Real-time Hand Gesture Recognition Based on Deep Learning in Complex Environments

Weixin Wu[1], Meiping Shi[1], Tao Wu[1], Dawei Zhao[1], Shuai Zhang[1], Junxiang Li[1]
1. College of Artificial Intelligent Science, National University of Defense Technology, Changsha, 410005
E-mail: fdwwxzxd@163.com, shimeip@163.com, wt.cs@163.com

Abstract: Real-time hand gesture recognition in complex environments has many challenges, such as poor real-time performances and robustness to environmental changes. This paper takes the hand gesture control of the unmanned vehicle as the application background, and focuses on the gesture detection and recognition of video streams based on deep learning in the complex environment. In this paper, we detect the hand in a complex environment by training the ssd_mobilenet model, and initialize the tracking with kalman filter. Then, we detect the hand keypoints by following the architecture of Convolutional Pose Machines(CPMs), in order to obtain the belief maps for all keypoints that are used as the train sets of Convolutional Neural Networks(CNNs). Finally, based on the results obtained by our classification, this paper proposes a method of multi-frame recursion to minimize the influences of redundant frames and error frames. In this paper, eight kinds of gestures for controlling vehicle are identified. The experimental results show that our method can successfully realize real-time hand gesture recognition in the video streams. The recognition accuracy can reach 96.7%, and the average recognition speed reaches 12 fps, which basically meets the real-time requirements and successfully applys to mobile terminals such as TX2 for engineering practice.

Key Words: Ssd_mobilenet; Convolutional Pose Machines(CPMs); Hand keypoints; Convolutional Neural Networks(CNNs); Multi-frame recursion.

1 INTRODUCTION

The development of artificial intelligence technology has greatly promoted the process of commercialization of driverless. At present, the autonomous driving technology is not mature enough, and it still needs people to assist driving. Therefore, human-computer interaction technology is the key to marketization of driverless. Now, the natural human-computer interaction methods mainly include voice interaction and gesture interaction etc. The voice interaction technology is relatively mature at present, but it is still susceptible to noise interference in the wild environment. While the gestures contain a lot of information that can convey semantics, emotions and conform to human daily life habits, that are the research focus of human-computer interaction technology.

With the rapid development of computer vision technology, gesture recognition based on vision has low cost and convenient remote interaction [1], so it is the focus of gesture recognition. Vision-based gesture recognition methods can be roughly divided into two types: one is to use Kinect,Leapmotion and other depth cameras [2] [3]or neural networks to obtain image depth information, such as position information of gestures; the other one is to split the gesture from the background by traditional methods and then extract the apparent image characteristics of the gesture itself to perform gesture recognition.

There are several challenges in gesture recognition based on the video streams:

1)The background is complex and easily affected by light [4].

2)There will be a large number of redundant gesture pictures.

3)The multi-view of the human hand and the inconsistent shape size.

In this paper, there are two contributions : First, we propose to fine-tune the ssd_mobilenet model [5] [6] to train our manually labeled data sets in order to detect the position of the human hand in real time, and remove a part of the complex background environment. Second, we add the background class and multi-frame recursion: take the belief maps of hand keypoints and background class as CNNs' input to get the classification result; Then, the classification result is recursively calculated multiple frames and the final classification result is obtained. Compared with the traditional classification method, our method achieves good performances.

The remainder of this paper is organized as follows. Section II introduces the related works. Section III introduce the system framework and details of our method. Discussion and analysis of experimental results in Section IV. Finally, Section V summarize the paper and future work.

2 RELATED WORK

The video-based gesture recognition process can be roughly divided into three steps:

1)Hand detection and segmentation. The mainstream direction of the traditional methods is to establish a skin color model [7], segmenting the hand from the complex background.

This work is supported by National Nature Science Foundation of China grant number 61790565.

