Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 December 2022 doi:10.20944/preprints202212.0498.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Coupling Retinaface and Depth Information to Filter False Positives

Loris Nanni [1,*] , Sheryl Brahnam[2], Alessandra Lumini[3], Andrea Loreggia[4]

[1]DEI, University of Padova, Via Gradenigo, 6, 35131 Padova, Italy; nanni@dei.unipd.it
[2]Information Technology and Cybersecurity, Missouri State University, 901 S. National Street, Springfield, MO 65804 USA; sbrahnam@missouristate.edu
[3]DISI, Università di Bologna, Via Sacchi 3, 47521 Cesena, Italy; alessandra.lumini@unibo.it
[4]DII, Università di Brescia, via branze 38, 25123 Brescia, andrea.loreggia@unibs.it
* Correspondence: nanni@dei.unipd.it;

Abstract: Face detection is an important problem in computer vision because it enables a wide range of applications, such as facial recognition and analysis of human behavior. The problem is challenging because of the large variations in facial appearance across different individuals and different lighting and pose conditions. One way to detect faces is to utilize a highly advanced face detection method, such as RetinaFace, which uses deep learning techniques to achieve high accuracy in various datasets. However, even the best face detectors can produce false positives, which can lead to incorrect or unreliable results. In this paper, we propose a method for reducing false positives in face detection by using information from a depth map. A depth map is a two-dimensional representation of the distance of objects in an image from the camera. By using the depth information, the proposed method is able to better differentiate between true faces and false positives. The authors evaluate their method on a combined dataset of 549 images, containing a total of 614 upright frontal faces. The results show that the proposed method is able to significantly reduce the number of false positives without sacrificing the overall detection rate. This indicates that the use of depth information can be a useful tool for improving face detection performance.

Keywords: face detection; depth map; deep learning; filtering

1. Introduction

The development of face detection algorithms has been motivated by the need to automate the process of identifying and locating faces in digital images [1]–[3]. These algorithms have evolved over time, starting with knowledge-based methods that relied on human expertise to define the features of a face. These were followed by feature invariant approaches, which sought to identify faces based on their geometric characteristics, such as the relative positions of the eyes, nose, and mouth. To tackle the problem of face detection, researchers have developed a number of approaches based on both traditional computer vision techniques and more recent deep learning methods [4], [5]. Traditional techniques often rely on hand-crafted features and ensembles of classifiers to detect faces in images. These methods are typically computationally efficient but can struggle with variations in face appearance and are sensitive to changes in illumination and pose. More recent approaches based on deep learning have shown great promise in overcoming these limitations. These methods use convolutional neural networks (CNNs) trained on large datasets of face images to learn highly discriminative features for face detection. CNNs have the advantage

© 2022 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 December 2022 doi:10.20944/preprints202212.0498.v1

of being able to automatically learn features from data, which can be more robust and generalizable than hand-crafted features. Additionally, CNNs can be trained using large-scale parallel computing, which allows for efficient training of very deep and complex models [14].

One of the most well-known face detection systems based on deep learning is the Single Shot Detector (SSD) [6], which uses a CNN to predict bounding boxes and class probabilities for faces in an image. SSDs are known for their high speed and real-time performance, making them well-suited for applications such as video surveillance and face tracking. Another popular approach to face detection is the Multi-task Cascade Convolutional Neural Network (MTCNN) [7], which uses a cascade of three CNNs to first identify potential face regions, then refine the bounding boxes and facial landmarks, and finally classify the detected faces. MTCNNs have been shown to achieve high accuracy on a range of face detection benchmarks.

Overall, face detection remains a challenging problem due to the wide variations in face appearance and the need for real-time performance. However, the development of deep learning methods has greatly improved the state-of-the-art in face detection and has opened up new possibilities for applications that rely on the detection and analysis of faces.

Template matching methods [5], on the other hand, use a pre-defined template of a face to search for matches in an image. These methods can be effective, but they are limited by the fact that the template must be carefully designed to account for variations in face appearance. Appearance-based methods, also known as holistic methods, have become increasingly popular in recent years. These methods use machine learning techniques to learn the characteristic features of a face from a large dataset of labeled images. Because these methods can learn to recognize faces automatically, they are not limited by the constraints of template matching approaches.

The Viola-Jones algorithm (VJ) [8] is a popular and effective method for detecting objects in images. It was specifically designed for real-time object detection and achieved this using three key techniques: an integral image strategy for efficient Haar feature extraction [9], a boosting algorithm called AdaBoost for combining a set of weak classifiers, and an attentional cascade structure for fast negative rejection.

One of the key advantages of the Viola-Jones algorithm is its efficiency, which allows it to run in real-time on standard hardware. This is achieved through the use of Haar-like features, which are simple, rectangular shapes that can be calculated quickly and easily. The boosting algorithm is also effective at reducing false positives, which can be a common problem in object detection algorithms.

However, there are also some limitations to the Viola-Jones algorithm. One of these is that it is not always effective in detecting objects in unconstrained environments, such as those found in the Face Detection Dataset and Benchmark (FDDB) [10], where it sometimes fails to detect faces [11]. This is because the Haar-like features used by the algorithm can struggle to handle variations in pose, lighting, facial expression, and other factors that can affect the appearance of a face.

