Human-AGV Interaction: Real-time Gesture Detection Using Deep Learning

Jiliang Zhang[1,2], Li Peng[1], Wei Feng[1], Zhaojie Ju[2],

Honghai Liu[2]

[1]Jiangnan University, Wuxi 214122, CN
[2]University of Portsmouth, Portsmouth PO1 3HE, UK
penglimail2002@163.com

Abstract. In this paper, we present a real-time human body gesture recognition for controlling Automated Guided Vehicle(AGV) in facility. Exploiting the breakthrough of deep convolutional networks in computers, we have developed a system that can detect the human gestures and give corresponding commands to the AGV according to different gestures. For avoiding interference of multiple operational targets in an image, we proposed a method to filter out the non-operator. In addition, we propose a human gesture interpreter with clear semantic information and build a new human gesture dataset with 8 gestures to train or fine-tune the deep neural networks for human gesture detection. In order to balance accuracy and response speed, we choose MobileNet-SSD as the detection network.

Keywords: Human Gesture, AGV, MobileNet-SSD, Deep Learning.

1 Introduction

Goods production flow in manufacturing plants has been largely and deeply automated in the last decades. To increase efficiency and reduce the cost of manual operators in manufacturing and distributing logistics, companies and organizations use robots as an effective tool. The Automated Guided Vehicle (AGV) is type of effeminate mobile vehicle that is primarily used to move materials from one place to another. AGVs are commonly used in manufacturing plants, warehouses, distribution centers and terminals. For navigation, an AGV system usually uses lane paths, signal paths or signal beacons. Various main sensors are also used in AGV, such as optical sensors, laser sensors, magnetic sensors and cameras.

AGV system has a strict requirement on environment. AGVs cannot work at the place where has no lane or signal. In this way, we hope that we can propose a new control method to break this limitation so that it can be applied to a wide range of scenarios, such as rural areas, urban outside, etc. The Natural User Interface (NUI) has been proposed recently instead of the physical remote. Visual human gesture is one of the most appealing methods to build an AGV system. We present a fast and

accurate detector that finds the hands, faces and bodies of multiple people in RGB images at the frame-rate, which can be used directly as an input to an AGV system.

The contributions of this paper are: (i)We create a new way to control AGV with deep learning. This method can enable AGVs to get rid of environmental constraints and apply them in a wider range of fields. (ii)We propose a filtering algorithm based on high level features, which is effective and low time cost, to filter out visual disturbances from non-operators in the image and implement them in our software program. The software uses a scalable CNN model that can be resized for speed/accuracy trade-off based on MobileNet-SSD; (iii)We propose a novel, simple but effective and semantically clear static gesture detection method for transmitting instruction commands based on the angle of the hand relative to the face; and establish a dataset based on the representation method which contains 8 human gestures.

The rest of the paper is structured as follows:

Section 2 reviews the development of object detection and the related research status of AGV system and human-robot interaction.

Section 3 presents the overall architecture of our system.

Section 4 introduces the network structure we choose and compares it with the current frequently-used object detection network model.

Section 5 describes the proposed method of human gestures and introduces the establishment of the matched dataset.

Section 6 explains the filtering algorithm of the AGV system and the process of filtering out the interference from non-operators.

Section 7 is the experimental results of the actual test of the system.

2 Background

2.1 Object Detection

The purpose of object detection is to identify objects from different backgrounds of complexity and separate the background to complete follow-up tasks such as tracking and recognition. Therefore, object detection is the basic task of high-level understanding and application, and its performance will affects the performance of mid - and high-level tasks directly such as subsequent target tracking, motion recognition, and behavioral understanding.

Object detection is important in the field of computer vision and image processing because of its wide range of applications for video surveillance, intelligent transportation, medical diagnostics and vision guidance. Therefore, it's important to have a robust and fast object detection algorithm. There are two branches particularly compelling in many CNN-based algorithms. The first uses two steps to solve the problem such as R-CNN [1], SPPnet [2], Fast R-CNN [3], and Faster R-CNN [4]. In this series, the first step is to find possible candidate regions, and then predict the corresponding categories and perform a box regression of the boundary candidate regions. Second is single stage series, including You Only Look Once (YOLO) [5], YOLOv2 [6], Single Shot MultiBox Detector (SSD) [7], which aim to remove the region proposal stage and then predict confidence and offset for every default box. The SSD

network provides a new neural network model for deep learning. In this architecture mode, some researchers replaced the front network VGG-16[8] with other types of networks such as Residual-101 [9] and inception[10], which are much deeper and more powerful to achieve a higher accuracy. The deepening of the network structure means more calculation parameters, which will bring the loss of calculation speed. So we hope to use a network to balance calculation speed and accuracy. Recently MobileNet [11] has been proposed to establish a real-time-capable object locator, providing the possibility of implementing neural networks on mobile devices.

