applied sciences

Article

Coupling RetinaFace and Depth Information to Filter False Positives

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

[1] Department of Information Engineering (DEI), University of Padova, 35131 Padova, Italy
[2] Department of Information Technology and Cybersecurity, Missouri State University, Springfield, MO 65804, USA
[3] Department of Computer Science and Engineering (DISI), University of Bologna, 47521 Cesena, Italy
[4] Department of Information Engineering (DII), University of Brescia, 25123 Brescia, Italy
[*] 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 an analysis of human behavior. The problem is challenging because of the large variations in facial appearance across different individuals and lighting and pose conditions. One way to detect faces is to utilize a highly advanced face detection method, such as RetinaFace or YOLOv7, 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 method proposed by the authors is tested on a dataset of 549 images, which includes 614 upright frontal faces. The outcomes of the evaluation demonstrate that the method effectively minimizes false positives without compromising the overall detection rate. These findings suggest that incorporating depth information can enhance the accuracy of face detection.

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

check for updates

Citation: Nanni, L.; Brahnam, S.; Lumini, A.; Loreggia, A. Coupling RetinaFace and Depth Information to Filter False Positives. *Appl. Sci.* **2023**, *13*, 2987. https://doi.org/10.3390/app13052987

Academic Editor: Javier Hernando

Received: 30 December 2022
Revised: 21 February 2023
Accepted: 23 February 2023
Published: 25 February 2023

1. Introduction

Face detection algorithms have been developed to automate the process of identifying and locating faces in digital images, driven by the growing demand for such capabilities [1–3]. These algorithms have undergone a series of advancements, beginning with knowledge-based methods that relied on human expertise to define the features of a face, followed by feature-invariant approaches that aimed to recognize faces based on their geometrical properties, such as the relative positioning of the eyes, nose, and mouth.

In response to the challenge of face detection, researchers have developed a variety of techniques utilizing both traditional computer vision methods and more contemporary deep learning methods [4,5]. Traditional methods frequently rely on handcrafted features and ensembles of classifiers to detect faces in images. Although computationally efficient, these methods often encounter difficulties with variations in facial 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 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.

The single-shot detector (SSD) is one of the most renowned face detection systems that employs deep learning [6]. It utilizes a convolutional neural network (CNN) to anticipate bounding boxes and class probabilities for faces in an image. SSDs are celebrated for their ability to deliver high-speed and real-time performance, rendering them ideal for use in 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, 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 opened up new possibilities for applications that rely on the detection and analysis of faces.

In contrast, template matching methods [5] employ a pre-established template of a face to look for corresponding matches in an image. While these methods can be successful, they have certain limitations. Specifically, the template must be meticulously constructed to accommodate for variations in facial appearance, thereby restricting the efficacy of this approach. 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] was a popular and effective method for detecting objects in images. It was specifically designed for object detection and relied on three techniques: an integral image strategy for the efficient extraction of Haar features, an ensemble called AdaBoost, and an attentional cascade structure. One of the main advantages of the Viola–Jones algorithm was its efficiency, which allowed it to run in real-time on standard hardware. This was achieved through Haar-like features, i.e., simple rectangular shapes that can be calculated quickly and easily. Additionally, the boosting algorithm was effective at reducing false positives, a common issue in detection algorithms. Despite its popularity, the Viola–Jones algorithm had certain limitations. It was not always effective at detecting faces in uncontrolled environments, where it has been known to miss face detections [9]. This is due to the limitations of the Haar-like features utilized by the algorithm, which may struggle to accommodate for variations in lighting and facial expression, or other factors that can influence facial appearance.

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. In their research, Markuš et al. [10] integrated a modified version of the Viola–Jones (VJ) method with an algorithm that can detect salient facial landmarks. Meanwhile, Liao et al. [9] introduced a novel feature named scale-invariant NPD and extended the VJ tree classifier to include a deeper quadratic tree structure.

With the advent of deep learning, convolutional neural networks (CNNs) have become increasingly popular for face detection and have shown impressive results in terms of accuracy and speed [11–14]. 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 [15] and Deep Dense Face Detector (DDFD) [12], 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 [16] 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 five-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 multiple 3D sensors and devices available for face detection, each with its own set of benefits and drawbacks. The Kinect [17] is a popular choice due to its affordability and user-friendliness. However, its performance is limited by the resolution of its depth map and its ability to capture only one depth value per pixel. Advanced sensors such as the MU-2 stereo imaging system [18] and the Minolta Vivid 910 range scanner [19] offer higher-resolution depth maps and more precise depth measurements. Nevertheless, these sensors are usually more expensive and complex to operate.

Overall, the use of 3D information in face detection can significantly improve 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 a more widespread adoption of 3D face detection techniques in various of applications.