To overcome these limitations, several extensions and enhancements to the original Haar-like features have been proposed. These include rotated Haar-like features, which are designed to be more robust to rotation, and sparse features, which are designed to be more efficient to compute. Additionally, more powerful image descriptors, such as LBP [12] and HoG [13], have been developed and used as alternatives to Haar-like features in some object detection algorithms.

The work of Li et al. at Intel Labs focused on improving the convergence speed of SURF cascade [14], a common technique used in face detection algorithms, by using multidimensional SURF features [15] and logistic regression instead of single-dimensional Haar features and decision trees. This approach was shown to be effective in improving the speed and accuracy of face detection algorithms.

The work of Mathias et al. [16] also focused on improving the performance of face detection algorithms, but from a different angle. They proposed two simple approaches that were shown to outperform several commercial face detectors, including those from Google Picasa, Face.com, Intel Olaworks, and Face++. The first approach was based on rigid templates, similar to the VJ algorithm, and the second used a deformable part model (DPM), which is a more generalizable object detection approach

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 December 2022 doi:10.20944/preprints202212.0498.v1

that combines latent variable estimation and clustering with multiple components and deformable parts to better handle intra-class variance. Both of these approaches showed promise in improving the performance of face detection algorithms. 2D face detectors are usually different in the techniques they use to detect faces in images. Nilsson et al. [17] use a method called Successive Mean Quantization Transform (SMQT) to extract features from the image, which they then apply to a classifier called a Split up Sparse Network of Winnows (SN). Asthana et al. [18] use a technique called face fitting, which involves modeling a face shape using a set of parameters that control a deformable model of the face. Markuš et al. [19] combine a modified version of the Viola-Jones (VJ) method with an algorithm for detecting salient facial landmarks. Liao et al. [11] propose a new feature called scale-invariant NPD, and also expand the VJ tree classifier to have a deeper quadratic tree structure.

While older techniques, such as for instance combining boosting with Modified Census Transform (MCT) [28], were successful in improving face detection algorithms, more recent techniques have continued to advance the field. For example, convolutional neural networks (CNNs) have become increasingly popular for face detection and have shown impressive results in terms of accuracy and speed [20]–[23]. These networks are able to learn complex patterns in data and can be trained on large datasets, which has helped improve the performance of face detection algorithms. Additionally, techniques such as transfer learning, which involves using pre-trained CNNs on large datasets and fine-tuning them for specific tasks, have further improved the performance of face detection algorithms. In the context of 2D face detection, deep learning methods have been shown to be effective in detecting faces in images and videos. These methods, such as R-CNN [24] and Deep Dense Face Detector (DDFD) [21], use convolutional neural networks (CNNs) to extract features from images and then classify them using support vector machines (SVMs). These methods have the advantage of being able to handle a wide range of face orientations and sizes, without requiring pose or landmark annotations. They have been shown to outperform traditional face detection methods in terms of accuracy and speed.

RetinaFace [25] is a face detection algorithm considered to be a state-of-the-art face detector because it is able to combine high-level and low-level semantic information in order to perform single-shot multi-level face localization. This means that it is able to detect faces in a single stage, using a 5-level feature pyramid network (FPN) that allows it to process multiscale feature maps. This improves the detection speed of the algorithm, making it faster and more efficient than many other face detection algorithms.

The use of 3D information in face detection can improve accuracy by providing additional cues about the shape and structure of the face, which can be used to differentiate it more effectively from other objects in the scene. Depth information can also help resolve occlusions, where part of the face may be obscured by another object, by allowing the algorithm to infer the shape of the face behind the occluder.

There are several different 3D sensors and devices on the market that can be used for face detection, each with its own unique set of strengths and limitations. The Kinect [26], for example, is a popular choice due to its low cost and ease of use, but its performance is limited by the resolution of its depth map and the fact that it can only capture a single depth value per pixel. More advanced sensors, such as the MU-2 stereo imaging system [27] and the Minolta Vivid 910 range scanner [28], can provide higher resolution depth maps and more accurate depth measurements, but they are typically more expensive and more difficult to use.

Overall, the use of 3D information in face detection can significantly improvement accuracy, but it also introduces additional challenges and complexities in terms of both hardware and software. As 3D sensing technology continues to advance and become more affordable, it is likely that we will see more widespread adoption of 3D face detection techniques in various of applications.

The work of Shotton et al. [29] used pairwise pixel comparisons in depth images to classify body joints and parts for pose recognition. Mattheij et al. [30] compared square regions in a pairwise fashion for face detection. Jiang et al. [31] integrated texture and stereo disparity information to filter out locations unlikely to contain a face. Anisetti et al. [32] a located faces by applying a coarse detection method followed by a

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 December 2022 doi:10.20944/preprints202212.0498.v1

technique based on a 3D morphable face model. Taigman et al. [3] found that combining a 3D model-based alignment with DeepFace, trained on the Labeled Faces in the Wild dataset [33], generalized well for detecting faces in an unconstrained environment. Nanni et al. [1] overcame the problem of increased false positives when combining different face detectors by applying different filtering steps based on information in the Kinetic depth map.

The best performing system developed in this work is validated on a challenging dataset [1] that contains depth and 2D images, with contains 549 samples including 614 upright frontal faces. The filtering steps used in the system successfully decrease the number of false positives without significantly affecting the detection rate of Retina Face. The paper describes the face detection strategy in Section 2, presents the experiments in Section 3, and provides a summary and notes on future directions in Section 4. The code and dataset used in the paper are available on GitHub at https://github.com/LorisNanni.

