07-05

16th International Conference on Machine Vision Applications (MVA)
National Olympics Memorial Youth Center, Tokyo, Japan, May 27-31, 2019.

Similar Finger Gesture Recognition using Triplet-loss Networks

Gibran Benitez-Garcia[†] Muhammad Haris[†] Yoshiyuki Tsuda[‡] Norimichi Ukita[†]
[†]Toyota Technological Institute, Japan [‡]DENSO CORPORATION, Japan
{gibran,mharis,ukita}@toyota-ti.ac.jp

Abstract

We propose an efficient gesture recognition method for continuous finger gestures in untrimmed videos. We aim to discriminate similar finger gestures such as flicking. This type of gestures which are conducted only by the orientation and movement of the fingers tends to be similar, making them difficult for a correct classification since a clear temporal boundary of each target gesture is ambiguous. Thus, the recognition should focus on the accuracy to find the temporal boundaries of the target gestures. We proposed a framework based on a triplet-loss network which learns to decrease the distance of true positive boundaries while increasing that of false positive ones. Finally, we adopt a temporal representation of the segmented gesture using a stack of feature maps for gesture classification. Real-time processing and high performance are achieved with relatively compact deep learning models, which are evaluated on a new dataset of vehicle driver finger gestures. Our approach outperforms the results of previous works for online temporal segmentation and gesture classification, and it can run in real-time at 53 fps.

1 Introduction

Hand gesture recognition is an essential part of human-computer interaction (HCI). In particular, touchless automotive user interfaces controlled by hand gestures can provide comfort and safety on driving while manipulating secondary devices like audio and navigation systems. A real-time vision-based system developed for hand gesture recognition has to deal with issues such as discrimination of natural movements of the hand while driving, intraclass variability in the duration of the gesture, recognition of continuous hand gestures, and a low computational cost for working online [1, 2, 3].

Recent proposals for continuous hand gesture recognition use deep convolutional neural networks (CNN) and 3D-CNN with multi-modal data inputs [4, 5, 6, 7]. Most of the state-of-the-art approaches from this task come from the ChaLearn LAP ConGD challenge [3], which focuses on recognizing gestures from continuous RGB-D videos. However, the gestures presented in this challenge are different from those needed for controlling touchless automotive interfaces. Since driver hand gestures have to avoid driving distractions [8], gestures performed while holding the steering wheel are chosen for this task. Thus, driver hand gestures are limited to finger gestures such as flicking, pointing or gesture combinations of fingers.

Flicking gestures are conducted only by the orientation and movement of isolated fingers. Therefore, various directions and continuous gestures may look similar in motion and appearance, making a challenge for recognition systems. For instance, Fig. 1 shows the similarities between isolated (denial) and continuous gestures (two flicking left), by illustrating the segmentation of the three motion states: preparation, nucleus, and retraction [9]. It is clear that the preparation and retraction are similar between different finger gestures, besides nucleus sate from continuous gestures may also share motion and appearance similarities, such as two continuous flicking left and an isolated denial gesture, as shown in the figure. A robust finger gesture recognition approach has to avoid similar states between different classes (preparation and retraction), as well as discriminate between similar gestures conducted continuously. These issues can be tackled by an efficient temporal segmentation method able to perform online. Temporal segmentation consists of defining temporal boundary frames (start and end) of the gesture so that the gesture classification can be focused on target gesture frames only.

In this paper, we propose a two-stage approach to overcome the challenges of continuous finger gesture recognition. Temporal segmentation is based on a triplet-loss network at the learning stage, which is suitable for discriminating the temporal boundaries of each gesture[1]. With this loss function, the network learns to decrease the distance between the correct temporal boundaries while increasing that between false positives boundary frames [10]. For gesture classification, we adopt a temporal representation of the segmented frames using a stack of feature maps to preserve the temporal information. In addition, we provide the spatial hand location based on the SSD (Single Shot MultiBox Detector) object detector [11]. Finally, we ensure real-time performance by taking advantage of a recent compact network architecture aimed at mobile deployment [12]. We evaluated our approach on a new dataset of continuous vehicle driver finger gestures, which consists of eight gestures captured from 20 subjects using an IR camera.