978-1-7281-0106-4/19/$31.00

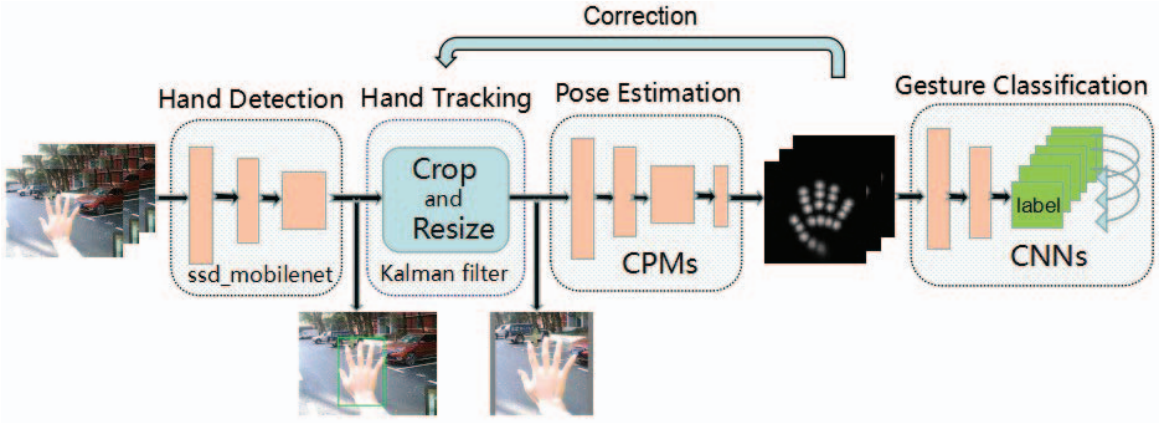


Figure 1: System framework

2)Tracking of the hand. The purpose of tracking is mainly to be able to accurately locate the position of the hand in each frame of the vedio in real time, showing the characteristic trajectory of the hand.

3)hand classification and recognition. The traditional method is to match the candidate image with the template image similarity, but this method is susceptible to complex background.

YoungJoo Lee et al. [8] employ chain code and entropy analysis to segment gesture regions and contours from complex backgrounds. The recognition accuracy can reach more than 90%, but the influence of illumination is obvious. The image depth information can effectively solve the interference in the complex illumination background. And because of the development of deep learning in recent years, researchers have focused more on the image depth information of gestures. Due to the low accuracy of Kinect's depth datas and the small detection range of leapmotion, the researchers try to use the deep learning method to output the position information of hand keypoints [9] [10]. Oberweger M et al. [11]not only apply CNNs to directly output the joint positions ,but also combine a constrained prior hand model and a specific refinement stage to increase the joint localization accuracy. Shih-En Wei et al. [12] design a sequential multi-stage convolution network which directly operates on belief maps from previous stages,that is now one of the best body and hand 2D pose estimation methods. Tomas Simon et al. [13] propose a multi-camera system to train the hand keypoint detector that effectively solves the occlusion problems of the keypoints, and output the 3D position of hand keypoints.

In recent years, many methods focus on many spatio-temporal features which capture shape, appearance, and motion cues via image gradientsand optical flow. K. Simonyan et al. [14] propose a dual-flow CNN that combines spatial and temporal networks to verify that CNNs trained on multi-frame dense optical streams with smaller training data sets can get very good performances. Tran et al. [15] employ deep 3-dimensional convolutional networks (3D ConvNets) to learn spatiotemporal features for videos. In order to solve many challenges of human computer interaction in real-world systems, Pavlo Molchanov et al. [16] apply a recurrent three dimensional convolutional neural network to detect and classify dynamic gestures from multi-modal data. These vision-based methods mostly require expensive depth camera equipments or are difficult to meet the real-time requirements of engineering practices. Our proposed method can be effectively migrated to engineering practice for the vehicle control.