2. Materials and Methods

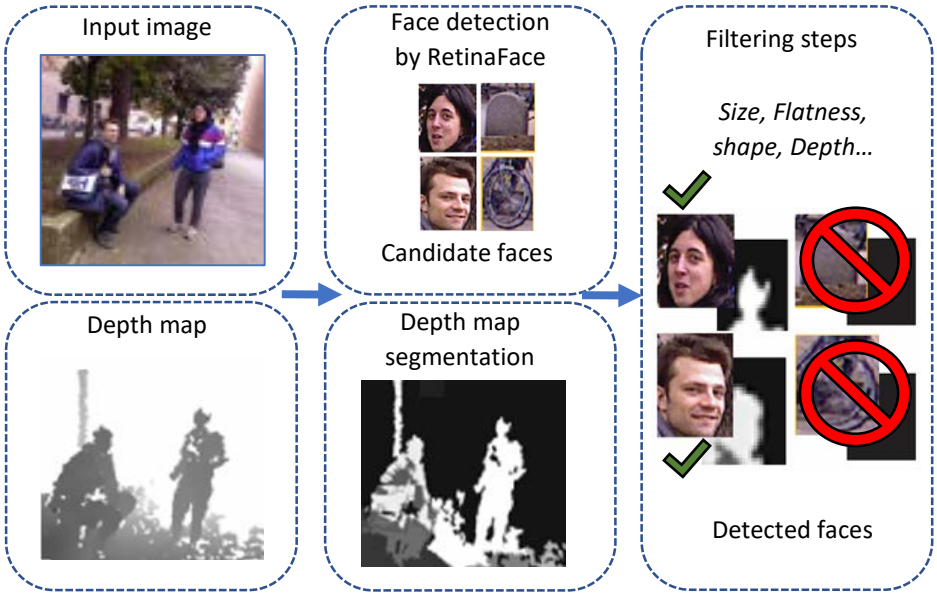

Figure 1. A general illustration of the proposed face detection approach. First, the input image undergoes a first round of detection RetinaFace (top). Second, segmentation is performed using the depth map of the image (bottom). Finally, the candidate faces undergo a filtering process to select regions with faces (right).

The face detection system developed in this work is illustrated in Figure 1. First, face detection with the face detector RetinaFace is performed on the raw color images (see the top two boxes in Figure 1). RetinaFace selects many candidate regions that contain no faces. Second, the number of false positives is reduced by aligning the depth maps (see the bottom two boxes). Alignment is accomplished by calibrating the color and depth information, as explained in [34]. Briefly, the positions of the depth samples in 3D space are calculated using the intrinsic camera parameters of focal length and principal point of the depth camera. These intrinsic parameters, along with the extrinsic parameters of the camera pair system, are then projected onto 2D space. Next, color and depth values are associated with each sample, as described in

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 December 2022 doi:10.20944/preprints202212.0498.v1

section 2.1. To reduce computation time, we apply this process only to those regions containing the candidate faces. Finally, these regions are filtered (see the box on the right of Figure 1) to remove false positives. The details of this last process are described in section 2.3.

2.1. Depth map alignment and segmentation

The raw color images and depth maps are segmented in tandem using a procedure similar to that described in [35], which involves transforming each sample into a six-dimensional vector and then clustering the point set using the Mean Shift algorithm [36]. One of the benefits of this algorithm is that it provides an excellent trade-off between segmentation accuracy and computational complexity.

Transformation into a six-dimensional vector is accomplished as follows. Every sample in the Kinetic depth map is a 3D point: $p_i, i = 1, \dots, N$, with N the number of points. As described in [34], the joint calibration of the depth and color cameras facilitates a reprojection of the depth samples over the corresponding pixels in the color image. In this way, each point can be associated with the 3D spatial coordinates (x, y, and z) of p_i and the RGB color components. However, these two representations are not directly comparable because they lie in entirely different spaces, and all components must be comparable to extract multidimensional vectors for the Mean Shift clustering algorithm. Thus, a conversion must be performed so the color values lie in the CIELAB uniform color space. This space represents color in a 3D space with values representing lightness (L) ranging from black (0) to white (100), values (a) ranging from green (-) to red (+), and (b) from blue(-) to yellow (+). With this conversion, the Euclidean distance between the color vectors can now be used in the Mean Shift algorithm.

The algorithm can be described more formally as follows. The color information of each scene point in the CIELAB color space, c, is defined as the 3D vector:

$$p_i^c = \begin{bmatrix} \mathrm{L}(p_i) \\ \mathrm{a}(p_i) \\ \mathrm{b}(p_i) \end{bmatrix}, \quad i = 1, \dots, N. \tag{1}$$

The geometry, g, is defined by the 3D coordinates of each point as:

$$p_i^g = \begin{bmatrix} \mathrm{x}(p_i) \\ \mathrm{y}(p_i) \\ \mathrm{z}(p_i) \end{bmatrix}, \quad i = 1, \dots, N. \tag{2}$$