Various approaches have been used for face detection using depth images, each with their own strengths and limitations. Some methods, such as [20], rely on comparing pixels in depth images to classify body joints and parts for pose recognition, while others, such as [21], compare square regions for face detection. In [3], a deep-learned approach for 2D images (DeepFace) is combined with a 3D-model-based alignment, which was effective in detecting faces in unconstrained environments. Anisetti et al. [22] used a coarse detection method and a 3D morphable face model for locating faces. Nanni et al. [1] proposed different filtering steps to be applied on information in the Kinect depth map to address the issue of increased false positives when combining different face detectors, they also shared a challenging dataset that contains depth and 2D images, which is used to validate the best-performing system developed in this work. The dataset has 549 samples, including 614 upright frontal faces. The filtering steps adopted in the system decrease the amount 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 (accessed on 30 December 2022).

2. Materials and Methods

Figure 1 depicts the face detection system that is developed in this work. First, face detection with the face detector RetinaFace is performed on a raw color image (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). By calibrating the color and depth information it is possible to accomplish the alignment, as explained in [23]. Briefly, the depth sample's positions in 3D space are computed through the intrinsic camera parameters of principal point and focal length of the depth camera. These intrinsic parameters, along with the extrinsic parameters of the camera pair system, are then projected onto 2D space. Next, depth values and color are combined with each sample, as described in 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. This last process is described in Section 2.3.

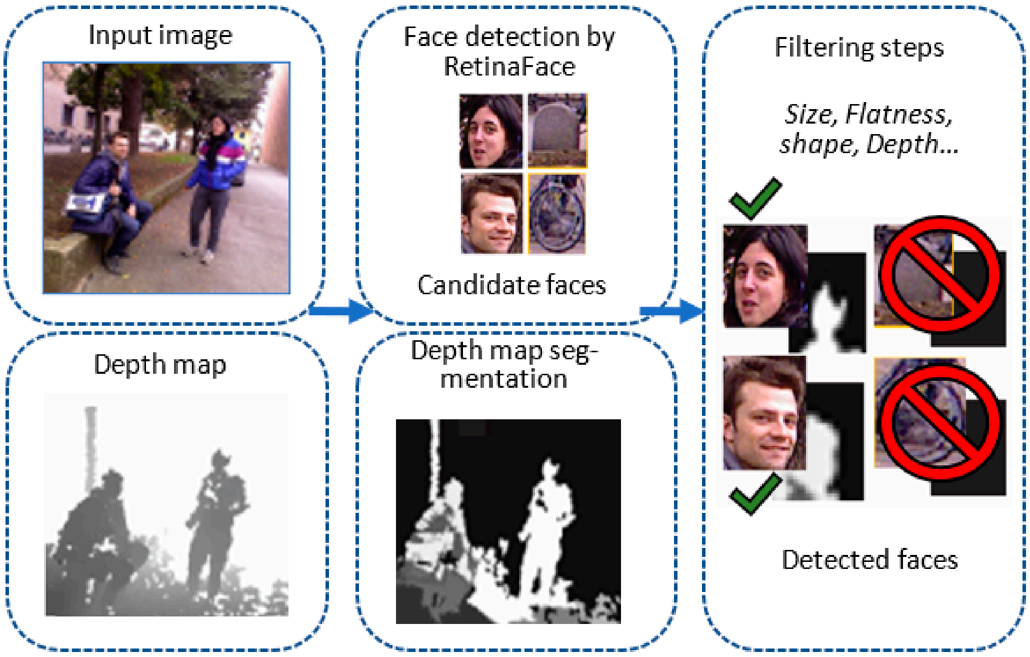

Figure 1. A general illustration of the proposed face detection approach. First, the input image undergoes a first round of detection RetinaFace (**top**). Second, segmentation is performed using the depth map of the image **(bottom)**. Finally, the candidate faces undergo a filtering process to select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

2.1. Depth Map Alignment and Segmentation

Depth maps and the raw color images are segmented in tandem, similar to the method in [24]. The procedure involves transforming each sample into a six-dimensional vector and then clustering the point set using the mean shift algorithm [25].

Transformation into a six-dimensional vector is accomplished as follows. Samples in the Kinetic depth map are 3D points: p_i, $i = 1, \ldots, N$, where N is the number of points. As described in [23], color cameras and the joint calibration of the depth facilitates reprojecting the depth samples. This involves mapping them onto the corresponding pixels in the color image. This process allows each point to be connected to its 3D spatial coordinates (x, y, and z), as well as the RGB color values. However, these representations cannot be directly compared because they exist in completely different spaces. In order to use the mean shift clustering algorithm to extract multidimensional vectors, it is necessary to make all components comparable. To make the color values comparable, they are converted to the CIELAB uniform color space, so colors are represented in a 3D space with lightness (L) values ranging from black (0) to white (100), (a) from green (-) to red (+), and (b) from blue (-) to yellow (+). The purpose of this conversion is to use the Euclidean distance between the color vectors in the mean shift algorithm.