[1]For simplicity, from now on we refer to the nucleus state of gestures just as a "gesture"

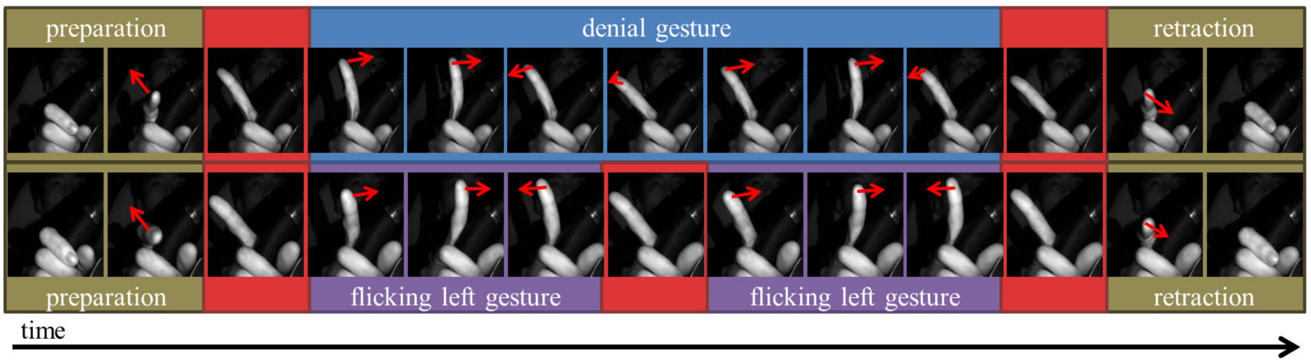


Figure 1. Example of similarities between isolated and continuous gestures. The top row shows a sequence of an isolated gesture (denial). The bottom row shows a sequence of two continuous gestures (flicking left). Red arrows illustrate a finger motion from the previous frame. Red frames represent the boundaries of target gestures. Our proposal is capable of correctly recognizing these three gestures.

The main contribution of this paper is a two-stage continuous finger gesture recognition approach based on a triplet-loss network and feature maps stacking for temporal segmentation and gesture classification, respectively. Empirical evaluation demonstrates that our proposal: 1) outperforms previous works for online real-time gesture recognition such as [13]; 2) achieves real-time performance (up to 53 fps) with small size models (19.2 MB in total).

2 Related Work

Based on the way to approach temporal segmentation, recent works for continuous action and gesture recognition can be divided into four categories: frame-wise [13, 14], sliding windowing [7, 15], binary classification [16, 6], and boundary similarity [17, 18]. Frame-wise-based methods classify gestures frame by frame, some approaches apply a link process of frame scores to temporal limitate the continuous gesture detected [13, 14]. The second category aims to spot target gestures by sliding temporal windows [7, 15], a fixed time window limits its performance when gestures with significant variations on length are detected.

Binary classification approaches aim to discriminate between gesture and no-gesture frames [16, 6]. The main drawback of this category comes when huge similarities between both classes are presented, leading to many false positive recognition errors. On the other hand, similarity-based methods focus on specific similitudes between the starting and ending frames of target gestures. For instance, in [17] the motionless of gesture boundaries is proposed to detect starting and ending frames. Issues of this category appear when there are boundary frames out of the initial similarity assumption.

Similar to binary classification approaches, we propose to detect possible boundaries by classifying between boundary and non-boundary frames. However, we overcome the possible false positive errors by introducing a triplet-loss network which detects start and end frames based on learned similitudes (conversely to similarity-based methods which are usually predefined). Additionally, inspired by [19, 15] we consider the motion by appending frames as extra channels in the input of our network. Thus, features extracted from the convolutional neural network (CNN) encode the motion contained in contiguous previous frames. In this way, we avoid expensive computational cost methods such as optical flow and two-stream CNN architectures.