Because the scene segmentation algorithm must be insensitive to the relative scaling of the point-cloud geometry and the color distances and geometry must be consistent, all components of p_i^g are normalized with respect to the average of the standard deviations of the point coordinates in the three dimensions $\sigma_g = (\sigma_x + \sigma_y + \sigma_z)/3$, which results in the following vector:

$$\begin{bmatrix} \bar{\mathrm{x}}(p_i) \\ \bar{\mathrm{y}}(p_i) \\ \bar{\mathrm{z}}(p_i) \end{bmatrix} = \frac{3}{\sigma_x + \sigma_y + \sigma_z} \begin{bmatrix} \mathrm{x}(p_i) \\ \mathrm{y}(p_i) \\ \mathrm{z}(p_i) \end{bmatrix} = \frac{1}{\sigma_g} \begin{bmatrix} \mathrm{x}(p_i) \\ \mathrm{y}(p_i) \\ \mathrm{z}(p_i) \end{bmatrix}. \tag{3}$$

The color information vectors are normalized as well. The average of the standard deviations of the L, a, and b color components are computed to produce the final color representation:

$$\begin{bmatrix} \bar{\mathrm{L}}(p_i) \\ \bar{\mathrm{a}}(p_i) \\ \bar{\mathrm{b}}(p_i) \end{bmatrix} = \frac{3}{\sigma_L + \sigma_a + \sigma_b} \begin{bmatrix} \mathrm{L}(p_i) \\ \mathrm{a}(p_i) \\ \mathrm{b}(p_i) \end{bmatrix} = \frac{1}{\sigma_c} \begin{bmatrix} \mathrm{L}(p_i) \\ \mathrm{a}(p_i) \\ \mathrm{b}(p_i) \end{bmatrix}. \tag{4}$$

Now that the color information vectors and geometry are normalized, they can be combined to produce the final representation f:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 December 2022 doi:10.20944/preprints202212.0498.v1

$$p_i^f = \begin{bmatrix} \bar{L}(p_i) \\ \bar{a}(p_i) \\ \bar{b}(p_i) \\ \lambda_{\bar{x}} \\ \lambda_{\bar{y}} \\ \lambda_{\bar{z}} \end{bmatrix}, \quad (5)$$

where the parameter λ adjusts the contribution to the final segmentation of color and geometry: low values of λ represent high color relevance, and low values indicate high geometry relevance. By varying the parameter λ the algorithm can become a color-based segmentation ($\lambda = 0$) or a geometry (depth) only segmentation ($\lambda \to \infty$) (see [35] for a complete discussion on automatically tuning λ to an optimal value).

After the vectors p_i^f are calculated, they can be clustered by the Mean Shift algorithm [36] to segment the sampled scene. Further refinement is possible by removing regions smaller than some threshold since these regions are usually the result of noise. An example of a segmented image using this method of segmentation is provided in Figure 2.

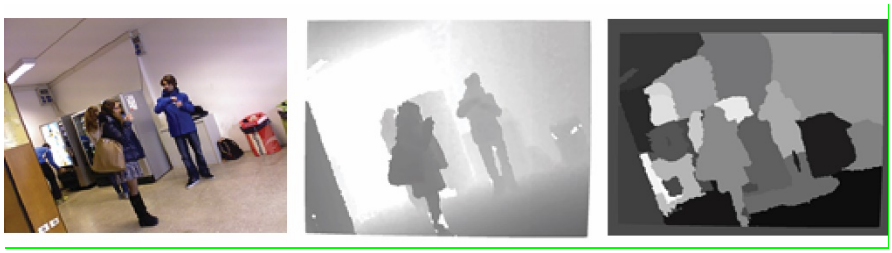

Figure 2. Raw color image (left), depth map (middle), and segmentation map (right).

2.2. Face detector: RetinaFace

RetinaFace [25] is a recent pixel-wise face detection method that, along with box classification and regression branches, applies extra-supervised and self-supervised techniques/learning tasks. The combination of these different tasks allows the detector to predict a face score, face box, five facial landmarks, and the 3D position and correspondence of each facial pixel. A schematic structure of the system is provided in Figure 3.

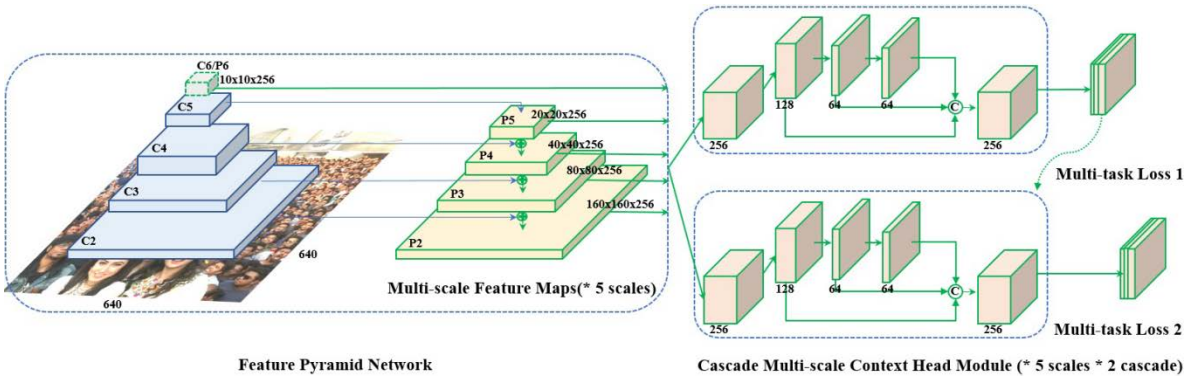

Figure 3. RetinaFace detector.