More formally, the algorithm works by considering each scene point's color information in the CIELAB color space, c, as a 3D vector:

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

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

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

In order to ensure that the scene segmentation algorithm is not affected by the relative scaling of the point-cloud geometry and to ensure that the color distances and geometry are consistent, all components of p_i^g are normalized based on the average standard deviation of the coordinates of the point in three dimensions $\sigma_g = (\sigma_x + \sigma_y + \sigma_z)/3$, which results in the following vector:

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

After normalizing the color information vectors, the final color representation is obtained by taking the average of the standard deviations of the L, a, and b color components:

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

After the normalization of the vectors of the color information and geometry, they are combined to produce the final representation f:

$$p_i^f = \begin{bmatrix} \overline{\mathrm{L}}(p_i) \\ \overline{\mathrm{a}}(p_i) \\ \overline{\mathrm{b}}(p_i) \\ \lambda_{\overline{\mathrm{x}}} \\ \lambda_{\overline{\mathrm{y}}} \\ \lambda_{\overline{\mathrm{z}}} \end{bmatrix} \tag{5}$$

The relative weight of color and geometry in the final segmentation can be controlled using the parameter λ. A lower value of λ assigns more importance to color information, while a higher value of λ emphasizes geometry. For more information on how to automatically determine the optimal value for λ, refer to [24].

The vectors p_i^f are clustered with the mean shift algorithm [25] to segment the sampled scene. Further refinement is possible by removing regions smaller than some thresholds 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.

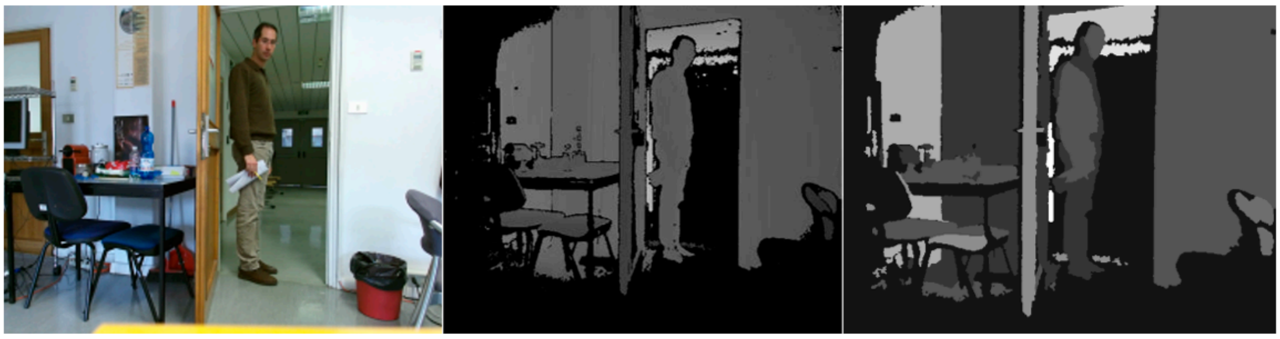

Figure 2. Raw color image (**left**), depth map (**middle**), and segmentation map (**right**).

2.2. Face Detectors: RetinaFace and YOLOv7

RetinaFace [16] 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 face detector system is able to perform multiple tasks, including predicting a face score, identifying the bounding box around a face, identifying five facial landmarks, and determining the 3D position and correspondence of each pixel on the face. The system's overall structure is depicted in Figure 3.

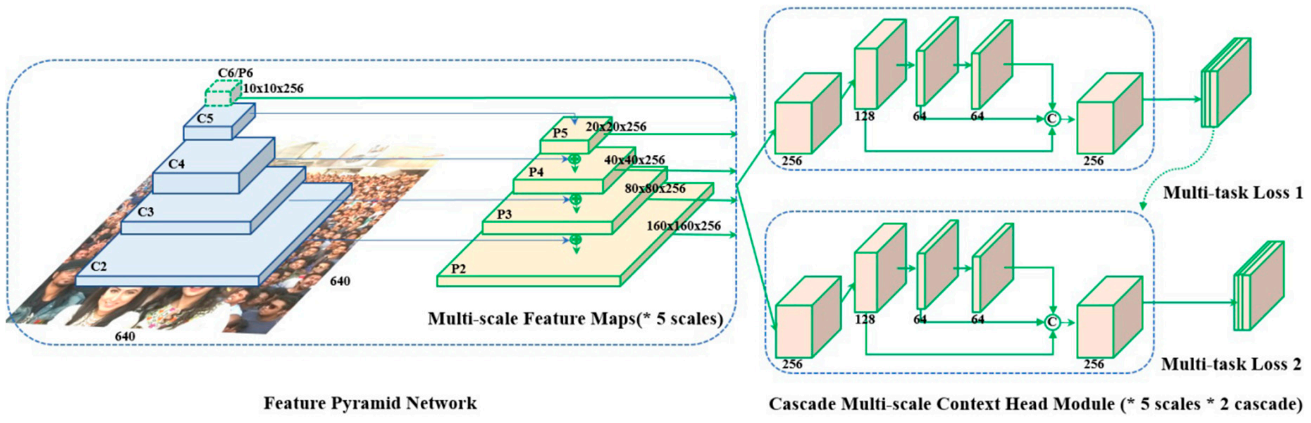

Figure 3. RetinaFace detector.

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 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:

$$\mathcal{L} = \mathcal{L}_{cls}(p_i, p_i^*) + \lambda_1 \, p_i^* \mathcal{L}_{box}(t_i, t_i^*) + \lambda_2 \, p_i^* \mathcal{L}_{pts}(l_i, l_i^*) + \lambda_3 \, p_i^* \mathcal{L}_{mesh}(v_i, v_i^*) \tag{6}$$

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.

