Human Detection and Gesture Recognition for the Navigation of Unmanned Aircraft

Markus Lieret[a], Maximilian Hübner[b], Christian Hofmann[c] and Jörg Franke[d]
Institute for Factory Automation and Production Systems, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Egerlandstr. 7-9, 91058 Erlangen, Germany

Keywords: Machine Learning, Gesture Recognition, Computer Vision, Unmanned Aircraft, Indoor Navigation.

Abstract: Unmanned aircraft (UA) have become increasingly popular for different industrial indoor applications in recent years. Typical applications include the automated stocktaking in high bay warehouses, the automated transport of materials or inspection tasks. Due to limited space in indoor environments and the ongoing production, the UA oftentimes need to operate in less distance to humans compared to outdoor applications. To reduce the risk of danger to persons present in the working area of the UA, it is necessary to enable the UA to perceive and locate persons and to react appropriately to their behaviour.
Within this paper, we present an approach to influence the flight mission of autonomous UA using different gestures. Thereby, the UA detects persons within its flight path using an on-board camera and pauses its current flight mission. Subsequently, the body posture of the detected persons is determined so that the persons can provide further flight instructions to the UA via defined gestures. The proposed approach is evaluated by means of simulation and real world flight tests and shows an accuracy of the gesture recognition between 82 and 100 percent, depending on the distance between the persons and the UA.

1 INTRODUCTION

In recent years, unmanned aircraft (UA) have become increasingly popular in numerous areas of application. Typical applications include filming and photography, surveying and inspection and the transport of medical goods. All listed application benefit in particular from the flexibility and three-dimensional workspace of the UA. In addition to those examples, intensive research is also being conducted on the use of autonomous UA in industrial contexts. Thereby, possible fields of application include the automation of stocktaking processes or inspections and the transport of urgently needed components within a factory site and between different locations.

However, the advantages of UA are countered by numerous concerns from the population. Studies have shown that 40 % of the population in Germany, for example, still have a rather negative attitude toward drones. In addition to the possible infringement on privacy, the reasons cited include the risk of crashes and associated injuries. Further, the rotating propellers of the commonly used multirotor systems also pose a high risk of injury. (Eißfeldt et al., 2020) (Lidynia C., 2017)

Particularly in industrial applications where UA can operate in the direct vicinity of persons (e.g. during material delivery or stocktaking in high bay warehouses), suitable measures must be taken to exclude hazards to persons and to ensure the acceptance of the UA. One of the most commonly used measure to reduce the risk of injury is the complete enclosure of rotors or the entire UA. However, since this reduces the usable payload of the UA, systems have also been developed that abruptly stop a rotor before a possible collision (Pounds and Deer, 2018). As this system can lead to an unintended crash of the UA that endangers people, practical safety measures for UA currently still rely on spatial or structural separation of the UA and human workers. However, this reduces the UA's application possibilities and flexibility, which is why these measures are not expedient in the medium term.

The research also focuses on the detection and localization of persons in the working area of the UA in order to be able to derive appropriate emergency measures before a hazardous situation occurs. For

[a] https://orcid.org/0000-0001-9585-0128
[b] https://orcid.org/0000-0003-2761-7046
[c] https://orcid.org/0000-0003-1720-6948
[d] https://orcid.org/0000-0003-0700-2028

Lieret, M., Hübner, M., Hofmann, C. and Franke, J.
Human Detection and Gesture Recognition for the Navigation of Unmanned Aircraft.
DOI: 10.5220/0010832500003124
In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022) - Volume 4: VISAPP*, pages 831-838
ISBN: 978-989-758-555-5; ISSN: 2184-4321

example, radio frequency identification (RFID) systems (Koch et al., 2007) and high-visibility vests detected by a colour camera (Mosberger and Andreasson, 2013), have been identified as fundamentally suitable systems. However, since these approaches require adjustments to the existing infrastructure and presuppose that the persons to be detected carry an RFID reader with them or reliably wear their personal protective equipment, these solutions are also only suitable for practical use to a limited extent.

In order to be able to use autonomous UA purposefully in areas where people are present and to minimize the risk to these people, it is necessary that people can be reliably detected with sensors that are located exclusively on the UA. Additionally, it must be possible for the UA to react to instructions from these persons, for example to initiate an emergency landing or continue its current mission after explicit clearance.