2.2 The Development of AGV

The world's first automated guided vehicle was developed by Basrrett Electronics in the United States in 1953. It was converted from a towed tractor with a car hopper. It worked based on the routine of the wire set in the air. In the late 1950s and early 1960s, there were many types of towed AGVs used in factories and warehouses. Recently, there are about 20,000 AGVs in the world running in thousands of large and small warehouses.

Starting from the electromagnetic induction guidance technology of underground embedding in the United Kingdom in 1954, the early AGVs were driven along the signals on the ground. The sensors on the AGV are selected the electromagnetic signals of a certain frequency to provide guidance for the AGV according to the strength of the signal. With the rapid development of electronic technology and microprocessor technology, AGV's intelligent technology has been generally developed. In the late 1980s, wireless guidance technology was introduced into the AGV system, such as laser and inertia guidance, greatly improving the flexibility and accuracy of the AGV system. The introduction of computer technology allows AGV to handle almost all manually controlled material handling processes. Fig.1 shows an example of an AGV from the internet.

Fig. 1. An AGV in the factory.

2.3 Human-robot interaction

Using gestures to achieve human-computer interaction has recently become popular. Many researchers have done relevant research. In[12][13][14], authors use Microsoft

Kinect to capture images of operators which contain RGB and depth data. With this data, localmachine can derive the skeletal model of the operator and match the corresponding vocabulary based on the skeletal features. But RGB-D sensors have frustrating problems, involving the Kinect with its driver and API library makes the system costlier and has a noticeable latency. Motion-based detection is one of the ways of implementation such as waving[15]. These recognition methods work slowly and are vulnerable to frame loss in the video stream, so they are hard to apply in real-time detection. Some researchers use more obvious objects, such as arm gestures[16] and colored gloves[17], to avoid the effects of the environment.

3 System Architecture Overview

Fig. 2 shows high level architecture of the system. Images are captured by on-board camera, then forward through pretrained neural network model. The outputs of the model are boxes including face, hands and body. According to the position of these boxes, local machine can interpret the gesture operator want to express, and then send matched command to the AGV. After receiving the command, the AGV will send feedback message to the local machine. The entire system works in a Wi-Fi environment.

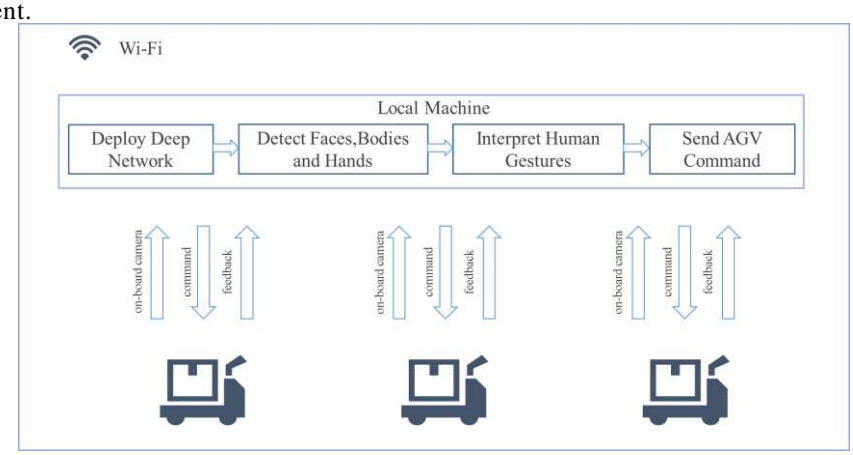


Fig. 2. High level architecture.

4 CNN Model

The network model we choose is MobileNet-SSD, which is an improvement of standard SSD model. Single Shot MultiBox Detector (SSD) is one of the fastest algorithms in the current object detection field, which uses fully convolutional neural network to detect all scaled objects in an image. This method discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. The network outputs a predicted score for each object category in each default box and adjusts the output box to better match the shape of the object.

In addition, the network extracts features from feature layers with different resolutions and combines prediction results of multiple feature maps together to identify natural objects in different sizes.