YOLOv7 is the latest and fastest single-stage real-time object detector in the YOLO family, introduced in July 2022 [26]. It is a fully convolutional neural network (FCNN)-based object detector with three components: Backbone, Neck, and Head. The Backbone extracts features, the Neck collects them, and the Head consists of output layers for final detection. YOLO models are trained to predict the locations and classes of objects with bounding boxes and use non-maximum suppression (NMS) for post-processing. We used YOLO as face detector according to following library (https://github.com/hpc203/yolov7-detect-face-onnxrun-cpp-py, accessed on 30 December 2022).

2.3. Filtering Steps

By leveraging on depth maps it is possible to apply a filtering approach with the aim of removing some false positives, as noted in Figure 4. 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 previously mentioned, the effectiveness of the filtering process can be further improved by considering the size of the face region identified in the depth map, as suggested in [27]. Starting from the estimation of dimension (W_{2D}, h_{2D}) of the candidate face in the 2D image, the corresponding 3D physical dimensions in millimeters (W_{3D}, h_{3D}) can be estimated using the following approach:

$$W_{3D} = W_{2D}\frac{\overline{d}}{f_x} \text{ and } h_{3D} = h_{2D}\frac{\overline{d}}{f_y}, \tag{7}$$

In this equation, f_x and f_y are the Kinect camera's focal lengths, which are calculated using the calibration algorithm described in [23], and $\overline{d}$ is the average depth of the samples within the bounding box of the candidate face region. The estimated 3D physical dimensions (W_{3D}, h_{3D}) are only accepted if each of them falls within the fixed range of [7.5 cm, 45 cm]. It is worth noting that $\overline{d}$ is the median of the depth samples.

2.3.2. Flatness\Unevenness Filter (STD)

STD [28] extracts flatness and unevenness information from the depth maps: regions with high flatness and unevenness are removed.

STD is a two-step filtering process:

Step 1: Segmentation using the depth map is applied;

Step 2: For each face candidate region, the standard deviation (STD) of the pixels in the depth map that belong to the largest region identified by the segmentation procedure is calculated. If the STD falls outside the range of [0.01, 2.00], that region is rejected.

2.3.3. Segmentation-Based Filtering (SEG and ELL)

SEG and ELL [1] both use different approaches to compare the dimensions of the segmented version of the depth image to a reference shape. In the case of SEG, the comparison is made to the bounding box of the image, while in the case of ELL, the reference shape is an ellipse. These evaluations allow for the comparison of the relative size of the largest area to the entire candidate image in the case of SEG, and for the assessment of the shape (i.e., whether it is elliptical) in the case of ELL (a least-squares criterion is adopted to evaluate the similarity of a region to an elliptical model). Regions that are smaller than 40% of the total area are not considered in the analysis. The score is computed with a “fit_ellipse” function.

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 that defines a candidate face is enlarged to include the surrounding depth map for further analysis. In this work, the expanded region is partitioned into two regions, R_U and R_D (see Figure 5). For each region R, a number of pixels n_R are counted whose depth value d_p is close to the average depth value of the face $\overline{d}$, as follows:

$$n_R = \left|\left\{p : \left|d_p - \overline{d}\right| < t_d \wedge p \in R\right\}\right| \quad (8)$$

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

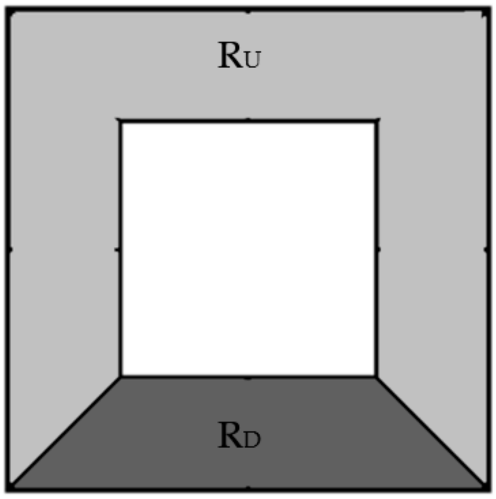

Figure 5. Example of how the expanded neighborhood of a candidate face region can be partitioned into two regions R_U and $R_{D.}$ Region $R_{D,}$ depicted in dark gray, should contain the body.

The ratio between the two regions, $3{\cdot}n_D/n_U$, is calculated, and a candidate face will be removed if the ratio drops below a specified threshold, t_r ($t_r = 0.8$ here).

3. Results and Discussion

3.1. Datasets

In our study, we utilized the dataset MERGED proposed in [1], which is obtained by combining the following four datasets: Microsoft Hand Gesture (MHG) dataset, Padua Hand Gesture (PHG) dataset, Padua FaceDec (PFD) dataset, and Padua FaceDec2 (PFD2) dataset. All of these datasets contain colored images of faces captured in unconstrained environments along with their corresponding depth maps. The faces in these datasets are frontal and upright with limited degrees of rotation (with less than ±30°). A summary of features is presented in Table 1.