Therefore, in the following a methodology is presented in which the UA uses an on-board RGB-D camera to capture the area in the direction of flight and recognizes persons present based on the colour image. Subsequently, the colour and depth data is used to determine the current pose and posture of the detected persons so that the persons can give instructions to the UA using simple gestures. The main contributions of this paper are the architecture of the overall system used to locate persons, react to their gestures and thus to enable a safe interaction between the UA and persons. Beyond, a novel approach to recognize and distinguish different gestures is presented.

2 RELATED WORK

Crucial for reliable gesture-based control of robots is the methodology used for gesture recognition. Due to the rapid development of machine learning methods in recent years, these have become an established tool for various available variants of gesture recognition. Nowadays, a large number of models exist that determine the human pose in two- or three-dimensional space based on colour and depth images. (Chen et al., 2020) Furthermore, various commercially available cameras and sensor systems such as Microsoft's Kinect series already support direct computation of human body posture when using the provided software development kits. (Le et al., 2013)

Available approaches for gesture-based control of UA can be fundamentally divided into two categories. There are approaches in which neural networks are used to classify discrete body postures or recognize body parts. Afterwards a flight command is executed based on the detected body posture or the relation of the recognized body parts to one another. This contrasts with methods where flight commands are executed based on gestures, which are derived from a skeleton model, determined using machine learning methods.

Maher et al. (Maher et al., 2017) use the YOLOv2 object detector to detect and locate the head and both hands in colour images. Individual gestures are then defined using the relative position between the right and left hand and the head, and linked to defined flight actions. The functionality of the resulting gesture control is eventually verified in an experiment. A similar approach is presented by Zhang et al. (Zhang et al., 2019a). They use MobileNet-SSD as detection network and also detect and locate the head and both hands to derive gestures to control a mobile robot. With their approach they are able to identify around 87 percent of the defined gestures correctly.

Instead of deriving the gestures from the position of different body parts, Kassab et al (Kassab et al., 2020) train different deep classification frameworks to identify the defined gestures in an image.

Sanna et al. (Sanna et al., 2012) present a system using a stationary Kinect camera in conjunction with the NITE skeleton tracker to relay motion commands to the UA using various gestures. The NITE skeleton tracker is also used by Yu et al. (Yu et al., 2017) in conjunction with the Asus Xtion Pro Live. The authors show that the average gesture recognition rates are greater than 90 % in this case. A similar setup is used by Tellaeche et al. (Tellaeche et al., 2018) to control a drone. Instead of geometric relationships between the joint points, an adaptive naive bayes classifier is used to determine the gestures. Again, the authors are able to achieve gesture recognition rates of greater than 90 % from different distances.

Extending the solely gesture based solutions, Zhang et al. present an approach that optionally allows control of a drone using a stationary Kinect camera eye tracking and voice commands. (Zhang et al., 2019b).

Asides, the OpenPose framework presented by Cao et al. (Cao et al., 2021) is oftentimes used to perform single- or multi-person 2D pose estimation and derive actions for mobile robots or drones from the provided skeleton model. Using this framework and the YOLO object detection system, Medeiros et al. (Medeiros et al., 2020) demonstrate, that an UA can be sent to different target objects by pointing gestures. Cai et al. (Cai et al., 2019) combine OpenPose with a Support Vector Machine, to perform robust gesture estimation and drone control based on the distance between the identified joints of the skeleton

model. Instead of a SVM Liu and Szirányi (Liu and Szirányi, 2021) use a deep neuronal network to identify gestures. However, they do not use the identified gestures to control the UA.

Besides the listed approaches, that are able to distinguish between several individual gestures, Monajjemi et al. (Monajjemi et al., 2016), present a recognition of persons based on arm waving gestures. They propose a periodic waving gesture detection algorithm in (Monajjemi et al., 2015) which is then used to attract the UA's attention and to communicate with the UA using simple waving gestures.

According to the previous literature review, the existing approaches focus primarily on the exclusive manual control of robots. Additionally, the sensor system is oftentimes not attached to the UA. Thus, motion blur does not affect the image quality and subsequently the gesture recognition. Further, many of those approaches are not suited for UA applications in large areas since in such scenarios UA and persons can meet on the fly, consequently requiring the need of instant human-UA interaction.