The architecture of the network presents in Fig.3. The front layers of standard SSD is VGG-16 while the network we chose is MobileNet. According to [11], the author has made the comparison of MobileNet to other popular models. The result shows in Table 1. We can see, the parameters of the MobileNet are greatly reduced, and the accuracy is reduced a little. We can conclude that MobileNet loses a small amount of accuracy to achieve a higher speed increase, which is the key of real-time detection. The system works based on three objects - hands, faces and bodies. In order to cater for the requirement, we changed the number of filters in the last convolutional layer of the model.

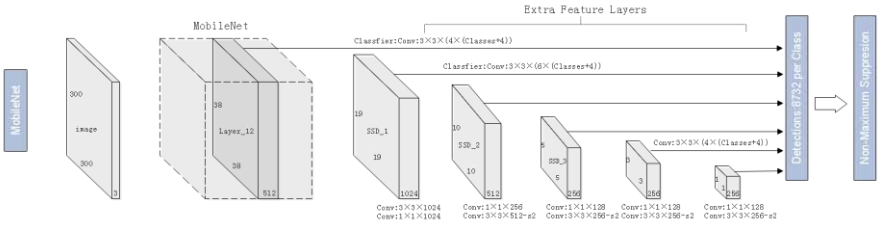


Fig. 3. Architectures of MobileNet-SSD

Table 1. MobileNet comparison to popular models

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
MobileNet	70.6%	569	4.2
GoogleNet	69.8%	1550	6.8
VGG 16	71.5%	15300	138

5 Human Gestures

5.1 Gesture Interpreter

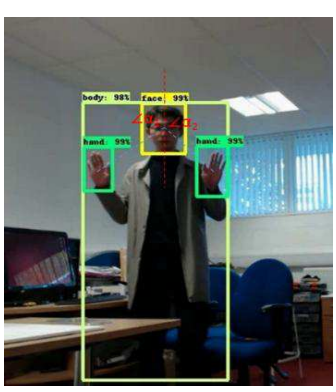
Many researchers have proposed multiple ways of gesture representation. In[18][19][20], researchers have presented different sets of gestural vocabularies. Due to little semantic information and real-time problems, they are not suitable for an AGV system. Some researchers defined some motion-based gestures such as waving-based gestures[15], based on sequence of different postures for frames. However, these detection methods are too slow to widely meet the requirements for real-time detection under the conditions of existing hardware devices in industrial environment. Relying on skin detection enabled detection of a user's arms and to generate richer commands[16], but skin detection lacks robustness and is not always feasible. The

6

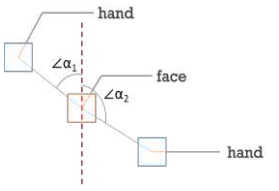
gestures we need are those stable enough to be uninterrupted by environmental factors, they should contain clear semantic information, so that users do not need calibration or training. And they should be easy to be understood by AGV system. In this way, we use static (posture) not dynamic gesture recognition in our proposal, make the system identify the command expressed by each frame and regard every four frames as a sequence. If the four commands contained in the sequence are all same, the command will take effect, which can avoid the unexpected effect caused by the loss or misidentification of certain frame to the current AGV state.

5.2 Gesture Design

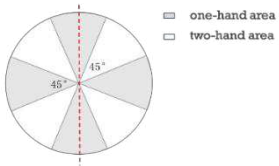
Our static gesture detection works based on the angle of each hand box's center to that of face's box. Different angles represent different gestures. We define $\angle\alpha_1$ as the angle between the line from the center point of the right-hand box to the center point of the face box and the line in the center of the face box. While $\angle\alpha_2$ is defined as the left one.(Fig.4) These two angles can present 8 human gestures with the different ranges of each one. We believe that gestures should have clear semantic information in practical applications, so we divide the gesture set into four one-hand gestures and four two-hand gestures. One-hand gestures present AGV movement commands(move forward, move backward, move left, move right), and two-hand gestures indicate function commands(lift up, lift down, turn CW 180°, focus on operator) as illustrated in Fig.5. We divide the 360° two-dimensional area into eight areas in each 45°, which represent the one-hand gesture area and the two-hand gesture area alternately, aiming to overcome the wrong interpretation of the instruction in the process of gesture formation caused by a high degree similarity of the same type gestures. Table 2 summarizes the correspondence between postures and AGV controlling commands. 'Stop' is a default command when the operator gesture does not satisfy any threshold mentioned in Table 2.