3 Proposed Method

The input of our gesture recognition method is a sequence of S frames, i.e., $(f_t, f_{t-1}, ..., f_{t-S+1})$, where t represents the current input frame. The output for each frame is a bounding box and the hand gesture label if the current frame is an ending boundary. As illustrated in Fig. 2, we introduce a novel proposal approach for temporal segmentation based on a triplet-loss function at the learning stage (Sec. 3.1). Finally, the gesture is classified using a temporal feature map stacking and two extra depth-wise convolutional layers (Sec. 3.2).

3.1 Triplet-loss based temporal segmentation

We claim that finger gestures can be temporally segmented by detecting boundary frames from their similarity. Before the triplet-loss procedure, we propose a method for boundary detection which additionally regresses the hand location of the driver. Boundary detection is defined as a binary classification, where the network discriminates between boundary and non-boundary frames. We build our proposal on the SSD

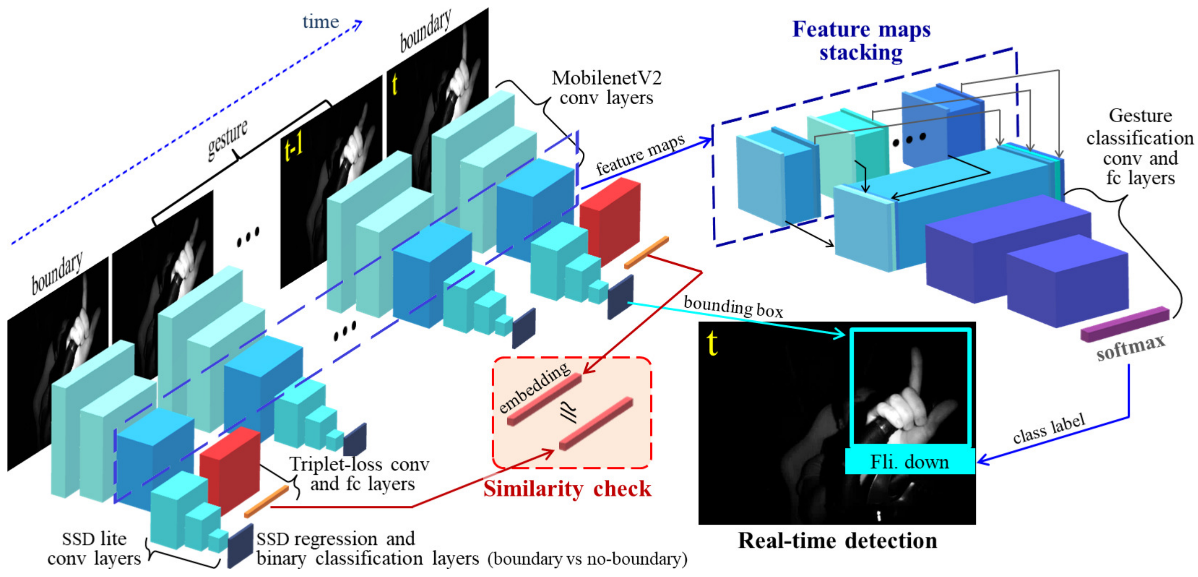


Figure 2. Overview of our hand gesture recognition proposal. From the current frame, we extract convolutional features and perform object detection to get hand location and binary classification results. If a boundary is identified, we propose an extra convolutional layer (red color cuboid) trained with a Triplet-loss function to define an embedding vector. When a second boundary is detected, we determine temporal boundaries by evaluating the similarity between both embeddings. Finally, we propose a temporal stacking of feature maps and extra convolutional layers for gesture classification (lighted blue cuboids).

detector [11], which classifies and regresses the bounding box of the object in a single-stage architecture. In order to ensure real-time performance, a compact and faster version named SSD lite is adopted as presented in [12].

MobilenetV2 [12] is defined as a base network for our proposal. This architecture exploits the depth-wise convolutions and introduce the inverted residual block of convolutions, making MobilenetV2 an efficient compact network capable to perform on mobile devices. Since NIR-frames have only one channel, and a single frame lacks motion information, we define the input of the network by appending previous frames as extra channels. Thus, the input I is defined as

$$I = \{f_t, f_{t-1}, f_{t-2}, ..., f_{t-S+1}\} \tag{1}$$

with size of (M, N, S), where S defines the number of appended frames, and M, N the spatial size of the frame, which is fixed to 300×300 for the SSD approach. We empirically found that $S = 3$ is suitable based on the tradeoff between computational cost and accuracy.

