2023 IEEE 13th International Conference on Pattern Recognition Systems (ICPRS) | 979-8-3503-3337-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICPRS58416.2023.10179079

Deep Learning-based Human Height Estimation from a Stereo Vision System

Henry O. Velesaca, Jorge Vulgarin, Boris X. Vintimilla
Escuela Superior Politécnica del Litoral, ESPOL,
Facultad de Ingeniería en Electricidad y Computación, CIDIS,
Campus Gustavo Galindo, 09-01-5863, Guayaquil, Ecuador
{hvelesac,jvulgari,boris.vintimilla}@espol.edu.ec

***Abstract*—This paper presents a deep learning-based human height estimation approach using a stereo vision system, which is part of a smart receptionist framework. The proposal consists of a smart screen with an integrated webcam and an additional webcam. The workflow of this approach initially acquires and processes the images in real-time, using deep neural networks detects a human in the images, aligns the images from the two cameras, then obtains the depth of the camera, and finally computes the height of the person under study. The proposed solution is evaluated by different people to check the effectiveness of the system. Obtained results show MAE below 1.0 cm in the computed heights, it is also compared with other techniques of the state-of-the-art.**

***Index Terms*—human height estimation, stereo vision system, deep learning, computer vision approaches, smart receptionist**

I. Introduction

The estimation of height is an important problem in many computer vision applications, such as human pose estimation (e.g., [1], [2]), visual surveillance (e.g., [3], [4]), and sports analytics. In recent years, several approaches have been proposed for measuring human height using vision-based techniques. However, the existing methods often require a large number of labeled images or a long training process to obtain high performance. Furthermore, most of these methods are designed for a monocular camera, which limits their applicability to environments where the viewpoint changes rapidly [5]. Human height estimation using computer vision is a challenging task that involves estimating the height of a person from an image or video using computer vision techniques. Human height estimation has a wide range of applications, including surveillance, biometrics, medical imaging, and sports analytics [6], [7].

Traditional approaches to human height estimation use manual measurements, which are time-consuming, labor-intensive, and prone to errors. Computer vision-based approaches to human height estimation have the potential to overcome these limitations by automating the process and providing accurate and reliable measurements. Computer vision-based human height estimation involves extracting features from the image or video that are related to human height. These features may include body proportions, relative size, and contextual information. Various computer vision techniques can be used for feature extraction, such as edge detection, segmentation, and pattern recognition. Once the features are extracted, these are used to estimate the height of the person. This can be done using regression-based techniques, where a regression model is trained on a dataset of images and corresponding height measurements. The regression model takes the extracted features as input and outputs the estimated height of the person [8]. Deep learning techniques, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have shown promising results for human height estimation in the last decades. These techniques involve training a neural network on a large dataset of images and corresponding height measurements, allowing the network to learn the relationship between the image features and the height of the person. The performance of human height estimation using computer vision is evaluated using metrics such as mean absolute error (MAE) and root mean squared error (RMSE), which measure the difference between the estimated height and the ground truth height [9]. In conclusion, human height estimation using computer vision is an important and challenging task with a wide range of applications. Traditional manual measurements are labor-intensive and prone to errors, while computer vision-based approaches have the potential to automate the process and provide accurate and reliable measurements. Various computer vision techniques can be used for feature extraction, and deep learning techniques have shown promising results for human height estimation.

In the present work, the problem of estimating human height from pairs of images captured from a stereo camera system that is part of a smart receptionist framework is proposed. The proposed approach consists of five stages: image acquisition, human detection, feature extraction, deep estimation, and height estimation. In the first stage, a deep learning network is used to perform the task of people detection and face feature extraction. In the second stage, with the characteristics obtained in the previous step and the camera calibration parameters, the distance between the camera and the person is estimated and finally the person's height estimate is obtained.

The manuscript is organized as follows. Section II presents works related to person detection and height estimation prob-

979-8-3503-3337-4/23/$31.00 ©2023 IEEE

lems. Section III presents the approach proposed for human height estimation, together with a summary of the dataset generated for the current work. Experimental results and comparisons with different approaches are given in Section IV. Finally, conclusions are presented in Section V.