(a)



(b)



(c)

Fig. 4. (a)(b) Demonstrates of how the angles $\angle\alpha_1$ and $\angle\alpha_2$ are determined from the bounding boxes. (c)Areas of controlling commands.

Table 2. The correspondence between postures and the AGV controlling commands.

$\angle\alpha_1$	$\angle\alpha_2$	corresponding command
One-hand gestrues（Movement commands）		
0°：22.5°	Or 0°：22.5°	Move forward
67.5°：112.5°	--	Move left
157.5°：180°	Or 157.5°：180°	Move backward
--	67.5°：112.5°	Move right
Two-hand gestures（Function commands）		
22.5°：67.5°	22.5°：67.5°	Lift up
22.5°：67.5°	112.5°：157.5°	Turn CW 180°
112.5°：157.5°	22.5°：67.5°	Focus on operator
112.5°：157.5°	112.5°：157.5°	Lift down

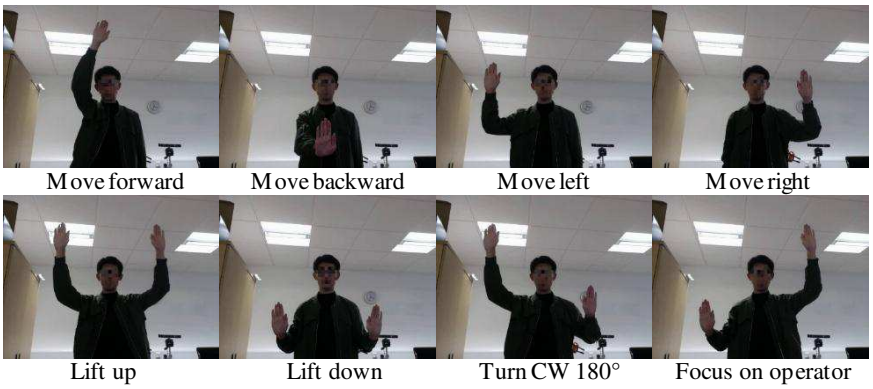


Fig. 5. Different gestures are designed to control the AGV.

6 Classification Filtering

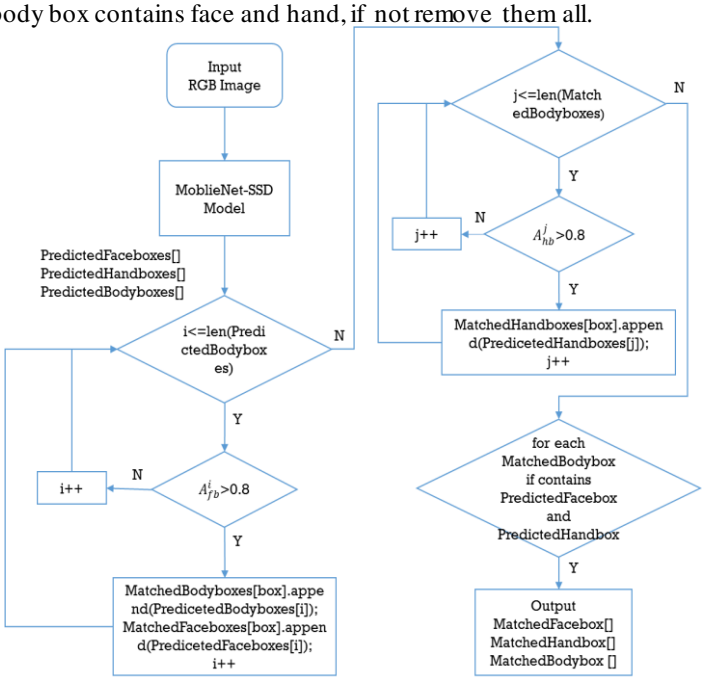
Considering the actual working environment of AGV, the video frames captured by the on-board camera may contain more than one person, which can be recognized as misleading commands. We use high level features to filter the output classification to solve this problem. The main idea of classification filtering is to match the results of the model and filter out the output boxes that have not been matched. A complete gesture consists of three parts: hands, face and body. If the output box cannot be part of the three parts then it will be filtered out. We use two parameters A_{hb}, A_{fb} to indicate how well the hand and face match the body. The parameters are defined as shown:

8

$$A_{hb} = \frac{S_{hand \cap body}}{S_{hand}}$$

$$A_{fb} = \frac{S_{face \cap body}}{S_{face}}$$

The agreement score A_{hb} for the functionality hb is calculated with the square of the common part of hand output box and body output box and the square of hand output box. In this way, the agreement score A_{fb} for the functionality fb is calculated with the square of the common part of face output box and body output box and the square of face output box. If the value of A_{hb} (or A_{fb})is close to 1, we can think that the hand (or face) output box is in the body output box, so that these two output boxes can be regarded matched. Fig.6 shows the process of filtering. MobileNet-SSD model outputs predicted boxes for faces, hands and bodies. The first loop matches the faces and bodies. The second loop matches the hands and bodies. Finally check if each matched body box contains face and hand, if not remove them all.



