANONYMOUS COMMERCIAL SUBMISSION FROM DEALTE
DEALTE, Saultekio 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 several techniques are used:

- 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 [18] and the MCT features proposed in [17]. The MBMCT features are parametrized by the top-left coordinate (x, y) and the size $w \times h$ of the rectangular cells in the 3×3 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 δ is the Kronecker delta function, $\bar{p}$ is the average pixel intensity in the 3×3 region and p_i is the average pixel intensity in the cell i. The feature is computed in constant time for any parametrization 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 [18] as weak learner. This weak learner is parametrized 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 parametrizations. 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

[17] B. Froba and A. Ernst. Face detection with the modified census transform. *Automatic Face and Gesture Recognition, IEEE International Conference on*, 0:91, 2004.

[18] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Z. Li. Face detection based on multi-block lbp representation. In *ICB*, pages 11–18, 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) [19]. 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,

$$H_i^* = \frac{G_i}{F_i}, \tag{4}$$

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:

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

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

$$H_{wv} = min_{H_{we}} \sum_i |F_{iwv} \odot H^*_{wv} - G_i|^2, \tag{6}$$

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

$$H^* = \frac{\sum_i G_i \odot F_i^*}{\sum_i F_i \odot F_i^*} \tag{7}$$

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 [19].

References

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

E. Robust Score Fusion-based Face Detection

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 [20] of the Viola-Jones (VJ) object detection algorithm [27]. 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 [22] 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, which is based on the face detection module of [25].

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 [21]; 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 [20, 24]. 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

[20] G. Bradski. The opencv library. dr. *Dobbs Journal of Software Tools*, 2000.

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

[22] 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*, volume 4642 of *Lecture Notes in Computer Science*, pages 29–38. Springer Berlin / Heidelberg, 2007.

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

[24] C. S. M, 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 *In International Conference on Computer Vision Theory and Applications - VISAPP*, 2008.

[25] H. Rara, A. Farag, S. Elhabian, A. Ali, W. Miller, T. Starr, and T. Davis. In *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference*, pages 1 –6, sept. 2010.

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 [28] general object detection framework via its implementation in OpenCV [27] 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 [26]).

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 activation of different detectors (face and multiple facial features or just 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

[26] M. Castrilln-santana, O. Dniz-surez, L. Antn-canals, and J. Lorenzo-navarro. Face and facial feature detection evaluation performance evaluation of public domain haar detectors for face and facial feature detection. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 2, pages 167 – 172, 2008.

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

[28] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):151–173, May 2004.

G. UCSD MPLab

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

We used the facial feature detection architecture described in [31]. Briefly, the face finder is a Viola Jones style cascaded detector [35]. The features used were Haar wavelets that were variance-normalized. The classifier was GentleBoost [32] with cascade thresholds set by the WaldBoost algorithm [34].

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 [33, 29, 30] along with custom sources such as TV shows, movies, and movie trailers.

References

[29] K. R. Bañziger, T and Scherer. Introducing the geneva multimodal emotion portrayal (gemep) corpus. *K. R. Scherer, T. Bañziger, and E. B. Roesch, editors, Blueprint for Affective Computing: A Sourcebook, Series in affective science*, pages chapter 6.1, 271–294, 2010.

[30] N. J. Butko and J. R. Movellan. Optimal scanning for faster object detection. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2751–2758, 2009.

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

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

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

[34] J. Sochman and J. Matas. Waldboost - learning for time constrained sequential detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 150 – 156 vol. 2, june 2005.

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