3 METHOD

The gesture interaction control for unmanned vehicles mainly faces two major challenges: the complex lighting conditions and poor real-time performances. In order to solve these problems, this paper first proposes to train our own hand detection data set based on the ssd_mobilenet model to obtain the detection box $\{cx, cy, w, h\}$ of the hand and initialize the hand position. Then the square patch is cropped at the hand position in the images, that can avoid the full image search and speed up the recognition speed. Since the ssd_mobilenet detection method does not accurately detect the hand in every frame, we propose a method of presetting the hand keypoints in the square patch of images and using the Kalman filter to track the hand keypoints. In order to obtain the characteristics which are easy to classify, we follow CPMs to output the belief map of each keypoint, that remove the influences of illumination and other factors. Subsequently the image features are extracted by CNNs. Then the multi-frame classification results are recursively counted to obtain the final results. The system framework includes: hand detection, hand keypoints tracking, belief maps output, CNNs classification and multi-frame prediction (see Fig. 1).

3.1 Detect hand by ssd_mobilenet model

This paper proposes a method of migration learning to fine-tune ssd_mobilenet_v1_coco model to detect hand. The output of the convolutional layer in Single Shot MultiBox Detector (SSD) is convolved with two dif-

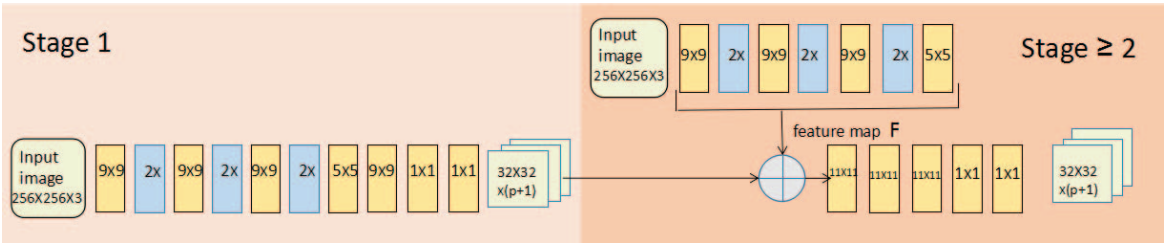


Figure 2: Structure of CPMs

ferent 3×3 convolution kernels, one for the classified confidence and the other one for the localization of the regression location called bounding-boxes $\{cx, cy, w, h\}$. The ssd_mobilenet model is based on SSD to replace the Backbone network(VisualGeometryGroup16) with a lightweight deep network model called mobilent that can improve detection speed and suit engineering practice on mobile terminals such as TX2.

We normalize each 640 × 480 image captured by the original video stream to size 320 × 280. Then, we take each 320 × 280 image as the input of the trained ssd_mobilenet model, and integrate the offset value of each border and the score of the classification. We get the bounding-boxes D_1 $\{cx, cy, w, h\}$ of the hand in the original frame(640 × 480). The speed of ssd_mobilenet model operates at 30 frames per second (FPS) on 4G GPU, which meets both realtime and high precision.

3.2 Hand keypoint estimation and tracking

For the hand keypoint estimation, we follow the CPMs to predict the confidence map for each keypoint. We denote each hand keypoint as $Y_p = (x, y)$ in the image,there are 21 keypoints in a hand(see Fig. 3), denoted as $Y = (Y_1, \cdots, Y_P)$, $P = 21$. We follow the center of the bounding-boxes $C(cx, cy)$,and preset 21 keypoints $Y' = (Y'_1, \cdots, Y'_{21})$ in the detection frame.Then we crop the image area $\mathbf{D}_{CPMs}$ $\{cx, cy, w_1, h_1\}$ with the area 2 × D_2(the minimum area surrounding the 21 keypoints) as input of the CPMs.