We present an approach in which the UA continuously captures its environment during an autonomous mission, detects persons and adapts its flight behaviour according to the gestures of the detected persons. Thus, extending to the state of the art, persons in the environment of the UA are given the possibility to influence the flight behaviour according to their subjective perception of danger. For this purpose, a human-UA interaction strategy is proposed and a novel approach for gesture recognition is developed, whose performance is on the level of the algorithms presented in the related work. The gestures are selected in such a way that they are on the one hand easy to understand and perform for persons and on the other hand robustly and with high accuracy recognizable to our developed algorithm.

3 METHODOLOGY

As described in the introduction, the goal of our research is to enable an UA to detect and localize persons in its working environment and subsequently adapt its flight behaviour based on recognized gestures. This allows the UA to stop the current flight mission as soon as a person has been detected and to continue only after explicit clearance. Furthermore, the UA can be requested to land or perform a suitable evasive manoeuvre by means of additional gestures. In the following, the general system architecture and the gesture recognition approach are presented in detail. Afterwards, additional information on the implementation and the used software and hardware components are provided.

3.1 System Architecture

To achieve the objectives described above, we propose a methodology as presented in Figure 1 (left). Thereby, the autonomous UA is equipped with an RGB-D camera, which captures the spatial area in the direction of flight. Based on the individual colour images, persons are recognized and the associated skeleton model is computed. Using the skeleton model, the gestures performed by the recognized persons are calculated and the associated flight instructions are transmitted to the autopilot.

As shown in Figure 1 (right), the overall system architecture is divided into two main components. First, the colour images provided by the camera are processed directly on the on-board computer of the UA to detect persons ahead of the UA. The processing is done on-board, to ensure a detection of persons even when the communication with the ground control station (GCS) is interrupted. When a person is detected, an appropriate stop signal is sent to the autopilot, which is responsible for the automated execution of automated flight missions. The autopilot then pauses the current mission and prompts the UA to hover in place at the current position and wait for further instructions. To be able to detect persons within an appropriate amount of time even with limited on-board calculation power, the YOLOv4 resp. Tiny-Yolov4 (Bochkovskiy et al., 2020) real-time object detection system is used for this task.

Additionally, the colour and depth data are compressed and sent wirelessly to the GCS for further processing. In the first step, the OpenPose framework (Cao et al., 2021) is used to determine the skeleton model of the detected persons. OpenPose is a convolutional neural network (CNN) based framework that requires a colour image as input and can determine a skeleton model with up to 25 body points of all detected persons in the given image. Using the procedure described in the following section 3.2, the resulting skeleton points are used to check whether one or more persons gesture in a predefined way.

If more than one person is detected in the image, the gestures are distance-filtered using the provided depth information. This ensures that only the command associated with the gesture of the person closest to the UA is forwarded to the auto pilot. The exception is the request for a landing, which is forwarded regardless of the distance. A detailed list of the defined gestures and associated commands is given in the following section.

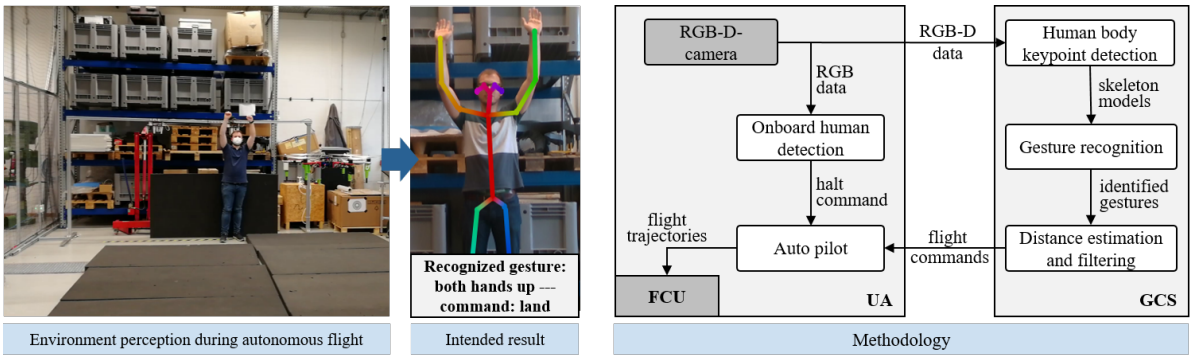


Figure 1: Left: Visualization of an autonomous UA, equipped with the proposed system. The UA recognizes the person and the performed gestures and conducts the associated command. Right: Architecture of the proposed framework.