RetinaFace is based on three main modules made by a feature pyramid network, the context head module, and the cascade multi-task loss. The first module is composed of a pyramid network that digests the input images by computing five different feature maps, each one at a different scale. These are then used by the context head modules to compute, for each of the feature maps, the multi-task loss described

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 December 2022 doi:10.20944/preprints202212.0498.v1

below. The first context head module makes a first bounding box for the anchor, which is fine-tuned by the second context head module to generate a more accurate bounding box.

Feature pyramid levels are computed using ResNet residuals, while the multi-task loss is computed as follows:

$$\mathcal{L} = \mathcal{L}_{cls}(p_i, p_i^*) + \lambda_1 p_i^* \mathcal{L}_{box}(t_i, t_i^*) + \lambda_2 p_i^* \mathcal{L}_{pts}(l_i, l_i^*) + \lambda_3 p_i^* \mathcal{L}_{mesh}(v_i, v_i^*)$$

where p_i is the predicted probability that the i-th anchor is a face, while p_i^* is the ground-truth value (1 if it is a face, 0 otherwise); t_i is a vector with the predicted coordinates for the bounding box of the i-th anchor, and t_i^* is the vector with the coordinates of the real bounding box. The vector l_i contains the predicted coordinates of the five facial landmarks, while l_i^* is the vector with the coordinates for the five ground-truth facial landmarks. The vector, v_i, has the 1068 vertices used for the mesh that represents the 3D face, and v_i^* is the corresponding ground-truth. The variables λ_1, λ_2, and λ_3 are loss-balancing parameters. $\mathcal{L}_{mesh}$is a combination of a vertex loss and an edge loss used to compute a 2D projection of a 3D representation of the face; $\mathcal{L}_{cls}$ is the classification loss for the binary classes of face/not face. $\mathcal{L}_{box}$ is the regression loss for the bounding box, and $\mathcal{L}_{pts}$ is the regression component used to compute the five facial landmarks.

2.3. Filtering steps

As noted in Figure 1, some false positives are removed by applying several filtering approaches that take advantage of the depth maps. Each of these filtering techniques is described below. For some examples of candidate faces that were rejected by the filters, see Figure 4.

Figure 4. Examples of images rejected by the filtering methods.

2.3.1. Filter based on image size (SIZE)

As mentioned above, filtering can be refined even more by evaluating the size of the face region extracted from the depth map as proposed in [37]. Size is initially based on the 2D position and dimension (W_{2D}, h_{2D}) of the pixels in a given candidate face region. This information can then provide an estimate the corresponding 3D physical dimension in mm (W_{3D}, h_{3D}) as follows:

$$W_{3D} = W_{2D} \frac{\bar{d}}{f_x} \quad \text{and} \quad h_{3D} = h_{2D} \frac{\bar{d}}{f_x}, \tag{6}$$

where f_x and f_y are the Kinect camera focal lengths calculated by the calibration algorithm detailed in [34], and $\bar{d}$ is the average depth of the samples in the bounding box of the candidate face. Regions are rejected when they lie outside the fixed range [0.075 cm, 0.45 cm]. Of note is that $\bar{d}$ is the median of the depth samples.

2.3.2. Flatness\unevenness filter (STD)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 December 2022 doi:10.20944/preprints202212.0498.v1

STD [38] extracts flatness and uneveness information from the depth maps : regions with high flatness and uneveness are removed.

STD is a two-step filtering process:

Step 1: segmentation using the depth map is applied;

Step 2: the standard deviation (STD) of the pixels of the depth map belonging to the largest region obtained by the segmentation procedure is calculated from each face candidate region. If the STD lies outside the range of [0.01, 2.00], those regions are rejected.

2.3.3. Segmentation-based filtering (SEG and ELL)

SEG and ELL [1] compare the dimension of the segmented version of the depth image to its bounding box, in the case of SEG, or to its shape, in the case of ELL. In the latter case, the shape should be elliptical. Two evaluations can be made from the information extracted from SEG and ELL. SEG makes possible a comparison of the relative dimension of the larger area to the entire candidate image. Regions where the area of the larger region is less than 40% of the entire area are rejected. ELL can provide the larger region with a fitness score using the Least-Squares criterion to evaluate its closeness to an elliptical model. In this work, the fitness score is calculated with MATLAB's function fit_ellipse [39].

2.3.4. Filtering based on the analysis of the depth values (SEC)

SEC [1] is based on the observation that faces are most commonly located on the top of the body and that the surrounding volume of a face is typically empty. When candidate faces produce a different pattern than expected, it is rejected. To calculate whether the pattern differs from what is expected, the rectangular region defining a candidate face is enlarged so that the depth map surrounding the face can be analyzed. In this work, the expanded region is partitioned into eight radial sectors, each radiating from the center of the candidate face. Figure 4 shows an example. For each sector Sec_i, the number of pixels n_i are counted whose depth value d_p is close to the average depth value of the face $\bar{d}$, as follows:

$$n_i = \left|\{p: |d_p - \overline{\bar{d}}| < t_d \wedge p \in Sec_1\}\right|, \tag{7}$$

where t_d (equals 50cm here) is the measure of closeness.