II. Related Works

As described above, this paper presents an approach to performing human height estimation. In this section, state-of-the-art algorithms in these topics are reviewed. Firstly, the most relevant object detection techniques approach (e.g., human detection) especially deep learning based, are summarized; secondly, human height estimation techniques are presented.

A. Object Detection Techniques

Object detection is a computer vision technique that involves identifying the location of objects within an image or video. Human height estimation is one application of object detection that involves detecting the location of a person in an image or video and estimating their height. One approach involves using stereo vision techniques to estimate the distance between the camera and the person, and then estimate the person's height. This approach requires multiple cameras to capture different views of the scene and compute the distance using triangulation [10]. The use of stereo vision and triangulation techniques requires multiple cameras, complex calibration procedures, and sophisticated algorithms to perform accurate height estimates. This can make the system difficult and expensive to implement, particularly for applications that require real-time processing. Other techniques for human height estimation using object detection may involve using features such as shadow length or perspective distortion to estimate the height of a person within an image or video. This technique may not be applicable in situations where the person's body points cannot be easily detected or are occluded. Furthermore, it can be a time-consuming and challenging process, particularly for real-time applications. Overall, object detection techniques for human height estimation can be used in a wide range of applications, including surveillance systems, sports analytics, and medical imaging [6], [7].

Another approach involves using deep learning models, such as convolutional neural networks (CNNs), to detect the presence of a person in an image and estimate their height based on various features such as the size and shape of their body [9]. One of the disadvantages of this approach is the large training data requirements and the computational complexity that requires the use of the GPU, which can limit the use in real-time applications. One of the popular object detection techniques that can detect and classify objects in real-time is YOLO (You Only Look Once) [11]. One application of object detection using YOLO is for human height estimation. This task involves detecting and localizing a person in an image or video and estimating their height. YOLO can have a large model size, which can be a limitation for resource-constrained environments, such as embedded systems and real-time applications. On the other hand, Mediapipe is an open-source cross-platform framework developed by Google that offers a wide range of computer vision and machine learning solutions, including object detection. The framework is designed to enable the rapid prototyping of various computer vision applications, including those related to human height estimation [12]. Although Mediapipe is optimized to run on CPU and presents good performance in the people detection task, it is also true that it presents an important limitation and that is that it can only detect one person at a time. Another popular approach for human height estimation using object detection is the Single Shot Multibox Detector (SSD) algorithm. SSD is a deep learning-based object detection technique that uses a convolutional neural network (CNN) to generate a set of bounding boxes and associated class probabilities for objects within an image [13]. Similar to YOLO, SSD can have a large model size and be limited for systems using GPUs, limiting use in real-time applications.

The approaches presented above provide pre-trained object detection models that can detect people in images or video. Once a person is detected, additional techniques can be used to estimate their height. Overall, object detection techniques for human height estimation can be a useful tool for a variety of applications, such as in security systems, sports analytics, and health monitoring.

B. Human Height Estimation Techniques

There are several techniques for estimating a person's height using machine vision, a review of these techniques is presented below.

The first technique "*body portions analysis*", is based on the idea that certain body proportions are relatively constant among individuals, regardless of their actual height. The most commonly used body proportions for height estimation are the head-to-trunk ratio and the leg-to-trunk ratio. The head-to-trunk ratio is the ratio of the length of the head to the length of the trunk, and the leg-to-trunk ratio is the ratio of the length of the legs to the length of the trunk. To use this technique, the algorithm must first detect and locate the head, trunk, and legs in the image. Once these body parts are identified, their lengths can be measured. The algorithm then uses the measured lengths to calculate the head-to-trunk and leg-to-trunk ratios. The algorithm can then use these ratios to estimate the height of the person by using a database of known head-to-trunk and leg-to-trunk ratios for different heights and then finding the closest match in the database to the ratios calculated from the image [14], [15], [16]. Below the equations used to calculate height using this technique are shown.