3.2 Gesture Recognition and Filtering

The recognition of the individual gestures is based on the geometric relationships between the provided skeleton points. To distinguish different gestures, two angles α and β are introduced. Thereby α represents the posture of the shoulder joint and β the posture of the elbow joint. Figure 2 shows the relationship of the angles and limbs used for the recognition of a gesture performed with the right arm. Gestures performed with the left arm are defined accordingly.

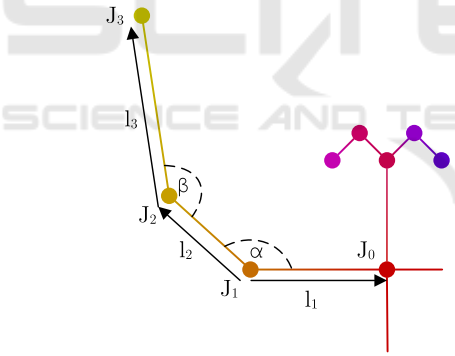


Figure 2: Schematic representation of the calculation of the parameters with which a gesture is detected.

Starting from the recognized joint points J_n, the associated position vectors are calculated using the coordinates of the individual joints provided by OpenPose. These position vectors are then used to compute the direction vectors $\vec{l}_n$, each representing a limb. The subsequent calculation of the angles is exemplified in the following equations for the angle α, which is calculated by

$$\alpha = \arccos \frac{\vec{l}_1 \circ \vec{l}_2}{|\vec{l}_1| \cdot |\vec{l}_2|} \tag{1}$$

with

$$\vec{l}_1 = \vec{j}_0 - \vec{j}_1 = \overrightarrow{OJ_0} - \overrightarrow{OJ_1} \qquad J_n \in \mathbb{R}^3 \tag{2}$$

$$\vec{l}_2 = \vec{j}_2 - \vec{j}_1 = \overrightarrow{OJ_2} - \overrightarrow{OJ_1} \qquad J_n \in \mathbb{R}^3 \tag{3}$$

where $\overrightarrow{OJ_n}$ is the position vector of the respective joint. The angle β is calculated accordingly. Based on the two angles, the gestures given in Table 1 can be distinguished. An unique identifier that will also be used within the evaluation in Section 4 names every gesture.

The identifier *lar (left arm raised)* indicates a raise of the left arm, *las (left arm sideways)* a laterally stretched left arm. The same gestures can be performed also with the right arm (*rar, ras*) or both arms (*bar, bas*). If no person is present or no gesture performed, the image is identified as *ng* (no gesture). Table 1 provides the angle ranges used to define the distinct gestures and the flight command associated with each gesture. Thereby, angles that apply to positions of the right arm are denoted with index r, positions of the left arm with the index l. A robust recognition of the individual gestures is ensured by choosing sufficiently large angle ranges and clear distance between the angle ranges of two different gestures. Multiple evaluations, both within the simulation and the real-world environment, were performed to optimize the angle ranges and obtain values that allow a reliable detection of the gestures independently of the body height of the performing person.

As mentioned, if more than one person performing a valid gesture is recognized, only the command indicated by the gesture of the person closest to the UA is performed. Therefore, the three-dimensional position of each skeleton point is calculated using the provided image coordinates, the depth image and the intrinsic camera parameters. The distance between the person and the UA is then calculated as the mean value of the distance of the individual skeleton points and used to filter the gestures.

Table 1: Unique identifier, ranges of the corresponding joint angles and associated flight command for the defined gestures.

identifier	angles	command
lar	$80° < \alpha_l < 160°$, $125° < \beta_l < 190°$	Descend 0.5 m
las	$155° < \alpha_l < 190°$, $155° < \beta_l < 190°$	Fly 1 m left
rar	$80° < \alpha_r < 160°$, $125° < \beta_r < 190°$	Ascend 0.5 m
ras	$155° < \alpha_r < 190°$, $155° < \beta_r < 190°$	Fly 1 m right
bar	$80° < \alpha_{l/r} < 155°$, $115° < \beta_{l/r} < 190°$,	Land
bas	$155° < \alpha_{l/r} < 190°$, $155° < \beta_{l/r} < 190°$	Continue

4 EVALUATION

The following evaluation of the presented methodology is based on a simulation as well as on real world flight tests.