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	512 × 424	1920 × 1080	High
MERGED	549	614	—	—	High

3.2. Performance Indicators

We report two performance indicators:

- **Detection rate (DR)**: this metric measures the accuracy of the face detection algorithm by comparing the number of faces correctly detected to the total amount of faces in the dataset. All faces in the dataset have been manually labeled for this evaluation. Let d_l, (d_r) be the Euclidean distance between the manually extracted $C_l(C_r)$ positions that are centered left and right, 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 d_{lr} is the Euclidean distance between the expected eye centers. 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 correctly extracted (i.e., incorrect eye detection, having ED > 0.35).

3.3. Experiments

The goal of the first experiment (Table 2) was to compare the detection rates of RetinaFace (https://github.com/elliottzheng/face-detection, accessed on 30 December 2022) and YOLO by adjusting the sensitivity threshold (on the score output of the detector). It will be observed that setting the threshold for increasing the detection rate generates more false positives.

Table 2. Performance (detection rate and false positives) of RetinaFace and YOLOv7.

Face Detector:	**RetinaFace**		**YOLOv7Tiny**		**YOLOv7Lite**	
Threshold	**DR**	**FP**	**DR**	**FP**	**DR**	**FP**
0.02	95.93	1152	83.06	909	73.94	1491
0.2	95.93	281	82.74	405	73.78	427
0.5	95.93	227	82.41	309	73.45	305
0.9	95.60	171	0.16	0	0	0
0.98	94.79	119	0	0	0	0

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

The second experiment aims to assess the performance of the filtering steps and their combinations. Table 3 presents the results of combining the three face detectors with the filtering steps.

Table 3. Performance of face detection methods combined by filtering steps.

Face Detector:		**RetinaFace**		**YOLOv7Tiny**		**YOLOv7Lite**	
Threshold	**Filter**	**DR**	**FP**	**DR**	**FP**	**DR**	**FP**
0.98	None	94.79	119	0	0	0	0
	SIZE	94.63	84	0	0	0	0
	SIZE + SEC	94.14	75	0	0	0	0
	all	92.67	71	0	0	0	0
0.5	None	95.93	227	82.41	309	73.45	305
	SIZE	95.77	111	82.08	199	73.29	230
	SIZE + SEC	95.11	95	81.60	180	72.96	213
	all	93.65	85	79.80	162	71.01	194
0.2	None	95.93	281	82.74	405	73.78	427
	SIZE	95.77	128	82.41	229	73.62	283
	SIZE + SEC	95.11	109	81.92	206	73.29	254
	all	93.65	94	80.13	179	71.34	220

It should be noted that the complexity of the three face detectors is very different: YOLOv7lite has a storage requirement of 3 MB, YOLOv7tiny has a requirement of 31 MB, and RetinaFace (with ResNet backbone) requires 107 MB. The detection time of YOLOv7tiny is 38% higher than YOLOv7lite, and RetinaFace is 243% higher than YOLOv7lite.

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;
- SIZE permits to reduce FP without a considerable reduction in DR, even when coupled with YOLOv7.

Obviously, the advantage of using depth map information is more pronounced when the effectiveness of the visual sensor is reduced due to darkness, fog, or other reasons that prevent a perfect view of the scene. To highlight this advantage, an additional test was performed in which the visual spectrum images were subjected to a fog filter. The effect was achieved using Gimp's foggify filter (https://docs.gimp.org/2.8/en/python-fu-foggify.html, accessed on 30 December 2022) with turbulence = 4.0 and opacity 100.

Figure 6 shows an example of perturbed images, while Table 4 shows the results of the same experiment in Table 3 applied to the perturbed images.

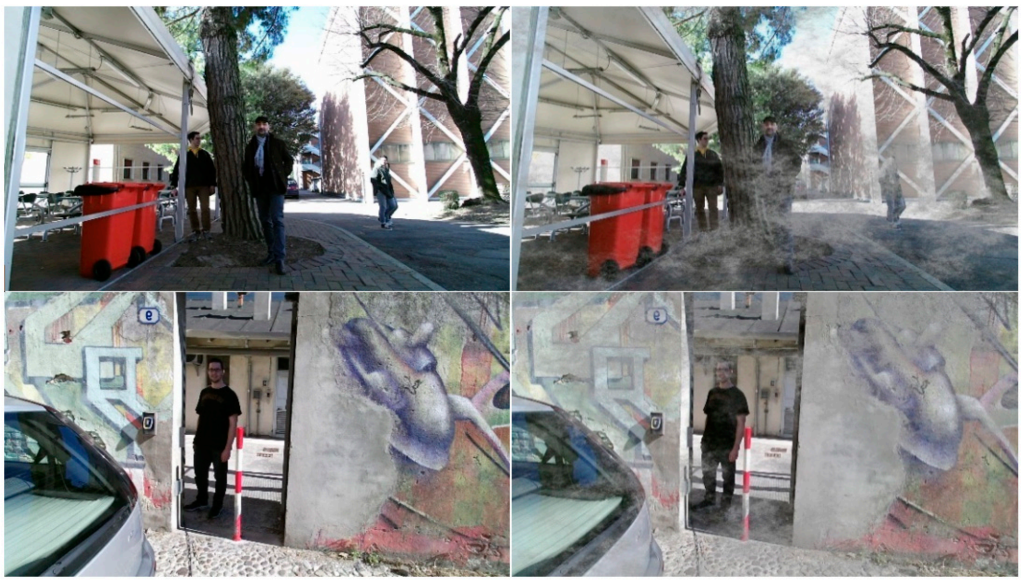
Figure 6. Original images (**left**), fog-perturbed images (**right**).