$$head\_trunk\_ratio = \frac{head\_length}{trunk\_length} \tag{1}$$

$$leg\_trunk\_ratio = \frac{legs\_length}{trunk\_length} \tag{2}$$

The second technique "*pattern analysis*" is based on the identification and analysis of known patterns in the image to estimate the person's height. This technique is based on the idea that certain body patterns, such as the position of legs and arms, are relatively constant across individuals, regardless of their actual height. To use this technique, the algorithm must first detect and locate the legs and arms in the image. Once these body parts are identified, the algorithm can analyze their position and orientation to estimate the height of the person. One possible approach is to use the position of the joints (knees, hips, shoulders, elbows) to estimate the height of the person. The algorithm could use the relative positions of these joints to calculate the length of the legs and arms, and then use this information to estimate the height of the person. Another approach is to use the position of the feet and head to estimate the height. This can be done by assuming that the feet and head are at the bottom and top of the person respectively, then the algorithm can use the position of these two points to calculate the height of the person [17], [18]. Below the equations used to calculate height using this technique are shown.

$$height = legs\_length + trunk\_length \tag{3}$$

$$height = feet\_position - head\_position \tag{4}$$

The next technique ”*context analysis*” is used to estimate a person's height using computer vision by analyzing the surrounding environment or objects in the image. This technique is based on the idea that the size and position of objects or structures in the environment can provide information about the height of the person relative to those objects. To use this technique, the algorithm must first identify and locate the objects or structures in the environment that can be used as reference points. This can be done using techniques such as object detection or image segmentation. Once these reference points are identified, the algorithm can analyze their size and position relative to the person to estimate the person's height. For example, if the algorithm detects a door frame in the image, it can use the known height of the door frame to estimate the height of the person relative to the door frame. Similarly, if the algorithm detects a tree or a lamp post in the image, it can use the height of these objects to estimate the height of the person relative to these objects. It is important to note that this technique can be affected by the distance of the person from the reference points, the angle of the camera, and the accuracy of the object detection or segmentation. Therefore, it is not always accurate. However, when combined with other techniques, such as pattern analysis or perspective analysis, it can improve the overall accuracy of the height estimation [19], [20]. Below the equation used to calculate height using this technique is shown.

$$height = \frac{size\_ref\_object \cdot distance\_person\_to\_object}{size\_person\_in\_image} \tag{5}$$

Another proposed technique to estimate the height of a person is *”perspective analysis”*, this technique is based on the analysis of perspective distortion in the image. The main idea behind this technique is that a person's height appears smaller in the image as the person moves further away from the camera or as the camera moves further away from the person. By analyzing the degree of perspective distortion, the algorithm can estimate the distance of the person from the camera, which can be used to estimate the person's height. To use this technique, the algorithm must first calibrate the camera to determine its intrinsic and extrinsic parameters. This can be done using a calibration pattern, such as a checkerboard, and techniques such as camera calibration or Zhang's method. Once the camera is calibrated, the algorithm can use intrinsic and extrinsic parameters to correct the perspective distortion in the image. The perspective analysis technique can also be combined with other techniques, such as object detection or segmentation, to improve the accuracy of height estimation. For example, if the algorithm detects the person's feet and head in the image, it can estimate the distance between them, which can be used to estimate the person's height. By combining the distance estimation from perspective analysis with the size and position of the reference objects from object detection or segmentation, the algorithm can improve the overall accuracy of the height estimation [21]. The algorithm is used to estimate the height of a person based on their distance from the camera and the known height of a reference object. Given the known height of the reference object $height\_ref$ in meters, the distance of the person from the camera d in meters, and the vertical distance between the top of the person's head and the top of the reference object in pixels $height\_px$, the height of the person $height\_m$ in meters can be estimated using the equation shown below. Note that this equation assumes that the person is standing upright and that their feet are at the same level as the reference object.