The number of pixels per sector is averaged on the two lower sectors (Sec_4 and Sec_5) and on the remaining sectors. The ratio between the two averages, n_u and n_l, is calculated as:

$$\frac{n_l}{n_u} = \frac{\frac{1}{2}(n_4+n_5)}{\frac{1}{6}(n_1+n_2+n_3+n_6+n_7+n_8)}. \tag{8}$$

A candidate face is removed if the ratio drops below a threshold, t_r ($t_r = 0.8$ here).

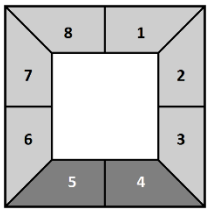

Figure 4. Example of how the expanded neighborhood of a candidate face region can be partitioned into eight sectors (the gray areas). Sectors Sec_4 and Sec_5, depicted in dark gray, should contain the body [1].

3. Results and Discussion

3.1. Datasets

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 December 2022 doi:10.20944/preprints202212.0498.v1

These are the four datasets we used to derive our system: 1) Microsoft Hand Gesture (MHG) [40], 2) Padua Hand Gesture (PHG) [41], 3) Padua FaceDec (PFD) [37], and 4) Padua FaceDec2 (PFD2) [1]. All four captured colored images of faces in unconstrained environments and included their corresponding depth maps. The faces are all frontal and upright and have constrained degrees of rotation. A separate dataset of faces was collected from the Padua FaceDec dataset [37] to run some preliminary tests and tune parameters. As explained in [1], these four datasets were merged to form a challenging dataset for face detection.

Here is a brief description of each dataset. A summary of features is presented in Table 1.

MHG [40] was originally intended for gesture recognition but contains an excellent collection of frontal faces taken from ten people. Forty-two images were selected from MHG for this study and manually labeled with the face position (see [1] for more details).

PHG [41] was also initially intended for gesture recognition and contains face images of ten different people, with each image having only one face. Fifty-nine PHG images were selected from this dataset, and face positions were manually labeled.

PFD [37] was created for face detection and contains 132 labeled images. Photographs were taken in the wild, both indoors and outdoors, with the Kinect 1 sensor. Any given image can have zero or more faces. If an image contains people, they are shown performing various daily activities throughout the day. Thus, lighting conditions vary greatly, and faces are often occluded by multiple degrees.

PFD2 [1] is similar to PFD but is larger and far more challenging. PFD2 contains 316 images captured with the Kinect 2 sensor. A 512×424 depth map and a 1920×1080 color image were obtained for each scene. Some images have zero faces; others show people whose faces are in atypical positions, i.e., with their heads tilted or adjacent to objects. Compared to Kinect 1, the outdoor depth data recorded by Kinect 2 are extremely noisy, making PFD2 an even more challenging dataset. The depth data was projected over the color frame and interpolated to the same resolution. This process produced two aligned depth and color fields.

MERGED contains 549 images with 614 faces, collected from the above datasets as described in [1]. This is the dataset used in the experiments reported in section 3.3. The intention behind this dataset was to form a larger and somewhat more challenging dataset. However, only fully upright frontal faces with less than ±30°rotation were included.

Table 1. Main characteristics of the datasets.

Dataset	Number Images	Number Faces	Depth Resolution	Color Resolution	Difficulty Level
MHG	42	42	640×480	640×480	Low
PHG	59	59	640×480	1280×1024	Low
PFD	132	150	640×480	1280×1024	High
PFD2	316	363	512x424	1920×1080	High
MERGED	549	614	---	---	High

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 December 2022 doi:10.20944/preprints202212.0498.v1

3.2. Performance indicators

Two popular performance indicators are reported:

- **Detection rate (DR)**: this is the ratio of the number of faces correctly detected and the total number of faces in the dataset. Recall that all faces were manually labeled. DR is evaluated at different precision levels in relation to eye distance formally as follows. Let $d_l, (d_r)$ be the Euclidean distance between the manually extracted centered left and right, $C_l(C_r)$ positions and let $C'_l, (C'_r)$ be the detected centered left and right eye positions. The relative error of detection (ED) is defined as $\max(d_l, d_r)/d_{lr}$, where the normalization factor d_{lr} is the Euclidean distance of the expected eye centers meant to provide measurement independent of the scale of the face in the image in relation to the image size. In this work, ED ≤ 0.35 is the value used as a criterion to claim a right eye detection.
- **False positives (FP):** this is the number of candidate faces with no face.

3.3. Experiments

The goal of the first experiment was to compare the detection rates of RetinaFace by adjusting the sensitivity threshold values of s (on the score output of RetinaFace), the default sensitivity threshold value in RetinaFace toolbox is s=0.9, the value of which is provided in parentheses in Table 2. It will be observed that setting the threshold for increasing the detection rate generates more false positives.

The results obtained by RetinaFace are clearly better than previous face detectors based on handcrafted methods or shallow neural networks, see [1] for details. RetinaNet can still be considered a state-of-the-art face detector.

Table 2. Performance of RetinaFace.

Face Detector(s)	DR	FP
RetinaFace (0.02)	95.93	1152
RetinaFace (0.2)	95.93	281
RetinaFace (0.5)	95.93	227
RetinaFace (0.9)	95.60	171
RetinaFace (0.98)	94.79	119

In Table 3, we evaluate the filtering steps, as detailed in section 2.3, along with their combinations on the MERGED dataset.

Table 3. Performance of RetinaFace obtained by combining different filtering steps on MERGED.