The triplet-loss function [10] aims to minimize the Euclidian distance between the anchor and "difficult positive" samples while penalizing that between the anchor and "difficult negatives". A difficult positive sample belongs to the same class as the anchor but has a large Euclidian distance. Conversely, "difficult negatives" present a short distance belonging to different classes. This phenomenon often occurs when detecting boundary frames of finger gestures. For instance, a difficult negative appears in the isolated gesture of the example shown in Fig. 1. The distance between boundary frames (highlighted in red) and the fourth frame of the denial nucleus is very short; although they belong to different classes (boundary vs. no-boundary). Therefore, we propose to build a Euclidean space where distances directly correspond to a measure of boundary similarity. The temporal segmentation is reduced to a similarity check between two embeddings of possible boundary frames, defining a correct detection if $D(\varepsilon_s, \varepsilon_e) > Th$, where Th is the similarity threshold, and $D(\varepsilon_s, \varepsilon_e)$ is the Euclidean distance of embeddings from possible starting (ε_s) and ending (ε_e) boundaries.

The training stage of a triplet-network relies entirely on the quality of triplets used to calculate the loss, being triplet mining crucial for the process. Online triplet mining obtains reliable triplets for each batch of samples, the biggest L2 distance between anchor-positive (x_i^a, x_i^p) and the smallest between anchor-negative (x_i^a, x_i^n), respectively [10]. The triplet-loss is defined by:

$$L = \sum_{i}^{B} \left[||f(x_i^a) - f(x_i^p)||_2^2 - ||f(x_i^a) - f(x_i^n)||_2^2 - \alpha \right] \tag{2}$$

where α is a margin that is enforced between positive and negative pairs, $f(x)$ is the embedding of feature x in to a Q-dimensional Euclidean space, and B is the number of samples per class in the batch. In our proposal, embeddings are defined as $\varepsilon = f(\psi(x))$, where $\psi(x)$ represents a depth-wise convolution with batch normalization, ReLu6 non-linearity and a fully-connected layer, x is the features maps obtained from MobileNetV2, and $Q = 60$.

Finally, the proposed temporal segmentation process (illustrated in Fig. 2) is defined in four steps: 1) find a boundary clip with at least two consecutive boundary frames, and store the embedding of the highest scored frame; 2) keep the feature maps of the no-boundary frames detected after the boundary clip; 3) when a second boundary clip is identified, define the temporal segmentation with $D(\varepsilon_s, \varepsilon_e) > Th$; 4) clear stored memory if no second boundary clip detected after T time; leave the embedding of the ending boundary otherwise (for continuous gestures).

3.2 Feature map stacking for gesture recognition

Recent works focused on gesture recognition tend to use 3DCNN or multi-stream CNN which results in huge networks difficult to train and implement for real-world applications [7, 16, 6, 17, 18]. In contrast to those methods, we focus on rendering the network compact and maintain acceptable accuracy. Therefore, we add a few depth-wise extra layers and reuse the feature maps calculated from the base network.

Inspired by [7], we adopt a temporal stacking method for the feature maps. Let M_i^ϕ the i-th feature map, $i \in \{1, 2, ..., C\}$, from the ϕ-th frame, $\phi \in \{1, 2, ..., t\}$, where C is the number of channels at the last convolutional layer of the base net, and t is the current frame (the ending boundary). Thus, the temporal stacking is defined as

$$M' = [M_1^1, M_1^2, ..., M_C^{t-1}, M_C^t] \tag{3}$$

In this way, we keep the temporal structure of the features before the next step which is temporal pooling. It is worth noting that we apply the same base network frame-wise. Thus, feature maps may contain similar information in the same temporal order since weights are shared[2]. Temporal pooling is applied so that the output size is equal to (M, N, CK), where $M = N$ is the spatial size of the feature maps, and K is a tunable parameter which defines the ideal representation length of all gestures. The M' pooled features are feed to two depth-wise convolutional layers with batch normalization, ReLu6 non-linearity, and dropout. The last fully-connected layer uses softmax to determine the class of the temporally segmented gesture. The gesture classification process is illustrated in the right part of Fig. 2.