$$height\_m = \frac{height\_ref \cdot d}{height\_px} \tag{6}$$

The last technique reviewed is *”depth analysis”*, this technique consists of estimating the depth or distance of objects in a scene from a camera or sensor. This is typically done using techniques such as stereo vision, structured light, or time-of-flight sensing. For this work, a review of stereo vision involves using two or more cameras to capture different views of a scene, which can be used to triangulate the 3D position of objects based on the differences in the images captured by each camera. The distance between the cameras (known as the baseline) and their internal parameters (known as the camera matrices) are used to calculate the depth of each pixel in the scene [22]. The depth of a point in a scene can be calculated using the equation shown below.

$$Z = \frac{f \cdot B}{xL - xR} \tag{7}$$

where Z is the depth of the point, f is the focal length of the camera, B is the distance between the left and right cameras (known as the baseline), xL is the x-coordinate of the point in the left image, and xR is the x-coordinate of the point in the right image.

It is important to note that this technique can be affected by the posture of the person, the angle of the camera, and the accuracy of the parts detection, so it is not always accurate.

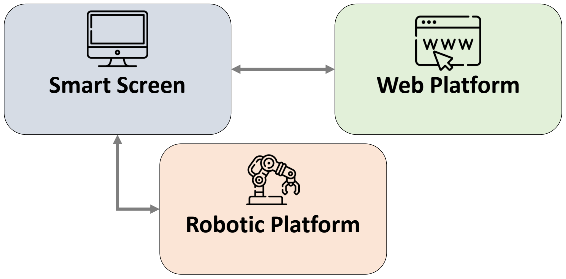

Fig. 1. General outline of the proposed architecture for the smart receptionist project, the Smart Screen module represents the work proposed in this paper.

However, when combined with other techniques, such as context analysis or perspective analysis, it can improve the overall accuracy of the height estimation. It is also worth noting that this technique can be used in combination with other computer vision techniques, such as object detection, image segmentation, and feature extraction, to improve the accuracy of height estimation.

III. Proposed Approach

The proposed approach consists of five stages, initially, the images are acquired from different cameras, and the person in the given image is detected, then some features are extracted to use them in the following steps, in the next stage the distance between the camera and the person is estimated (eg, depth) and finally the height of the person is calculated. Since the first stages are based on learning approaches, the success of the whole pipeline would depend on the quality of these results. That is why different techniques have been evaluated to establish which is the best option. Before presenting the proposed approach the dataset generated in the current work is detailed.

A. Image Acquisition

This section details the process carried out for the acquisition of the data using for its evaluation the different techniques applied in the pipeline of this work. Images, where humans were present, have been acquired in a controlled environment using a multi-touch smart screen, two visible spectrum cameras (e.g., webcams), with a resolution of 640×480 pixels, together with one LED lamp of 18 watts each, placed on top. The camera has been placed orthogonal to the plane containing the person and the color space used is RGB. Additionally, the smart screen features an Intel Core i7-1225U and NVIDIA GeForce MX550 GPU.

On the other hand, the present work is part of an architecture for a smart receptionist framework. Figure 1 shows the general scheme of the entire project, and the module called ”Smart Screen” represents the work proposed in this document. The other modules as well as the integration of each one are part of the future work.

B. Human Detection

After the images have been acquired, the next step is to detect humans present in the images. For this task, two deep learning networks are used for evaluation, such as YOLO v7 and MediaPipe. Pretrained weights from the COCO dataset [23] were used for each network. In this stage, the person under study present in the scene is first identified, then the bounding box is detected and later features of the person's face are extracted to carry out subsequent calculations. Both Yolo v7 and MediaPipe networks allow extractring facial features with high precision.

C. Feature Extraction

After detecting the person in the scene, the next step is feature extraction. In the present work, the most important features are extracted from the person in the study. The Table I shows the features extracted and used for the depth estimation stage, while the Table II shows the features extracted and used for the height estimation stage.

TABLE I
Feature Extraction for Depth Estimation

Neural Network	**Key Points**	**Extracted Key Points**	
	Model Output	***Face***	***Feet***
YOLO v7 Pose	17	5	2
MediaPipe Face Detector	6	6	-
MediaPipe Pose	32	11	2
MediaPipe Face Mesh	468	468	-

From the inference output of each neural network, all available key points for the face were selected, while 2 key points of the feet are also selected if these are available in the output of the neural network. A subset of specific key points is selected for the height estimation.