In Figure 7 a plot of detection rate vs. number of false positives is reported (by varying the acceptance threshold) for each of the considered approaches: in the first row, results on the original dataset are shown, while the second row reports results from the fog-perturbed dataset. The usefulness of filtering is more evident when considering regions that do not contain faces. Therefore, we define a strong false positive (SFP) as a candidate face that has an ED score greater than 1. In Figure 8, a plot of the detection rate versus the number of strong false positives is presented.

Table 4. Performance of face detection methods combined by filtering steps on the fog-perturbed images.

Face Detector:		RetinaFace		YOLOv7Tiny		YOLOv7Lite	
Threshold	**Filter**	**DR**	**FP**	**DR**	**FP**	**DR**	**FP**
0.98	none	79.15	93	0	0	0	0
	SIZE	78.99	66	0	0	0	0
	SIZE + SEC	78.66	63	0	0	0	0
	all	77.36	59	0	0	0	0
0.5	none	87.46	219	69.54	264	59.77	274
	SIZE	87.30	132	69.22	186	59.61	217
	SIZE + SEC	86.64	118	68.73	172	59.61	201
	all	85.34	101	67.26	153	57.98	189
0.2	none	88.64	321	72.15	363	62.54	398
	SIZE	88.27	171	71.82	233	62.38	289
	SIZE + SEC	87.62	149	71.34	211	62.38	257
	all	86.32	123	69.71	180	60.59	234

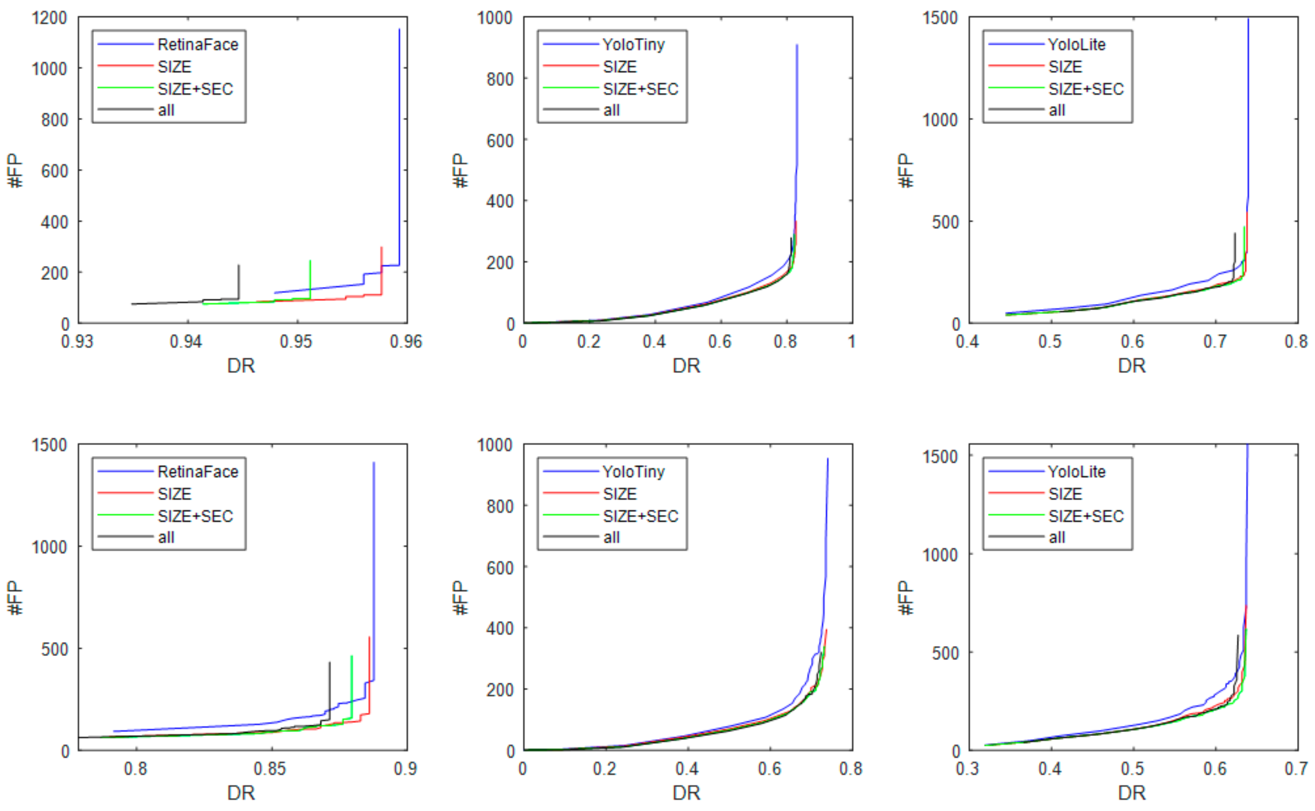

Figure 7. Plot of the detection rate vs. number of false positives for the 3 face detectors (RetinaFace, YOLOv7Tiny, YOLOv7Lite). First row: original dataset, second row: fog-perturbed dataset.