	Filter combination	DR	FP
Retina (0.98)	none	94.79	119
	SIZE	94.63	84
	SIZE + SEC	94.14	75
	SIZE + STD + SEG + ELL + SEC	92.67	71
Retina (0.5)	none	95.93	227
	SIZE	95.77	111
	SIZE + SEC	95.11	95
	SIZE + STD + SEG + ELL + SEC	93.65	85

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 December 2022 doi:10.20944/preprints202212.0498.v1

The following conclusions can be drawn from the above table:

- The best trade-off of DR and FP is obtained by RetinaFace (0.5) with the SIZE filter applied. Clearly, SIZE increases the effectiveness of RetinaFace on the test set;
- The other filters reduce the number of FP but also decrease DR;
- Coupling SIZE and SEC is the best approach for minimizing FP without a considerable reduction of DR.

Even though the proposed approach has only been evaluated on a single dataset, we believe it would perform well in real-world conditions, because the MERGED dataset is highly realistic. It contains many images collected from the wild, many of which include multiple frontal faces, not just a single one.

4. Conclusions

In this paper, we combine a state-of-the-art face detector, RetinaFace, with a set of filters generated from the depth map. We demonstrate that the filters reduce the false positives produced by RetinaFace while maximizing the detection rate. Our method for reliable face detection uses information in the depth maps and filters to increase effectiveness, measured as a high detection rate with a lower number of false positives compared to a standalone RetinaFace. This effectiveness was demonstrated on a challenging dataset that was generated by combining frontal images from several datasets with different illumination settings, both indoors and outdoors. Many of the images in this dataset also contained multiple faces, often located in cluttered environments.

Although we use a state-of-the-art face detector, it produced many false positives on our dataset. When a low detection threshold is applied to increase the detection rate, the reported experiments show that the filters based on depth maps are a feasible way to increase the trade-off between detection rate (DR) and false positives (FP) with a state-of-the-art face detector.

Author Contributions: conceptualization, L.N. and A.L.; methodology, L.N.; software, L.N. and A.L..; validation, L.N., S.B. and A.L.; formal analysis, L.N.; investigation, A.L.; resources, S.B.; writing—original draft preparation, A.L. and S.B.; writing—review and editing, S.B.; visualization, S.B. and A.L.

Conflicts of Interest: "The authors declare no conflict of interest."

References

[1] L. Nanni, S. Brahnam, and A. Lumini, "Face detection ensemble with methods using depth information to filter false positives," *Sensors (Switzerland)*, vol. 19, no. 23, p. 23, 2019, doi: 10.3390/s19235242.

[2] X. Zhu, X. Liu, Z. Lei, and S. Z. Li, "Face Alignment in Full Pose Range: A 3D Total Solution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 1, pp. 78–92, 2019, doi: 10.1109/TPAMI.2017.2778152.

[3] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1701–1708, doi: 10.1109/CVPR.2014.220.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 December 2022 doi:10.20944/preprints202212.0498.v1

[4] A. Kumar, A. Kaur, and M. Kumar, "Face detection techniques: a review," *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 927–948, Aug. 2019, doi: 10.1007/s10462-018-9650-2.

[5] M. H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 1, pp. 34–58, 2002, doi: 10.1109/34.982883.

[6] S. Zhang *et al.*, "Shot Scale-Invariant Face Detector. 2017 IEEE International Conference on Computer Vision (ICCV) 2017," pp. 192–201.

[7] B. Jiang, Q. Ren, F. Dai, J. Xiong, J. Yang, and G. Gui, "Multi-task cascaded convolutional neural networks for real-time dynamic face recognition method," *Lect. Notes Electr. Eng.*, vol. 517, pp. 59–66, 2020, doi: 10.1007/978-981-13-6508-9_8/COVER.

[8] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, vol. 1, doi: 10.1109/cvpr.2001.990517.

[9] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *IEEE International Conference on Image Processing*, 2002, vol. 1, pp. 22–25, doi: 10.1109/icip.2002.1038171.

[10] Jain V. and Learned-Miller E., "A Benchmark for Face Detection in Unconstrained Settings. University of Massachusetts; Amherst, MA. – USA: 2010. Technical Report, UMass Amherst Technical Report.," vol. 2010, p. 11.

[11] S. Liao, A. K. Jain, and S. Z. Li, "A Fast and Accurate Unconstrained Face Detector," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 211–223, 2016, doi: 10.1109/TPAMI.2015.2448075.

[12] H. Jin, Q. Liu, H. Lu, and X. Tong, "Face detection using improved LBP under bayesian framework," in *Proceedings - Third International Conference on Image and Graphics*, 2004, pp. 306–309, doi: 10.1109/icig.2004.62.

[13] Q. Zhu, S. Avidan, M. C. Yeh, and K. T. Cheng, "Fast human detection using a cascade of histograms of oriented gradients," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, vol. 2, pp. 1491–1498, doi: 10.1109/CVPR.2006.119.

[14] J. Li and Y. Zhang, "Learning SURF cascade for fast and accurate object detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3468–3475, doi: 10.1109/CVPR.2013.445.

[15] J. Li, T. Wang, and Y. Zhang, "Face detection using SURF cascade," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 2183–2190, doi: 10.1109/ICCVW.2011.6130518.

[16] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool, "Face detection without bells and whistles," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8692 LNCS, no. PART 4, pp. 720–735, 2014, doi: 10.1007/978-3-319-10593-2_47.