[2]We are aware that this temporal structure may be lost after convolutional and fully connected layers. However, this is part of the limitations of using CNN models by considering real-time performance.

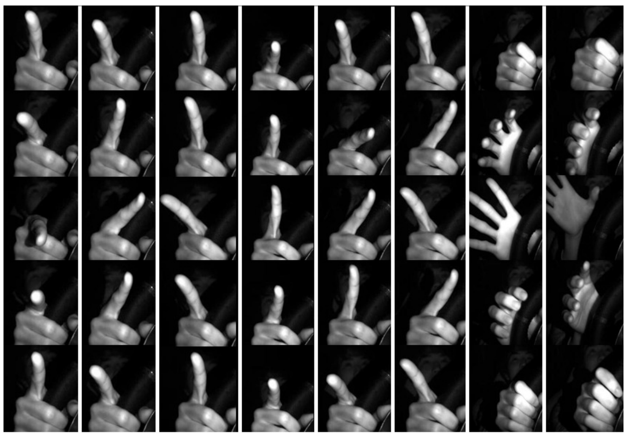


Figure 3. Example of the eight classes of our dataset: (a) flicking down, (b) flicking left, (c) flicking right, (d) flicking up, (e) circle, (f) denial, (g) open, and (h) release.

4 Experimental Results

4.1 Dataset

We evaluate our approach with a new dataset which consists of more than 2,800 instances of continuous finger gestures performed by a driver holding the steering wheel. Eight classes are included: flicking down, left, right, up, denial, circle, open and release. A total of 20 subjects recorded gestures in three different positions of the steering wheel. Videos are recorded at 30 fps with a NIR camera. Individual frames are normalized to 640×480 pixels, 8-bit depth. Figure 3 illustrates all the gestures included in the dataset. The whole data is split into training, validation and test sets, which contains 1,536, 432 and 928 instances respectively. Frame-wise annotations of motion states are provided for the complete dataset, whereas, hand bounding boxes just for the training set. Therefore we cannot evaluate the performance of hand location properly.

4.2 Implementation details

We implement all models using PyTorch, with training executed on a single Nvidia GeForce GTX 1080. We resize the input frames to 300 × 300 as the original SSD proposes. All models were trained with the training set, and the best model is selected based on its accuracy on the validation set, then reported results are based on unseen data from the testing set. In total, we have three different models, SSD lite, triplet-network, and gesture classification. Each of them was optimized with different parameters. We trained the base network with SSD lite on the top. Thus, we freeze all layers of MobileNetV2 for the other two models.

We train the SSD lite model using mini-batch SGD with Nesterov momentum of 0.9, and mini-batches of

Table 1. Results comparison between our proposal variations and ROAD.

Method	tIoU@0.3	tIoU@0.5	tIoU@0.8
Binary	0.73	0.68	0.59
Similarity	0.76	0.71	0.61
Triplet	0.82	0.77	0.67
ROAD	0.74	0.70	0.57

size 64. The initial learning rate was 0.01 and decayed by 0.98 every epoch. The MobileNetV2 architecture was reduced 25% from its original proposal with an input of $300\times300\times3$ so that the last convolutional layer outputs feature maps of size $10 \times 10 \times 240$. Since the dataset includes two classes different than one finger gestures (open and release) we train with three classes: rest (holding the steering wheel), boundary, and no-boundary frames.

The triplet network was trained using the Adam optimizer with a learning rate of 0.0001 with exponential decay after 150 epochs, $B = 32$, and $\alpha = 0.2$. The extra depth-wise convolutional layer uses 120 filters of size 1×1, and the fully connected layer includes 60 hidden neurons. We experimentally set Th to 0.55. We train the gesture classification model similar to SSD lite, with SGD and the same parameters except for the learning rate 0.045, and the size batch of 32. We vary the value of K between the length of the shortest and the longest gesture (in terms of the number of frames), we get the best results with $K = 8$, so that the stacked feature maps are pooled to $10 \times 10 \times 1920$. Both depth-wise convolutional layers include 960 and 480 1×1 filters, respectively.