In the first stage $\mathbf{S}^1$ of CPMs (see Fig. 2), we normalize the image area $\mathbf{D}_{CPMs}$ $\{cx, cy, w_1, h_1\}$ obtained from the detection to size 256×256 as the original input image of the CPMs. The feature map extraction is followed by the first prediction stage that produces all the beliefs of part $p + 1$, $\mathbf{B}^1 = \left\{B_1^1 \cdots, B_{P+1}^1\right\}$, $P = 21$, so that the belief maps $\mathbf{B}^1 \in R^{w_1 \times h_1 \times (P+1)}$ for all the keypoints can be obtained(P parts plus one for background), which are 22 belief maps 32 × 32. Then, in all remaining stages $\mathbf{S}^t$, $t \geq 2$, the belief maps of the previous stage $\mathbf{S}^{t-1}$ is connected to the image feature $\mathbf{F}$, in order to produce a more accurate new belief maps $\mathbf{B}^t$. After 6 prediction stages, we adjust the belief maps 32 × 32 of 21 hand keypoints to the original image size of 256 × 256. And finally, the accurate new belief maps $\mathbf{B}^t \in R^{w_1 \times h_1 \times P}$ and the position $Y_c = (Y_1, \cdots, Y_P)$ of 21 hand keypoints can be obtained.

In order to carry out continuous position estimation for the hand, this paper uses the Kalman filter to track 21 hand keypoints.We take the preseted position of 21 keypoints $Y' = (Y'_1, \cdots, Y'_{21})$ as the values of the Kalman filter at time instant $k - 1$, and predict $Y'_k = Y'_{k-1} = Y'$ as the position of keypoints at the current time instant k in Eqn 1. We take $Y_c = (Y_1, \cdots, Y_P)$ obtained by CPMs as the measured value at current time instant k. The Kalman gain K_k can be calculated by Eqn. 2 and Eqn. 3. We further update and correct the position in Eqn. 4, in order to obtain a more accurate new estimate Y_k^r, which is used to calculate and update the image region $\mathbf{D}_{CPMs}$ $\{cx, cy, w_1, h_1\}$. If the tracking fails, we restart detection and perform tracking initialization. Finally, we continuously output the belief maps $\mathbf{B}^t \in R^{w_1 \times h_1 \times P}$ of 21 hand keypoints, thereby eliminating the influence of complex background and ensuring the real-time performance. This paper expects to extract features from these belief maps for accurate classification. For this classification is easy to obtain a large number of train sets, we must think of CNNs.

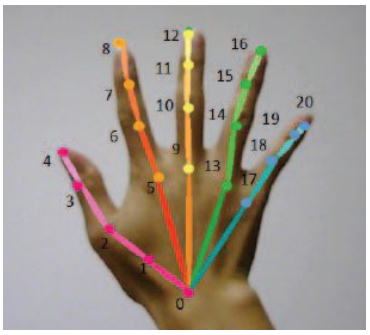


Figure 3: Definition of 21 hand keypoints

$$Y'_k = A \cdot Y'_{k-1} + B \cdot u_{k-1}. \tag{1}$$

$$P_k = A \cdot P_{k-1} \cdot A^T + Q. \tag{2}$$

$$K_k = P_k \cdot H^T \cdot (H \cdot P_k \cdot H^T + R)^{-1}. \tag{3}$$

$$Y_k^r = Y'_k + K_k \cdot (Y_c - H \cdot Y'_k). \tag{4}$$

3.3 CNNs classification and multi-frame prediction

The AlexNet network [17] consists of 5 convolutional layers and 3 fully connected layers. Compared with other CNNs (LeNet,GoogLeNet), it has the advantages of shallow network structure, few parameters and strong generalization abilities. And the fine-tuning method can

(a) Indoor pictures of 8 gestures

(b) Outdoor pictures of 8 gestures

(c) Redundant pictures generated during handover

Figure 4: Sample pictures of data sets.(Labels are defined in turn as: go forward 0 , go backward 7, turn left 2, turn right 3, speed up 4, slow down 5, stop 1 and hide 6 in (a) and (b))

make the AlexNet learn the characteristics of new data sets fast, so this paper proposes a method to increase the background class for fine-tuning training by the AlexNet, and multi-frame recursive results.

We divide the belief maps of the eight gestures into eight types of images, each of which has 500 images (train set: validation set = 8: 2). The final validation accuracy is 100%, indicating that the fine-tuning AlexNet has a good effect of feature extraction. How to solve the problem of redundant frames in the video streams is the key to real-time gesture recognition. We propose to add a background class. Firstly, the redundant frames generated in the eight gesture switching processes are also outputted their belief maps by CPMs, that is grouped into a background class with a total data set of 600. Then we retrain the AlexNet and collect total 9 classes(8 gestures plus 1 background). After 30 generations of training, the final validation accuracy is 99.3%.