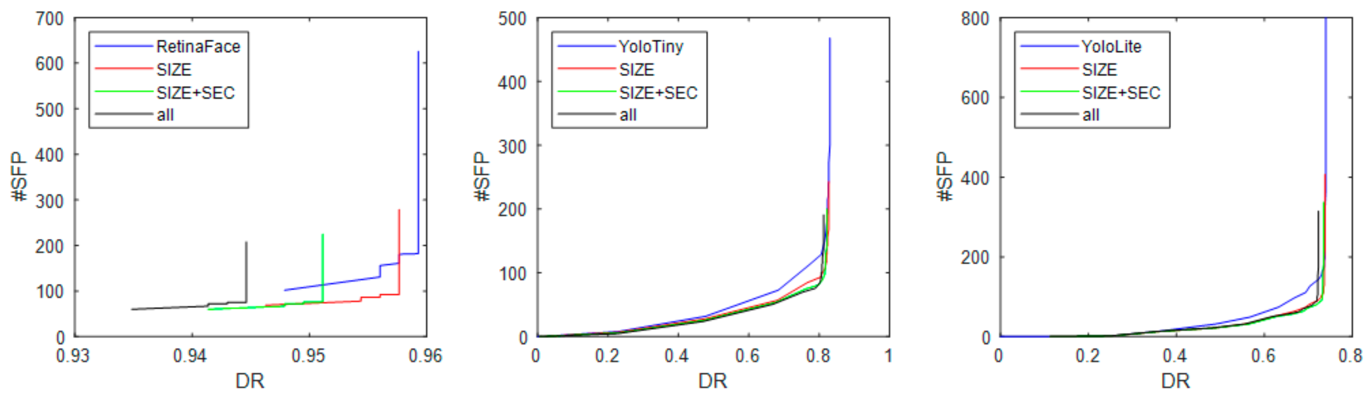

Figure 8. Plot of the detection rate vs. number of string false positives for the 3 face detectors (RetinaFace, YOLOv7Tiny, YOLOv7Lite) on the original dataset.

According to the plots in Figure 7 it is clear that the best trade-off between the detection rate and the number of false positives is gained by coupling the face detector with the filtering method named "size". It happens for all the approaches tested in this paper, but undoubtedly this is more interesting for YOLO since the performance is lower.

The advantages of all the filtering approaches are more evident in the presence of strong false positives, as shown in Figure 8.

As for the choice of basic face detector, obviously it must be dictated by the system requirements: YOLO is faster, but less accurate, while RetinaFace is very accurate but requires more resources and a longer computation time. It is important to note that the most suited application of our idea will be in AR/VR headsets, mainly in portable devices where lightweight networks will be used, at least for the next few years.

Although the proposed approach has only been evaluated using a single dataset, it is believed to be effective in real-world conditions because the MERGED dataset is highly realistic. It includes many images taken in natural settings and features multiple frontal faces, not just a single one, making it more representative of real-world conditions. Moreover, the fog-perturbed version of the dataset makes the benchmark even more difficult.

4. Conclusions

We use two state-of-the-art face detectors called RetinaFace and YOLOv7 and apply a set of filters based on the depth map to improve the accuracy of the detection. We demonstrate that the filters reduce the false positives produced by base method 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 with a standalone face detector working on visual spectrum images. This effectiveness was demonstrated on two challenging datasets: the first is the combination of several datasets, including images with different illumination settings, both indoors and outdoors, the second is a fog-perturbed version of the first. Many of the images contain multiple faces, often located in cluttered environments.

Although we used 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 several state-of-the-art face detectors.

Though it may seem that using a 3D camera is an unnecessary cost for the face detection problem, it must be considered that sensors with a depth map function will be widespread and inexpensive in the near future.

Author Contributions: Conceptualization, L.N. and A.L. (Alessandra Lumini); methodology, L.N.; software, L.N. and A.L. (Alessandra Lumini); validation, L.N., S.B. and A.L. (Alessandra Lumini); formal analysis, L.N.; resources, S.B.; investigation, A.L. (Andrea Loreggia); writing—original draft preparation, A.L. (Andrea Loreggia) and S.B.; writing—review and editing, A.L. (Alessandra Lumini); visualization, A.L. (Andrea Loreggia) and S.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