In order to enable the network to adapt to different distance to the operator, all these images were taken in the range from 0.5m to 5m. We used MobileNet-SSD model pretrained on COCO 90-class dataset, and got applied model after 200000 steps of iteration. The ground station is a desktop PC (with an Intel Core i7-3820 @ 3.60GHz × 8 and equipped with NVIDIA GPU GeForce GTX 690), which performs not so good in calculation but closes to industrial computers.

7.2 Experiments on Human-AGV Interaction

We create a test dataset with the same distribution as the training dataset for evaluation, which contains all eight gestures. There are 200 images for each gesture. The test frames in this section all contain single operator without background interference from non-operators. Table 3. presents the accuracy of MobileNet-SSD and some popular object detection models working based on our proposed gesture method. From table 3. obviously, MobileNet-SSD achieves the real-time requirement for gesture detection beside preserving the detection accuracy in terms of gesture interaction. InceptionV2-SSD are also fast enough with 14.23 fps to be applied in practice, but MobileNet-SSD can work at a higher speed, which means it can be applied to a wider scope of environments and have less requirement for hardware devices.

Table 3. The accuracy of MobileNet-SSD compared with popular models

network model	move forward	move backward	move left	move right	lift up	lift down	turn CW 180°	focus on operator	total	fps_mean
MobileNet-SSD	88%	92%	**95.5%**	95.5%	75.5%	95%	82.5%	86.5%	88.81%	**27.74**
InceptionV2-SSD	93%	**100%**	87.5%	90.5%	85.5%	97%	91%	89%	91.69%	14.23
Faster-rcnn-InceptionV2	96.5%	95.5%	91%	**100%**	94%	**100%**	**99.5%**	98.5%	**96.88%**	1.17
Faster-rcnn-resnet50	**97%**	76.5%	94.5%	92.5%	**97.5%**	**100%**	**99.5%**	**99.5%**	94.63%	0.29

7.3 Experiments on Classification Filtering

We have established a dataset specifically for evaluating the filtering effect, which has 2000 images. Each gesture has 250 images, of which 20% have interferences from non-operators. Table 4. lists the comparison results of the filtering algorithm applied in MobileNet-SSD model. It can be clearly seen that our proposed algorithm can effectively eliminate interference from non-operators, because the algorithm utilizes high level features without involving too many calculations, so it has less impact on real-time performance. There are comparisons of two frames in Fig.7.

Table 4. The results of the filtering algorithm

	move forward	move backward	move left	move right	lift up	lift down	turn CW 180°	focus on operator	total	fps_mean
unfiltered	75.6%	82%	78.8%	80.4%	64%	82.4%	72.4%	73.6%	76.15%	27.76
filtered	88%	90%	92%	91.2%	72%	97.6%	81.2%	83.6%	86.95%	27.44

10

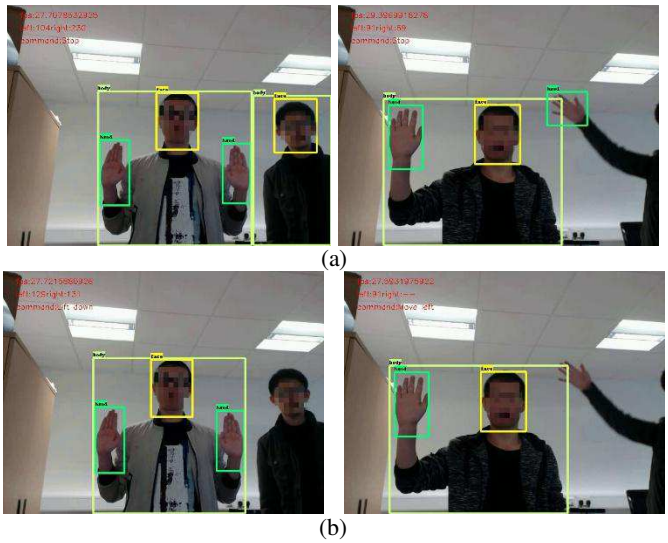


Fig. 7. (a)The output frame without being filtered. (b)The output frame after being filtered.