In this paper, the belief maps $\mathbf{B}^t \in R^{w_1 \times h_1 \times P}$ outputted by CPMs are used as the input of the trained AlexNet(see Fig. 5) to obtain the label of each frame. We have considered that the background class is easily misjudged, and the project actually requires 100% accuracy rate . Therefore, this paper proposes a multi-frame recursive method to eliminate redundant frames and error frames. We define a series of frames produced by the video stream as $\{S_1, S_2, \cdots, S_F\}$. Starting from the first frame, we takes the labels of n frames as a set $L = \{L_{S_1}, L_{S_2}, \cdots, L_{S_n}\}$. Then we count the class with the highest frequency of the same label in set L as the result of these n frames. Through our method, although the recognition speed drops slightly, we effectively improve the accuracy and stability of gesture recognition in the video streams.

4 EXPERIMENTS

4.1 Experiment platform and data sets

In engineering, we migrate the method to the mobile TX2 and placed a USB camera on top of the helmet. This

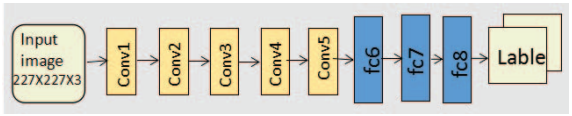


Figure 5: Structure of trained CNNs

Figure 6: Gesture recognition helmet

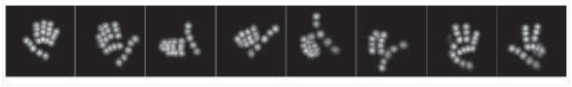
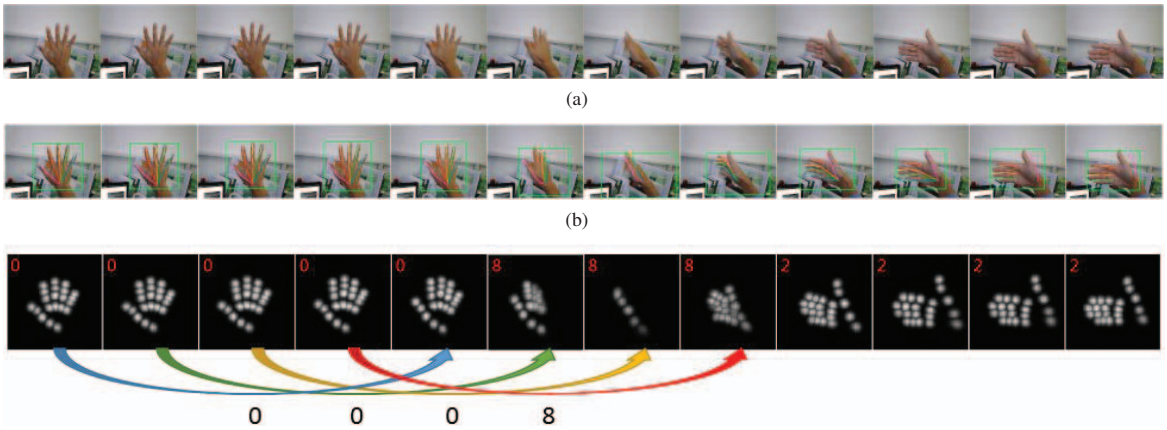


Figure 7: Belief maps of 8 gestures

enabled us to implement a low-cost, easy-to-wear gesture recognition helmet(see Fig. 6). We used the helmet to collect image data and experiments.