TABLE II
Feature Extraction for Height Estimation

Neural Network	**Key Point Index**			
	Eyes	***Mouth***	***Nose***	***Feet***
YOLO v7 Pose	14, 15	-	0	10, 13
MediaPipe Face Detector	0, 1	3	-	-
MediaPipe Pose	2, 5	9, 10	-	29, 30
MediaPipe Face Mesh	33, 263	61, 291	-	-

D. Depth Estimation

All the key points extracted from the face in the previous step are then used to determine the distance between the person and the camera. For this task two depth estimation techniques, MiDaS and classic Stereo Vision are evaluated.

The first technique MiDaS is a deep learning-based approach that outputs the relative depth from the camera to the objects in the scene taking as input a single RGB image, to convert from relative to absolute depth a linear transformation must be found, this achieves positioning several objects in the

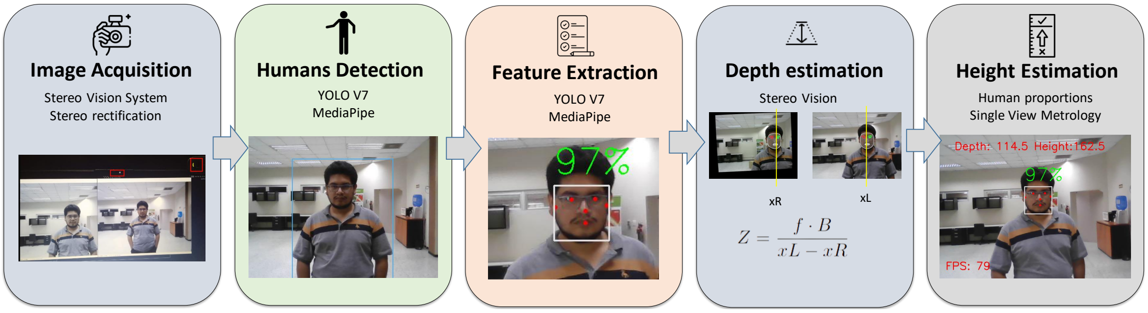

Fig. 2. Overall pipeline proposed for human height estimation.

scene at know depths and finding the linear transformation that best maps the relative depth to the absolute depth [24].

The second technique evaluated is depth estimation using classic Stereo Vision, the first step to using this technique is image rectification, the rectification process aligns the z and y axis of the images, so the disparity can be found only in the x axis. A chessboard calibration pattern and OpenCV library [25] are used in this task. Finally, the depth can be calculated using the Equation 7; the distance between the two cameras was set to $B = 15cm$, the focal length f measured in pixels was calculated using OpenCV [25], xR and xL corresponds to a set of keypoints extracted from images in the of the left and right cameras respectively.

E. Height Estimation

After finding the distance from the person to the cameras, the last step in the pipeline is to find the person's height, for this task, two techniques are used.

The first technique *"single view metrology"* does not use the depth found in the previous section, instead it places an object of known geometry in the scene and allows to measure the height of any other object or person in the scene relative to the height of the first object. However, one of the limitations of this technique is that it requires finding the position where the person's feet touch the ground to accurately predict the height, as explained in the work presented by [20]. This limitation also implies that only YOLO v7 Pose and MediaPipe Pose can be used in combination with this technique.

The second technique *"human proportions"* uses the standard proportions of the human body and the human face to predict the height of the person. In the case of YOLO v7 the eyes and the nose key points are used while MediaPipe uses the eyes and the mouth key points, these points are used to find the head size in pixels.

For MediaPipe (Pose, Face detector, Face mesh) the size of the head in pixels is found using Equation 8 and for YOLO v7 using Equation 9.

$$head\_size\_cm = distancePx(eyes, mouth) * 3 \tag{8}$$

$$head\_size\_cm = distancePx(eyes, nose) * 6 \tag{9}$$

where $eyes$ is the average of the eyes coordinates, $mouth$ is the coordinates of the mouth, $nose$ is the coordinates of the nose, and $distancePx$ is the Euclidean distance between 2 points in the pixel space of the image.