Within the simulation, a virtual person is placed in a suitable environment and the individual gestures are generated by corresponding animation of the virtual character. A virtual RGB-D camera with a resolution of 640x480 captures the environment as well as the movements and gestures of the person.

For the real world flight tests a custom-built hexarotor system is used, which contains a Pixracer flight control unit running the PX firmware v1.12.2. An Intel RealSense D435 camera is used to capture RGB-D images of the environment with a resolution of 640x480 for both colour and depth images.

For example, either a LattePanda Alpha 864s or an NVIDIA Jetson Xavier NX can be used as the on-board computer on the UA. The on-board person detection using the Jetson in conjunction with the YOLOv4 or TinyYolo CNN will not be evaluated further, as numerous analyses on the achievable accuracy and performance of those CNNs on mobile hardware already exist. The following results were obtained using the LattePanda Alpha 864s.

The overall software architecture is implemented using the Robot Operating System (ROS) and a custom ROS wrapper for the OpenPose framework. For evaluation, the camera data is transmitted from the UA to a GCS via WLAN. After the skeleton model is determined using OpenPose, the joint points are transmitted to an additional ROS node, where the determination of the gestures and the filtering is performed. Finally, the flight commands associated with the gestures are transmitted back to the navigation and control framework running on the UA. A desktop PC with an Intel Xeon W-1390P and a NVIDIA RTX Quadro 6000 GPU is used as GCS for recognizing the persons and gestures.

For the simulation-based evaluation the data processing procedure is analogous. The simulation of the image data required for gesture recognition is implemented using the Unity game engine, the simulation of the UA and the flight movements is running simultaneously in Gazebo. However, data transmission via WLAN is omitted in the simulation. Thus, the simulation and calculation are carried out exclusively on the ground station.

To benchmark the performance of the proposed gesture recognition approach, we evaluate the classifier performance using a confusion matrix for multiclass classification and determine the accuracy A and the macro average of the sensitivity S_{macro}, the precision P_{macro} and the F1-score $F1_{\text{macro}}$.

Therefore, for each class c, which represents a distinct gesture, the true positives (TP), the false positives (FP), the true negatives (TN) and the false negatives (FN) are calculated. Based on those values, the evaluation metrices A, S_{macro}, P_{macro} and $F1_{\text{macro}}$ are calculated as follows, whereby N indicates the total number of classes c.

$$A = \frac{\sum_{i=0}^{N} TP_i + TN_i}{\sum_{i=0}^{N} TP_i + TN_i + FP_i + FN_i} \tag{4}$$

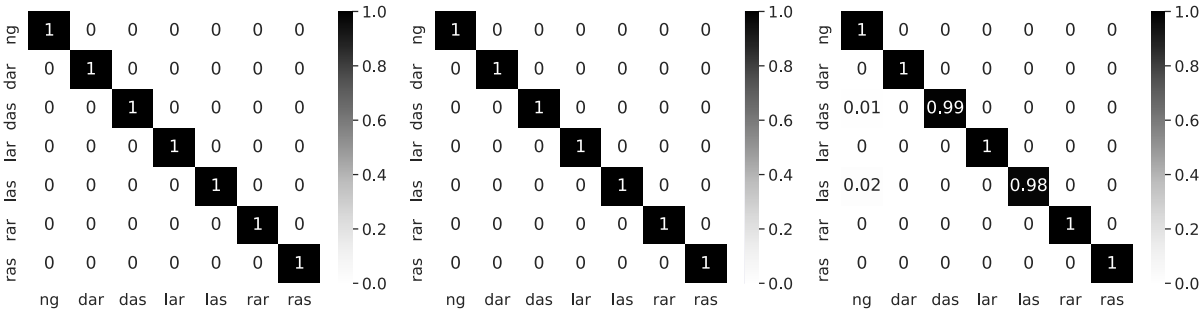
$$P_{\text{macro}} = \frac{1}{N} \sum_{i=0}^{N} \frac{TP_i}{TP_i + FP_i} \tag{5}$$

$$S_{\text{macro}} = \frac{1}{N} \sum_{i=0}^{N} \frac{TP_i}{TP_i + FN_i} \tag{6}$$

$$F1_{\text{macro}} = \frac{1}{N} \sum_{i=0}^{N} \frac{2TP_i}{2TP_i + FN_i + FP_i} \tag{7}$$