4.3 Temporal segmentation and gesture recognition

The performance of our gesture spotting approach is evaluated based on the class-aware temporal Intersection-over-Union score (tIoU) at different detection thresholds. The score counts when the gesture label is correctly predicted and the tIoU between predicted and ground-truth boundaries is higher than the detection threshold.

We compare our results with the proposal of [13]: Real-time Online Action Detection (ROAD). This is one of the few state-of-the-art approaches that can perform online without comprising the memory storage with huge models (such as 3DCNN-based approaches). We trained ROAD with a reduced VGG-16 network and the conventional SSD, by using their publicly available code on Pytorch. Table 1 shows performance of variations of our approach compared with ROAD. "Triplet" is our approach as described in Fig. 2; "Similarity" variation excludes the triplet-loss network, thus the similarity check is based on a hot-vector extracted from the last 1×1 convolutional layer of SSD lite; "Binary" temporally segment the gesture excluding the similarity check based on the binary classification of SSD lite (boundary vs. no-boundary frames).

As can be seen from Table 1, the results of our triplet-loss-based approach outperform the other two variations, showing that by measuring the similarities between boundary frames the temporal segmentation is improved. We also overcome the results of ROAD, which problems rely on its frame-wise recognition. Figure 4 shows the average tIoU@0.5 per each gesture. Flicking up gesture presents the lowest results due to the appearance of gesture and boundary frames are very similar. On the other hand, the recognition performance of correct temporal detections is presented in the confusion matrix shown in Fig. 5. The misrecognition error of flicking right with flicking up (26%) is due to the apparent similarity of these gestures generated by the perspective when the hand is at the farthest point from the camera location.

Table 2 presents the test time speed of the four approaches altogether with the model size in terms of storage memory consumption. We can notice that the

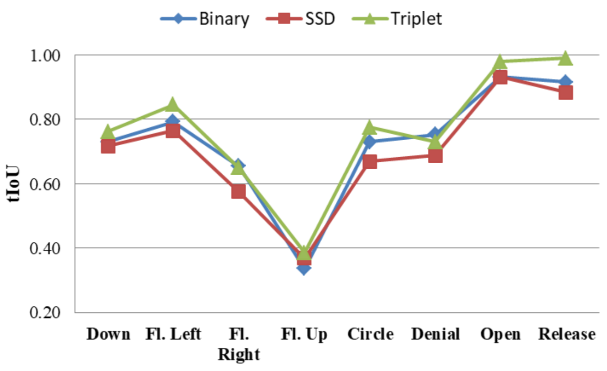
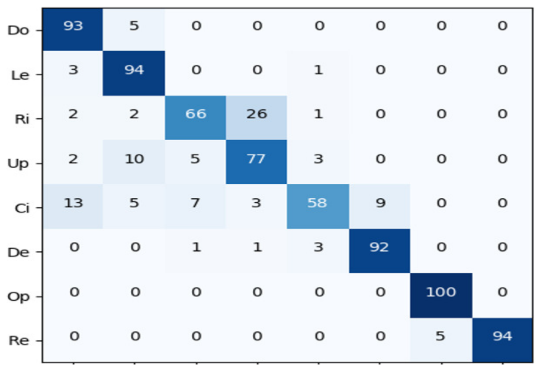


Figure 4. Average tIoU per class using a detector threshold of 0.5. Results from each variation of our approach (Binary, SSDlite, and Triplet-loss).



Do	93	5	0	0	0	0	0	0
Le	3	94	0	0	1	0	0	0
Ri	2	2	66	26	1	0	0	0
Up	2	10	5	77	3	0	0	0
Ci	13	5	7	3	58	9	0	0
De	0	0	1	1	3	92	0	0
Op	0	0	0	0	0	0	100	0
Re	0	0	0	0	0	0	5	94

Figure 5. Confusion matrix of Triplet-loss.

Table 2. Test time speed and model size comparison.