8 Conclusion

In this paper, we introduce a system to control AGV through a series of human gestures successfully, which can work at a high accuracy and fast speed even if the hardware devices are not so powerful. With our new collected dataset built for human gestures, we design an interpreter mapping each gesture to a controlling command. In addition, the gesture method and classification filtering algorithm we designed get better results in the experiment, which can achieve real-time and preserve high gesture detection accuracy.

Acknowledgements

This research was supported by the 111 Project(B12018) and Jiangsu Planned Projects for Postdoctoral Research Funds(1601085C). We thank our colleagues from Portsmouth University, England and Jiangnan University, China, who provided insight and expertise that greatly assisted the research.

Reference

1. Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik: Rich feature hierarchies for accurate object detection and semantic segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2014, pp. 580-587 (2014)
2. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: Computer Vision - ECCV 2014 , pp. 346-361 (2014)
3. Ross Girshick: Fast R-CNN. In: The IEEE International Conference on Computer Vision (ICCV) 2015, pp. 1440-1448 (2015)
4. Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun: Faster R-CNN: Towards real-time object detection with region proposal networks. In: IEEE Transactions on Pattern Analysis & Machine Intelligence 2017, pp. 1137-1149, vol. 39 (2017)
5. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi: You Only Look Once: Unified, Real-Time Object Detection. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016, pp. 779-788(2016)
6. Redmon, Joseph, and Ali Farhadi, "YOLO9000: better, faster, stronger." In: arXiv preprint arXiv:1612.08242 (2016).
7. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg: SSD: Single shot multibox detector. In: Computer Vision - ECCV 2016 , pp. 21-37 (2016)
8. Simonyan, Karen, and Andrew Zisserman: Very deep convolutional networks for large-scale image recognition. In: arXiv preprint arXiv:1409.1556 (2014).
9. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun: Deep residual learning for image recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016, pp. 770-778 (2016)
10. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich: Going Deeper With Convolutions. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015, pp. 1-9 (2015)
11. Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. In: arXiv preprint arXiv:1704.04861(2017)
12. M. Lichtenstern, M. Frassl, B. Perun, and M. Angermann: A Prototyping Environment for Interaction Between a Human and a Robotic Multi-agent System. In: 7th Annual ACM/IEEE International Conference on Human-Robot Interaction (HRI), ser. HRI '12. New York, NY, USA: ACM 2012, pp. 185-186 (2012)
13. Sanna, A., Lamberti, F., Paravati, G., Manuri, F.: A kinect-based natural interface for quadrotor control. In: Entertainment Computing 4(3) 2013, pp. 179-186 (2013)
14. T. Naseer, J. Sturm, and D. Cremers: FollowMe: Person following and gesture recognition with a quadrocopter. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2013, pp. 624-630 (2013)
15. M. Monajjemi, S. Mohaimenianpour, and R. Vaughan: UAV, come to me: End-to-end, multi-scale situated HRI with an uninstrumented human and a distant UAV. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2016, pp. 4410-4417 (2016)
16. T. Sun, S. Nie, D. Y. Yeung, and S. Shen: Gesture-based piloting of an aerial robot using monocular vision. In: IEEE International Conference on Robotics and Automation (ICRA) 2017, pp. 5913-5920 (2017)

17. J. Nagi, H. Ngo, L. M. Gambardella, and G. A. D. Caro: Wisdom of the swarm for cooperative decision-making in human-swarm interaction. In: IEEE International Conference on Robotics and Automation (ICRA) 2015, pp. 1802-1808 (2015)
18. W. S. Ng and E. Sharlin: Collocated interaction with flying robots. In: 20th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pp. 143-149 (2011)
19. F. Taralle, A. Paljic, S. Manitsaris, J. Grenier, and C. Guettier: A Consensual and Non-ambiguous Set of Gestures to Interact with UAV in Infantrymen. In: 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems, ser. CHI EA '15. New York, NY, USA: ACM, pp. 797-803 (2015)
20. J. R. Cauchard, J. L. E, K. Y. Zhai, and J. A. Landay: Drone & Me: An Exploration into Natural Human-drone Interaction. In: ACM International Joint Conference on Pervasive and Ubiquitous Computing, ser. UbiComp '15. New York, NY, USA: ACM, pp. 361-365 (2015)