4.1 Simulation

To evaluate the proposed gesture recognition approach in a simulated environment, the Game Engine Unity is used. A suitable human model is placed in a simulated environment and captured by a virtual camera. The camera is positioned 1.5 m above the floor and in a distance of 3 m, 6 m and 9 m to the person. The simulated person performs the previously defined gestures and the results of the gesture recognition based on the virtual camera images (cf. Figure 4) are evaluated. The confusion matrices for the results of the gesture recognition from the three distances are depicted in Figure 3 a) to c).



(a) 3 m distance (b) 6 m distance (c) 9 m distance

Figure 3: Normalized confusion matrix for the results of the gesture recognition and classification during the simulation in Unity. For each gesture and distance 90 images have been evaluated, resulting in a total of seven classes and 630 images per distance.

Figure 4: Camera image captured in the simulated environment.

It can be seen, that the gestures are correctly recognized in the vast majority of the images. In a few images where no person is present, different objects are falsely interpreted as humans by the OpenPose framework and thus a skeleton model is calculated although no person is contained in the image. Due to the joint angles in those false-positive skeleton models, the gesture recognition algorithm recognises either a *das* or *las* gesture.

As the simulated persons do not move and no image noise is simulated, the values of the confusion matrix and the corresponding values of A, S_{macro}, P_{macro} and $F1_{\text{macro}}$ do not decrease with distance. The resulting values of the individual parameters are provided in Table 2.

4.2 Real-world Experiments

The evaluation of the gesture recognition during real world flight tests is analogous to the simulation. The UA hovers at an altitude of 1.5 m to 2.0 m above the floor and captures a person form a distance of 3 m, 6 m and 9 m. Figure 1 (left) shows an exemplary camera image captured with a distance of 9 m between the

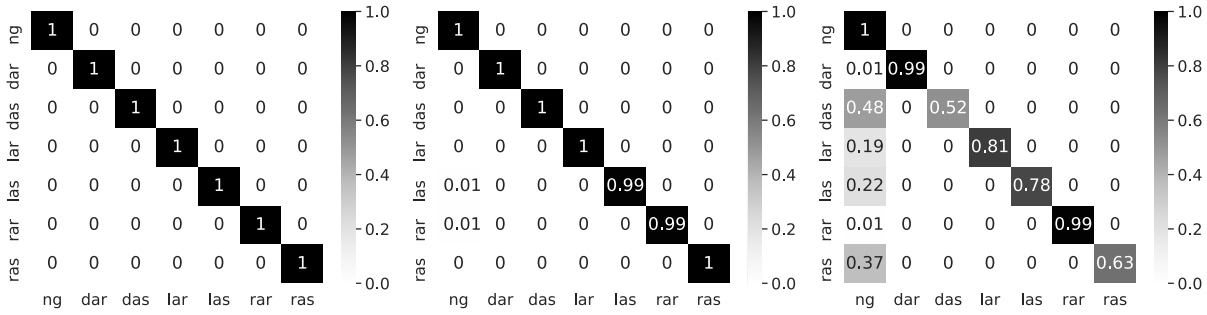
Table 2: Performance measures of the proposed gesture recognition approach when applied to simulation-generated images.

Distance	3 m	6 m	9 m
Precision	1.0	1.0	0.99
Sensitivity	1.0	1.0	0.99
Accuracy	1.0	1.0	0.99
F1-Value	1.0	1.0	0.99

person and the UA. The confusion matrices for the results of the gesture recognition from the three distances are depicted in Figure 5 a) to c). It can be seen, that up to a distance of 6 m the proposed algorithm is capable to identify the vast majority of gestures correctly. In Figure 5 b), only 95 percent of the *rar* gestures were detected correctly, because OpenPose did not provide a valid skeleton model for the remaining 5 % of the images.

Contrary to the simulation, with increasing distance between the UA and the person, the percentage of true positive and true negative recognitions decreases. This can be traced back to noise occurring in the real world images, vibrations affecting the image stability and the increased movement of the UA during hover state, which can reach a peak-to-peak amplitude of 0.2 m. Those factors prevent a detection of the complete skeleton model when the distance between the UA and the person becomes too large. Moreover, the joint angles of a real person do not stay constant like in the simulation but fluctuate a bit, thus decreasing the accuracy further.