Combined with our habits, we define 8 gestures: go forward, go backward, turn left, turn right, speed up, slow down, stop and hide, for controling the unmanned vehicle. In the complex indoor and outdoor environment: sunlight, cloudy, and wild, we collect 4600 pictures of 8 gestures switching between each other. Each of these types of gesture pictures is 500, and the total background class is 600. Part of the collection is shown in Fig. 4. And we can get belief maps for 8 gestures with CPMs.(see Fig. 7)



(a)

(b)

(c)

Figure 8: (a) is original pictures of hand. (b) is the effect of hand detection and keypoints tracking. (c): After CNNs classification, we get the label of each picture, we further carry out multi-frame recursion

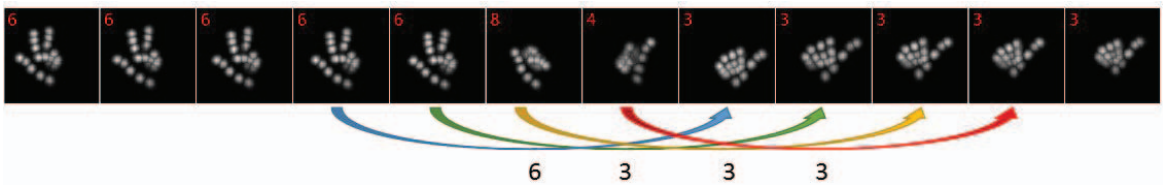


Figure 9: During the switching process of gesture 6 and gesture 3, the background class 8 is misidentified as gesture 4, and we use this method to eliminate the error frame.

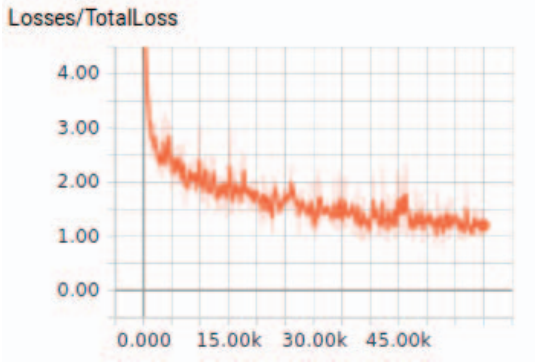


Figure 10: Training loss curve

4.2 Hand detection model training

We manually labeled 1000 hand images in different environments and poses as train set from our own data sets (train set: validation set: test set = 7.6: 1.4: 1), when the learning rate is 0.0005, after 60000 iterations, the total loss stabilized at 1.211 (see Fig. 10) and the Mean Average Precision (mAP) was 0.96.

4.3 Classification result analysis

From the original picture to the final result(see Fig. 8), the average recognition speed of the experiment generally floats at 10-13 fps, so we take the continuous 12 frames in the process of gesture switching in the video streams for analysis. After hand detection and keypoints tracking, we use the belief maps obtained by CPMs to perform AlexNet classification and display the classification results on the belief maps. After many experiments, the number of the redundant frames generated by gesture switching are about 1-3 frames at 10-13fps, so we recursively take n=5 frames in order to minimize the impact of redundant frames. Even with a very small number of misclassified frames, we can exclude the effects in this way, as shown in Fig. 9.

Table 1: Comparison of different methods

Method	Single frame classification time (t/s)	Accuracy	Average FPS
Hog+svm	0.15	87%	4.1
VGG19	0.023	97%	9.7
Ours	0.006	96.7%	11.6

4.4 Comparative experiment

We capture 300 video stream frames generated by switching between 8 gestures from the video streams as test sets. This paper compares the traditional classification method hog+svm with the deeper network structure VGG19 than Alexnet in the same test environment.