[17] M. Nilsson, J. Nordberg, and I. Claesson, "Face detection using local SMQT features and split up snow classifier," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2007, vol. 2, pp. 15–20, doi: 10.1109/ICASSP.2007.366304.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 December 2022 doi:10.20944/preprints202212.0498.v1

[18] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic, "In Robust discriminative response map fitting with constrained local models, CVPR, Portland, OR, 2013; IEEE: Portland, OR," vol. 2013.

[19] N. Markuš, M. Frljak, I. S. Pandžić, J. Ahlberg, and R. Forchheimer, "Fast Localization of Facial Landmark Points," *Croat. Comput. Vis. Work.*, vol. 2014, pp. 39–43, 2014, doi: 10.20532/ccvw.2014.0001.

[20] H. Li, Z. L. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015," pp. 5325–5334.

[21] S. S. Farfade, M. Saberian, and L. J. Li, *Multi-view face detection using Deep convolutional neural networks*. Multi-view face detection using deep convolutional neural networks; Cornell University, 2015.

[22] W. Yang, L. Zhou, T. Li, and H. Wang, "A Face Detection Method Based on Cascade Convolutional Neural Network," *Multimed. Tools Appl.*, vol. 78, no. 17, pp. 24373–24390, 2019, doi: 10.1007/s11042-018-6995-0.

[23] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, 2016, doi: 10.1109/LSP.2016.2603342.

[24] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587, doi: 10.1109/CVPR.2014.81.

[25] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-shot multi-level face localisation in the wild," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5203–5212.

[26] R. Min, N. Kose, J. Dugelay, and A. KinectFaceDB:, "Kinect Database for Face Recognition," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 44, p. 11, 2014.

[27] S. Gupta, K. R. Castleman, M. K. Markey, and A. C. Bovik, "Texas 3D Face Recognition Database," in *Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation*, 2010, pp. 97–100, doi: 10.1109/SSIAI.2010.5483908.

[28] T. C. Faltemier, K. W. Bowyer, and P. J. Flynn, "Using a multi-instance enrollment representation to improve 3D face recognition," *IEEE Conf. Biometrics Theory, Appl. Syst. BTAS'07*, pp. 1–6, 2007, doi: 10.1109/BTAS.2007.4401928.

[29] J. Shotton *et al.*, "Real-time human pose recognition in parts from single depth images," *Commun. ACM*, vol. 56, p. 1, 2013.

[30] R. Mattheij, E. Postma, Y. van den Hurk, and P. Spronck, "Depth-based detection using Haar-like features," in *Belgian/Netherlands Artificial Intelligence Conference*, 2012, vol. 2012, pp. 162–169.

[31] F. Jiang, M. Fischer, H. K. Ekenel, and B. E. Shi, "Combining texture and stereo disparity cues for

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 27 December 2022 doi:10.20944/preprints202212.0498.v1

real-time face detection," *Signal Process. Image Commun.*, vol. 28, p. 9, 2013.

[32] M. Anisetti, V. Bellandi, E. Damiani, L. Arnone, and B. Rat, "A3fd: accurate 3d face detection," in *signal processing for image enhancement and multimedia processing*, I. Signal, Ed. processing for image enhancement and multimedia processing, Damiani, E.; Dipanda, A.; Yetongnon, K.; Legrand, L.; Schelkens, P.; Chbeir, R., Eds. Springer: Boston, MA, 2008, pp. 155–165.

[33] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, *Labeled faces in the wild: a database for studying face recognition in unconstrained environments; University of Massachusetts*. Amherst, 2007.

[34] D. C. Herrera, J. Kannala, J. Heikkil"a, J. Depth, and C. Camera, "Calibration with Distortion Correction," *IEEE Trans Pattern Anal. Mach Intell*, vol. 34, pp. 2058–2064, 2012.

[35] C. D. Mutto, P. Zanuttigh, and G. M. Cortelazzo, "Fusion of Geometry and Color Information for Scene Segmentation," *IEEE J. Sel. Top. Signal Process.*, vol. 6, p. 5, 2012.

[36] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans Pattern Anal. Mach Intell*, vol. 24, p. 5, 2002.

[37] L. Nanni, A. Lumini, F. Dominio, and P. Zanuttigh, "Effective and precise face detection based on both gray-level image and depth map," *Appl. Comput. Informatics*, vol. 10, no. 1–2, pp. 1–13, 2014, doi: 10.1016/j.aci.2014.04.001.

[38] A. Lumini and L. Nanni, "Fair comparison of skin detection approaches on publicly available datasets," *Expert Syst. Appl.*, vol. 160, p. 113677, Dec. 2020, doi: 10.1016/J.ESWA.2020.113677.

[39] Suparyanto dan Rosad (2015, "fit_ellipse - File Exchange - MATLAB Central," *Suparyanto dan Rosad (2015*, 2020. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/3215-fit_ellipse.

[40] Z. Ren, J. Meng, and J. Yuan, "Depth camera based hand gesture recognition and its applications in Human-Computer-Interaction. 2011 8th International Conference on Information," *Commun. Signal Process.*, pp. 1–5, 2011.

[41] F. Dominio, M. Donadeo, and P. Zanuttigh, "Combining multiple depth-based descriptors for hand gesture recognition," *Pattern Recognit Lett*, vol. 50, pp. 101–111, 2014.