In consequence, OpenPose does provide less accurate skeleton models, when the distance between the UA and the persons reached 9 m. As it can be seen in Figure 5 c), when the skeleton model is faulty or missing, our algorithm can not detect a valid gesture or estimates a wrong gesture. Thus, also the values of A, S_{macro}, P_{macro} and $F1_{\text{macro}}$ decrease with increas-



(a) 3 m distance (b) 6 m distance (c) 9 m distance

Figure 5: Normalized confusion matrix for the results of the gesture recognition and classification during the real world flight test. For each gesture and distance 90 images have been evaluated, resulting in a total of seven classes and 630 images per distance.

ing distance. The resulting values of the individual parameters are provided in Table 3.

Table 3: Performance measures of the proposed gesture recognition approach when applied to real world images captured by the on-board camera of an UA.

Distance	3 m	6 m	9 m
Precision	1.0	0.99	0.82
Sensitivity	1.0	0.99	0.92
Accuracy	1.0	0.99	0.82
F1-Value	1.0	0.99	0.83

During the real-world flight experiments, also the response time of the GCS achievable with the presented approach for gesture recognition and the utilized hardware is evaluated. This analysis is based on the time interval between the receipt of a camera image on the GCS and the dispatch of a corresponding flight command to the to the UA. As stated above, we did not evaluate the reaction time of YOLOv4 on mobile hardware, as this has already been subject of various studies.

First, we determine the interval T_{OP}, which indicates the computing time used by OpenPose to calculate the skeleton model. Second, we calculate the interval T_{total}, which additionally includes the computation time of the previously presented algorithm for gesture recognition and for deriving the associated flight command. The resulting intervals are provided in Table 4. It can be seen, that the interval T_{total} has an average value of 47.32 ms, indicating that the gesture recognition can be performed at an update rate of around 20 Hz with the used hardware. However, it must be taken into account that the evaluation is not performed directly on the UA, and thus the total time must be increased by the transmission times of the radio network used.

Table 4: Time interval T_{OP} required to determine the skeleton model and interval T_{total} indicating the overall time required to perceive a person, recognize a gesture and transmit a corresponding flight command to the UA

Interval (ms)	T_{OP}	T_{total}
Minimum	3,00	16,09
Maximum	35,27	101,38
Mean	19,59	47,32
Standard deviation	8,75	16,70

5 CONCLUSIONS

Within this paper, we have presented an approach to enable autonomous UA to perceive and locate persons within their flight path and to perform flight manoeuvres indicated by the perceived person using different gestures. The proposed methodology for gesture recognition is based on a skeleton model of the detected persons and uses the angles between individual limbs to distinguish different gestures. The evaluation of the proposed approach is conducted within a simulation and based on real word flight experiments and reveals an average accuracy of 0.94 for gestures performed in a distance between 3 m and 9 m to the UA.

Within future research, we will focus on increasing the robustness of the gesture recognition, especially when the UA is still further away from a person or the person is partially concealed. Additionally, we will add a 3D-segmentation pipeline as presented in (Kedilioglu et al., 2021) to determine the point cloud corresponding to the perceived persons and calculate a bounding box enclosing each person. An additional spatio-temporal tracking of each person in combination with a suitable path-planning approach will allow the UA to perform more suitable evasion manoeuvres without endangering the persons.

REFERENCES

Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection.

Cai, C., Yang, S., Yan, P., Tian, J., Du, L., and Yang, X. (2019). Real-time human-posture recognition for human-drone interaction using monocular vision. In Yu, H., Liu, J., Liu, L., Ju, Z., Liu, Y., and Zhou, D., editors, *Intelligent Robotics and Applications*, pages 203–216, Cham. Springer International Publishing.

Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., and Sheikh, Y. (2021). Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):172–186.

Chen, Y., Tian, Y., and He, M. (2020). Monocular human pose estimation: A survey of deep learning-based methods. *Computer Vision and Image Understanding*, 192:102897.

Eißfeldt, H., Vogelpohl, V., Stolz, M., Papenfuß, A., Biella, M., Belz, J., and Kügler, D. (2020). The acceptance of civil drones in germany. *CEAS Aeronautical Journal*, 11.