The traditional method is far less than the deep learning model in terms of average fps and accuracy, and it is not possible to classify background classes and 8 gestures very well. The VGG19 network is about the same as the AlexNet in accuracy, but not as good as the AlexNet net-

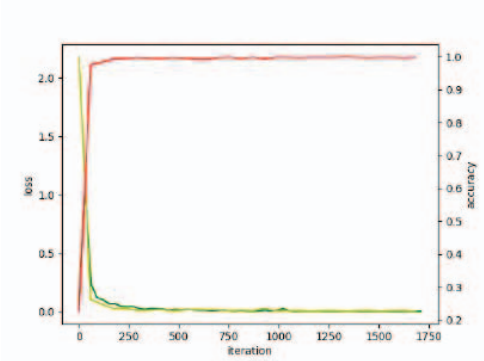


Figure 11: Red: accuracy of validation set. Green: loss value of validation set. Yellow: loss value of train set

work on average fps. The comparative experimental results of these three methods are shown in Table 1. When we fine-tune AlexNet, the effect is the best in learning rate 0.0006. On the basis of fine-tuning, the loss value of the train set and validation set can easily drop to a low point, and the convergence speed is faster(see Fig. 11).

5 Conclusion

This paper proposes a method for real-time recognition of video streams based on deep learning in complex environments. It is applied to human-computer interaction control of vehicle in engineering. The ssd_mobilenet model is used to effectively detect the hand, and the Kalman filter is used to track the hand keypoints. We use the CNNs to classify the keypionts belief maps which are outputted by the CPMs. The proposed algorithm has an accuracy of more than 96% for gesture recognition in complex environments, effectively reduces the false alarm rates. The setting of the background class and the multi-frame recursion method make us more reliable in the practical application of the project. The recognition speed achieves on the TX2 is 2fps, and the recognition accuracy is about 97%, which basically meets the engineering practice requirements. In our future work, we will focus on expanding our data sets to make them more robust and improve the computational efficiency of mobile devices such as TX2 for better real-time performances.

REFERENCES

[1] E. Ueda, Y. Matsumoto, M. Imai, and T. Ogasawara, Hand pose estimation for vision-based human interface, in IEEE International Workshop on Robot and Human Interactive Communication, 2001. Proceedings, 2002, pp. 473478.

[2] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, Robust part-based hand gesture recognition using kinect sensor, IEEE Transactions on Multimedia, vol. 15, no. 5, pp. 11101120, 2013.

[3] C. Keskin, F. Kra, Y. E. Kara, and L. Akarun, Real time hand pose estimation using depth sensors, in IEEE Inter- national Conference on Computer Vision Workshops, 2011, pp. 12281234.

[4] E. Ohn-Bar and M. M. Trivedi, Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations, IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 6, pp. 2368 2377, 2014.

[5] 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, 2016, pp. 21 37.

[6] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, Mo- bilenets: Efficient convolutional neural networks for mobile vision applications, 2017.

[7] H. Francke, J. Ruiz-Del-Solar, and R. Verschae, Real-time hand gesture detection and recognition using boosted classifiers and active learning, in Pacific-Rim Symposium on Image and Video Technology, 2007, pp. 533547.

[8] J. Lee, Y. Lee, E. Lee, and S. Hong, Hand region ex- traction and gesture recognition from video stream with complex background through entropy analysis, Conf Proc IEEE Eng Med Biol Soc, vol. 2, no. 2, pp. 15131516, 2004.

[9] U. Iqbal, M. Garbade, and J. Gall, Pose for action - action for pose, 2016

[10] V. Ramakrishna, D. Munoz, M. Hebert, J. A. Bagnell, and Y. Sheikh, Pose machines: Articulated pose estimation via inference machines, vol. 8690, pp. 3347, 2014.

[11] M. Oberweger, P. Wohlhart, and V. Lepetit, Hands deep in deep learning for hand pose estimation, CoRR, vol. abs/1502.06807, 2015.

[12] S. E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, Convolutional pose machines, pp. 47244732, 2016.

[13] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, Hand keypoint detection in single images using multiview bootstrapping, in IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 46454653.

[14] K. Simonyan and A. Zisserman, Two-stream convolutional networks for action recognition in videos, vol. 1, no. 4, pp. 568576, 2014.

[15] T. Du, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in IEEE International Conference on Computer Vision, 2015, pp. 44894497.

[16] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural networks, in Computer Vision and Pattern Recognition, 2016, pp. 42074215.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, Imagenet classification with deep convolutional neural networks, in International Conference on Neural Information Process- ing Systems, 2012, pp. 10971105.