Then the size of 1 pixel in centimeters must be found. This is a linear function that uses the distance from the camera to the person (e.g., $depth$) as input. It also uses a positive conversion constant k and a correction factor $offset$ which should be zero, the value of these constants depends on the stereo cameras configuration being used, and the values of k and $offset$ can be calculated by placing several objects of known size in the scene at different heights and finding the linear regression that best fits the data. Equation 10 shows the function using these values.

$$f(depth) = k * depth + offset \tag{10}$$

After finding the function that gets the size of 1 pixel in centimeters and the head size in pixels a multiplication of these values gets the head size in centimeters. Equation 11 shows the calculation of this value.

$$head\_size\_cm = f(depth) \cdot head\_size\_pixel \tag{11}$$

Finally, the human height can be found using the standard body proportion that determines that the height of a human is roughly 8 times the size of their head [26]. Equation 12 shows the calculation of this value.

$$human\_height = 8 \cdot head\_size\_cm \tag{12}$$

IV. Experimental Results

This section presents the experimental results obtained with the proposed pipeline. Evaluations on the two stages: human detection and height estimation are provided showing the performance of each one of them in comparison with the state-of-the-art approaches.

A. Human Detection

The first task evaluated is the detection of humans in the scene. To carry out this task, two architectures based on deep learning are evaluated, such as YOLO v7 and MediaPipe. Due to the context and restrictions of the general framework of which this work is a part (see Fig. 1), the execution and obtaining of results is in real-time, so one of the metrics to be evaluated is the frames per second (FPS) that each technique can obtain. YOLO v7 allows the detection of multiple people in the scene in addition to the use of GPU, while MediaPipe is optimized to run on CPU and only allows the detection of 1 person in the scene. Because YOLO v7 detects several people at the same time the number of FPS is lower than using MediaPipe.

The results in Table III show that both MediaPipe and YOLO v7 can be used in real-time applications, such as the smart receptionist framework described in this work. In general, MediaPipe shows better performance and fewer hardware requirements. Although its main disadvantage is that it does not allow multiple detections of people in the same scene.

TABLE III
HUMAN DETECTION TECHNIQUES EVALUATED WITH DIFFERENT FPS

Evaluated Architectures	**FPS for Human Detection**	
	CPU	***GPU***
YOLO v7 Pose	1.27	27.8
MediaPipe Face Detector	68.7	-
MediaPipe Face Mesh	57.9	-
MediaPipe Pose	42.2	-

B. Depth Estimation

The next step is one of the most important and consists of estimating the distance between the camera and the face of the person under study. To carry out this task, two techniques MiDaS and classic Stereo Vision depth estimation were used. The mean absolute error was determined from a set of distances ranging from 30 to 150 cm.

TABLE IV
DEPTH ESTIMATION TECHNIQUES EVALUATED WITH DIFFERENT METRICS

Evaluated Techniques	**Depth Estimation**		
	MAE	σ	***Var***
MiDaS	15.27	20.82	433.47
Stereo Vision	1.34	4.95	24.59

C. Height Estimation

The last step within the pipeline is the estimation of the person's height, for which a study was carried out with 10 people of different heights, with which the results of the 10 techniques evaluated are shown in Table V. The Table V shown in the last row that the technique used by **MediaPipe + Face Mesh + Tracking + Stereo Vision (SV)** obtains an MAE value of less than 1.0 cm, in addition to a standard deviation σ = 4.89 and a variance (Var) of 23.91, which shows the low dispersion of the data as well as a low mean absolute error value. In general, it is shown that the use of MediaPipe presents better results in all calculated metrics (e.g., MAE, σ and Var) in conjunction with tracking, face mesh, and stereo vision techniques.

Table VI shows a comparison of different approaches based on the MAE metric. It can be observed that the proposal presented in this work obtains 0.89 cm improving values obtained with other proposals of the state-of-the-art.