Kassab, M. A., Ahmed, M., Maher, A., and Zhang, B. (2020). Real-time human-uav interaction: New dataset and two novel gesture-based interacting systems. *IEEE Access*, 8:195030–195045.

Kedilioglu, O., Lieret, M., Schottenhamml, J., Würfl, T., Blank, A., Maier, A., and Franke, J. (2021). Rgb-d-based human detection and segmentation for mobile robot navigation in industrial environments. In *VISIGRAPP (4: VISAPP)*, pages 219–226.

Koch, J., Wettach, J., Bloch, E., and Berns, K. (2007). Indoor localisation of humans, objects, and mobile robots with rfid infrastructure. In *7th International Conference on Hybrid Intelligent Systems (HIS 2007)*, pages 271–276.

Le, T.-L., Nguyen, M.-Q., and Nguyen, T.-T.-M. (2013). Human posture recognition using human skeleton provided by kinect. In *2013 International Conference on Computing, Management and Telecommunications (ComManTel)*, pages 340–345.

Lidynia C., Philipsen R., Z. M. (2017). Droning on about drones—acceptance of and perceived barriers to drones in civil usage contexts. *Savage-Knepshield P., Chen J. (eds) Advances in Human Factors in Robots and Unmanned Systems. Advances in Intelligent Systems and Computing*, 499.

Liu, C. and Szirányi, T. (2021). Real-time human detection and gesture recognition for on-board uav rescue. *Sensors*, 21(6).

Maher, A., Li, C., Hu, H., and Zhang, B. (2017). Realtime human-uav interaction using deep learning. In Zhou, J., Wang, Y., Sun, Z., Xu, Y., Shen, L., Feng, J., Shan, S., Qiao, Y., Guo, Z., and Yu, S., editors, *Biometric Recognition*, pages 511–519, Cham. Springer International Publishing.

Medeiros, A. C. S., Ratsamee, P., Uranishi, Y., Mashita, T., and Takemura, H. (2020). Human-drone interaction: Using pointing gesture to define a target object. In Kurosu, M., editor, *Human-Computer Interaction. Multimodal and Natural Interaction*, pages 688–705, Cham. Springer International Publishing.

Monajjemi, M., Bruce, J., Sadat, S. A., Wawerla, J., and Vaughan, R. (2015). Uav, do you see me? establishing mutual attention between an uninstrumented human and an outdoor uav in flight. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3614–3620.

Monajjemi, M., Mohaimenianpour, S., and Vaughan, R. (2016). Uav, come to me: End-to-end, multi-scale situated hri with an uninstrumented human and a distant uav. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4410–4417.

Mosberger, R. and Andreasson, H. (2013). An inexpensive monocular vision system for tracking humans in industrial environments. In *2013 IEEE International Conference on Robotics and Automation*, pages 5850–5857.

Pounds, P. E. I. and Deer, W. (2018). The safety rotor—an electromechanical rotor safety system for drones. *IEEE Robotics and Automation Letters*, 3(3):2561–2568.

Sanna, A., Lamberti, F., Paravati, G., Ramirez, E. H., and Manuri, F. (2012). A kinect-based natural interface for quadrotor control. In *Intelligent Technologies for Interactive Entertainment. 4th International ICST Conference, INTETAIN 2011, Genova, Italy, May 25-27, 2011, Revised Selected Papers*.

Tellaeche, A., Kildal, J., and Maurtua, I. (2018). A flexible system for gesture based human-robot interaction. *Procedia CIRP*, 72:57–62. 51st CIRP Conference on Manufacturing Systems.

Yu, Y., Wang, X., Zhong, Z., and Zhang, Y. (2017). Ros-based uav control using hand gesture recognition. In *2017 29th Chinese Control And Decision Conference (CCDC)*, pages 6795–6799.

Zhang, J., Peng, L., Feng, W., Ju, Z., and Liu, H. (2019a). Human-agv interaction: Real-time gesture detection using deep learning. In Yu, H., Liu, J., Liu, L., Ju, Z., Liu, Y., and Zhou, D., editors, *Intelligent Robotics and Applications*, pages 231–242, Cham. Springer International Publishing.

Zhang, S., Liu, X., Yu, J., Zhang, L., and Zhou, X. (2019b). Research on multi-modal interactive control for quadrotor uav. In *2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC)*, pages 329–334.