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

References

1. Nanni, L.; Brahnam, S.; Lumini, A. Face Detection Ensemble with Methods Using Depth Information to Filter False Positives. *Sensors* **2019**, *19*, 5242. [CrossRef] [PubMed]
2. Zhu, X.; Liu, X.; Lei, Z.; Li, S.Z. Face Alignment in Full Pose Range: A 3D Total Solution. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 78–92. [CrossRef] [PubMed]
3. Taigman, Y.; Yang, M.; Ranzato, M.; Wolf, L. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1701–1708.
4. Kumar, A.; Kaur, A.; Kumar, M. Face detection techniques: A review. *Artif. Intell. Rev.* **2019**, *52*, 927–948. [CrossRef]
5. Yang, M.-H.; Kriegman, D.; Ahuja, N. Detecting faces in images: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 34–58. [CrossRef]
6. Zhang, S.; Zhu, X.; Lei, Z.; Shi, H.; Wang, X.; Li, S.Z. S3fd: Shot Scale-Invariant Face Detector. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), 22–29 October 2017; pp. 192–201.
7. Jiang, B.; Ren, Q.; Dai, F.; Xiong, J.; Yang, J.; Gui, G. Multi-task Cascaded Convolutional Neural Networks for Real-Time Dynamic Face Recognition Method. *Lect. Notes Electr. Eng.* **2020**, *517*, 59–66. [CrossRef]
8. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR, Kauai, HI, USA, 8–14 December 2001.
9. Liao, S.; Jain, A.K.; Li, S.Z. A Fast and Accurate Unconstrained Face Detector. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 211–223. [CrossRef] [PubMed]
10. Markuš, N.; Frljak, M.; Pandžić, I.S.; Ahlberg, J.; Forchheimer, R. Fast Localization of Facial Landmark Points. *arXiv* **2014**, arXiv:1403.6888. [CrossRef]
11. Li, H.; Lin, Z.; Shen, X.; Brandt, J.; Hua, G. A convolutional neural network cascade for face detection. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 5325–5334.
12. Farfade, S.S.; Saberian, M.; Li, L.J. Multi-view face detection using Deep convolutional neural networks. In Proceedings of the 5th ACM on International Conference on Multimedia Retrieva, Shanghai, China, 23–26 June 2015.
13. Yang, W.; Zhou, L.; Li, T.; Wang, H. A Face Detection Method Based on Cascade Convolutional Neural Network. *Multimed. Tools Appl.* **2019**, *78*, 24373–24390. [CrossRef]
14. Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Process. Lett.* **2016**, *23*, 1499–1503. [CrossRef]
15. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
16. Deng, J.; Guo, J.; Ververas, E.; Kotsia, I.; Zafeiriou, S. Retinaface: Single-shot multi-level face localisation in the wild. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, Seattle, WA, USA, 14–19 June 2020; pp. 5203–5212.
17. Min, R.; Kose, N.; Dugelay, J.L. Kinectfacedb: A kinect database for face recognition. *IEEE Trans. Syst. Man Cybern. Syst.* **2014**, *44*, 1534–1548. [CrossRef]
18. Gupta, S.; Castleman, K.R.; Markey, M.K.; Bovik, A.C. Texas 3D Face Recognition Database. In Proceedings of the 2010 IEEE Southwest Symposium on Image Analysis & Interpretation (SSIAI), Austin, TX, USA, 23–25 May 2010; pp. 97–100.
19. Faltemier, T.C.; Bowyer, K.W.; Flynn, P.J. Using a Multi-Instance Enrollment Representation to Improve 3D Face Recognition. In Proceedings of the 2007 First IEEE International Conference on Biometrics: Theory, Applications, and Systems, Crystal City, VA, USA, 27–29 September 2007; pp. 1–6.
20. Shotton, J.; Sharp, T.; Kipman, A.; Fitzgibbon, A.; Finocchio, M.; Blake, A.; Cook, M.; Moore, R. Real-time human pose recognition in parts from single depth images. *Commun. ACM* **2013**, *56*, 116–124. [CrossRef]
21. Mattheij, R.; Postma, E.; Van den Hurk, Y.; Spronck, P. Depth-based detection using Haar-like features. In Proceedings of the Belgian/Netherlands Artificial Intelligence Conference, Maastricht, The Netherlands, 25–26 October 2012; pp. 162–169.

22. Anisetti, M.; Bellandi, V.; Damiani, E.; Arnone, L.; Rat, B. A3fd: Accurate 3d face detection. In *Signal Processing for Image Enhancement and Multimedia Processing*; Signal, I., Ed.; Springer: Boston, MA, USA, 2008; pp. 155–165.
23. Herrera, D.; Kannala, J.; Heikkilä, J. Joint Depth and Color Camera Calibration with Distortion Correction. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2058–2064. [CrossRef] [PubMed]
24. Mutto, C.D.; Zanuttigh, P.; Cortelazzo, G.M. Fusion of Geometry and Color Information for Scene Segmentation. *IEEE J. Sel. Top. Signal Process.* **2012**, *6*, 505–521. [CrossRef]
25. Comaniciu, D.; Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 603–619. [CrossRef]
26. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696. [CrossRef]
27. Nanni, L.; Lumini, A.; Dominio, F.; Zanuttigh, P. Effective and precise face detection based on color and depth data. *Appl. Comput. Inform.* **2014**, *10*, 1–13. [CrossRef]
28. Lumini, A.; Nanni, L. Fair comparison of skin detection approaches on publicly available datasets. *Expert Syst. Appl.* **2020**, *160*, 113677. [CrossRef]