TABLE V
HUMAN HEIGHT ESTIMATION TECHNIQUES EVALUATED WITH DIFFERENT METRICS

†SV (Stereo Vision). †SVM (Single View Metrology).

Techniques	**Human Height**		
	MAE	σ	***Var***
YOLO v7+Pose+MiDaS	5.26	15.24	232.25
YOLO v7+Pose+SVM	2.80	7.45	55.50
YOLO v7+Pose+SV	1.11	6.24	38.93
MediaPipe+Pose+Tracking+MiDaS	6.93	7.58	57.45
MediaPipe+Pose+Tracking+SVM	4.82	3.92	15.36
MediaPipe+Pose+Tracking+SV	0.95	5.98	35.76
MediaPipe+Face Detector+MiDaS	7.58	10.23	104.65
MediaPipe+Face Detector+SV	9.78	15.89	252.49
MediaPipe+Face Mesh+Tracking+MiDaS	7.32	9.38	87.98
MediaPipe+Face Mesh+Tracking+SV	**0.89**	**4.89**	**23.91**

TABLE VI
COMPARISON OF DIFFERENT APPROACHES ESTIMATING HUMAN HEIGHT BASED ON THE MAE METRIC

Techniques	**Human Height**
	MAE
Bieler et. al [19]	3.90
Jeon et. al [27]	1.43
Kim et. al [28]	4.28
Lee et. al [18]	1.37
Momeni-K et. al [20]	1.86
MediaPipe+Face Mesh+Tracking+SV (our)	**0.89**

V. CONCLUSIONS

This paper proposes a deep learning-based human height estimation approach using a stereo vision system to be used in a smart receptionist framework. Ten different combinations of techniques have been tried to find the height of a person, as shown in Table V the most accurate technique is **MediaPipe + Face Mesh + Tracking + Stereo Vision (SV)**. Also in Tables III and IV-A shown that using a MediaPipe with only CPU can run on real-time which makes it suitable for height estimation in the context of a smart receptionist framework. Finally, Table VI shows that the proposal presented in this work with a value of $MAE = 0.89cm$ improves the values of techniques presented in the state-of-the-art.

In order to improve the accuracy of the person's height, improvements can be studied. The environmental lighting conditions would affect the results, the use of a different color space than RGB (such as HSV) could be a solution. Finally, the proposed approach relies on the vertical pixel distance between the eyes, mouth, and nose but the influence of the face pitch angle and its correction must be studied.

ACKNOWLEDGMENT

This work has been partially supported by the ESPOL EPASI project (CIDIS-01-2018).

REFERENCES

[1] J. L. Charco, A. D. Sappa, B. X. Vintimilla, and H. O. Velesaca, "Transfer learning from synthetic data in the camera pose estimation problem." in *VISIGRAPP*, 2020, pp. 498–505.

[2] J. L. Charco, A. D. Sappa, B. Vintimilla, and H. O. Velesaca, "Camera pose estimation in multi-view environments: From virtual scenarios to the real world," *Image and Vision Computing*, vol. 110, p. 104182, 2021.

[3] H. O. Velesaca, S. Araujo, P. L. Suárez, A. Sánchez, and A. D. Sappa, "Off-the-shelf based system for urban environment video analytics," in *Int. Conf. on Systems, Signals and Image Processing*. IEEE, 2020, pp. 459–464.

[4] J. L. Charco, A. D. Sappa, and B. X. Vintimilla, "Human pose estimation through a novel multi-view scheme." in *VISIGRAPP*, 2022, pp. 855–862.

[5] S. Yoon, T. Choi, and S. Sull, "Depth estimation from stereo cameras through a curved transparent medium," *Pattern Recognition Letters*, vol. 129, pp. 101–107, 2020.

[6] M. Caon, "Voxel-based computational models of real human anatomy: a review," *Radiation and environmental biophysics*, vol. 42, pp. 229–235, 2004.

[7] A. Jalal, A. Nadeem, and S. Bobasu, "Human body parts estimation and detection for physical sports movements," in *Int. Conf. on Communication, Computing and Digital Systems*. IEEE, 2019, pp. 104–109.

[8] S. Li, V. H. Nguyen, M. Ma, C.-B. Jin, T. D. Do, and H. Kim, "A simplified nonlinear regression method for human height estimation in video surveillance," *Journal on Image and Video Processing*, vol. 2015, no. 1, pp. 1–9, 2015.

[9] A. Trivedi, M. Jain, N. K. Gupta, M. Hinsche, P. Singh, M. Matiaschek, T. Behrens, M. Militeri, C. Birge, S. Kaushik *et al.*, "Height estimation of children under five years using depth images," in *Int. Conf. Engineering in Medicine & Biology Society*. IEEE, 2021, pp. 3886–3889.

[10] C.-M. Wang and W.-Y. Chen, "The human-height measurement scheme by using image processing techniques," in *2012 International Conference on Information Security and Intelligent Control*. IEEE, 2012, pp. 186–189.

[11] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, 2022.

[12] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee *et al.*, "Mediapipe: A framework for building perception pipelines," *arXiv preprint arXiv:1906.08172*, 2019.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European Conference on Computer Vision*. Springer, 2016, pp. 21–37.

[14] J. P. Wilson, A. M. Kanaya, B. Fan, and J. A. Shepherd, "Ratio of trunk to leg volume as a new body shape metric for diabetes and mortality," *PLoS One*, vol. 8, no. 7, p. e68716, 2013.

[15] M. Vukotic, "Body height and its estimation utilizing foot length measurements in montenegrin adolescents: a national survey," *Nutrición hospitalaria: Organo oficial de la Sociedad española de nutrición parenteral y enteral*, vol. 37, no. 4, pp. 794–798, 2020.

[16] G. Yang, D. Li, G. Ru, J. Cao, and W. Jin, "Body height estimation system based on binocular vision." *International Journal of Online Engineering*, vol. 14, no. 4, 2018.

[17] N. Sarafianos, C. Nikou, and I. A. Kakadiaris, "Predicting privileged information for height estimation," in *International Conference on Pattern Recognition*. IEEE, 2016, pp. 3115–3120.

[18] D.-s. Lee, J.-s. Kim, S. C. Jeong, and S.-k. Kwon, "Human height estimation by color deep learning and depth 3d conversion," *Applied Sciences*, vol. 10, no. 16, p. 5531, 2020.

[19] D. Bieler, S. Gunel, P. Fua, and H. Rhodin, "Gravity as a reference for estimating a person's height from video," in *Proceedings of Int. Conf. on Computer Vision*, 2019, pp. 8569–8577.

[20] M. Momeni-K., S. C. Diamantas, F. Ruggiero, and B. Siciliano, "Height estimation from a single camera view," in *Int. Conf. on Computer Vision Theory and Applications*, 2012.

[21] S.-W. Park, T.-E. Kim, and J.-S. Choi, "Robust estimation of heights of moving people using a single camera," in *Proceedings of the Int. Conf. on IT Convergence and Security*. Springer, 2012, pp. 389–405.

[22] Y. M. Mustafah, R. Noor, H. Hasbi, and A. W. Azma, "Stereo vision images processing for real-time object distance and size measurements," in *Int. Conf. on Computer and Communication Engineering*. IEEE, 2012, pp. 659–663.

[23] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*. Springer, 2014, pp. 740–755.

[24] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," 2019. [Online]. Available: https://arxiv.org/abs/1907.01341

[25] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[26] Y.-P. Guan, "Unsupervised human height estimation from a single image," *Journal of Biomedical Science and Engineering*, vol. 2, pp. 425–430, 01 2009.

[27] S. H. Jeon, J. K. Song, J. S. Park, and B. W. Yoon, "Video based pedestrian height estimation using winer optimization," *Journal of Korea Multimedia Society*, vol. 19, no. 2, pp. 264–270, 2016.

[28] S.-M. Kim, J.-K. Song, B.-W. Yoon, and J.-S. Park, "Height estimation using kinect in the indoor," *The Journal of the Korea institute of electronic communication sciences*, vol. 9, no. 3, pp. 343–350, 2014.