Method	Speed	Model size
Binary	15 ms (67 fps)	15 MB
Similarity	17 ms (59 fps)	15 MB
Triplet	19 ms (53 fps)	19.2 MB
ROAD	25 ms (40 fps)	95.1 MB

fastest method is the one based on SSD lite for temporal segmentation; however, it achieves the lowest tIoU score. On the other hand, our triplet-loss approach can perform on real-time with an acceptable model size.

5 Conclusion

In this paper, we proposed a continuous finger gesture recognition approach. Temporal segmentation is based on the triplet-loss function at the training stage by emphasizing the appearance and motion similarities of the boundary frames. Our gesture classification approach leverages the detected target gesture frames by modeling the motion sequence with stacked feature maps extracted frame-wise. We can conclude that our proposal is suitable for similar finger gesture recognition and it can achieve real-time performance with compact convolutional neural networks.

References

[1] M. Asadi-Aghbolaghi, A. Clapés, M. Bellantonio, H. J. Escalante, V. Ponce-López, X. Baró, I. Guyon, S. Kasaei, and S. Escalera, "Deep learning for action and gesture recognition in image sequences: A survey," in *Gesture Recognition*. Springer, 2017, pp. 539–578. 1

[2] P. Wang, W. Li, P. Ogunbona, J. Wan, and S. Escalera, "Rgb-d-based human motion recognition with deep learning: A survey," *Computer Vision and Image Understanding*, 2018. 1

[3] J. Wan, S. Escalera, G. Anbarjafari, H. J. Escalante, X. Baró, I. Guyon, M. Madadi, J. Allik, J. Gorbova, C. Lin *et al.*, "Results and analysis of chalearn lap multi-modal isolated and continuous gesture recognition, and real versus fake expressed emotions challenges," in *IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2017, pp. 3189–3197. 1

[4] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. C. Niebles, "Sst: Single-stream temporal action proposals," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6373–6382. 1

[5] S. Buch, V. Escorcia, B. Ghanem, L. Fei-Fei, and J. C. Niebles, "End-to-end, single-stream temporal action detection in untrimmed videos," in *British Machine Vision Conference (BMVC)*, 2017. 1

[6] G. Zhu, L. Zhang, P. Shen, J. Song, S. Shah, and M. Bennamoun, "Continuous gesture segmentation and recognition using 3dcnn and convolutional lstm," *IEEE Transactions on Multimedia*, 2018. 1, 2, 4

[7] M. Zolfaghari, K. Singh, and T. Brox, "ECO: efficient convolutional network for online video understanding," in *15th European Conference on Computer Vision (ECCV)*, 2018, pp. 713–730. 1, 2, 4

[8] NHTSA, "Investigation and prosecution of distracted driving cases," report No. DOT HS 812 407, National Traffic Law Center, 2017. 1

[9] P. K. Pisharady and M. Saerbeck, "Recent methods and databases in vision-based hand gesture recognition: A review," *Computer Vision and Image Understanding*, vol. 141, pp. 152–165, 2015. 1

[10] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823. 1, 3

[11] 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 (ECCV)*, 2016, pp. 21–37. 1, 3

[12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520. 1, 3

[13] G. Singh, S. Saha, M. Sapienza, P. Torr, and F. Cuzzolin, "Online real-time multiple spatiotemporal action localisation and prediction," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3657–3666. 2, 5

[14] V. Kalogeiton, P. Weinzaepfel, V. Ferrari, and C. Schmid, "Action tubelet detector for spatio-temporal action localization," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 4415–4423. 2

[15] O. Kopuklu, N. Kose, and G. Rigoll, "Motion fused frames: Data level fusion strategy for hand gesture recognition," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 2103–2111. 2

[16] H. Wang, P. Wang, Z. Song, and W. Li, "Large-scale multimodal gesture segmentation and recognition based on convolutional neural networks," in *IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2017, pp. 3138–3146. 2, 4

[17] P. Wang, W. Li, Z. Gao, C. Tang, and P. O. Ogunbona, "Depth pooling based large-scale 3-d action recognition with convolutional neural networks," *IEEE Transactions on Multimedia*, vol. 20, no. 5, pp. 1051–1061, 2018. 2, 4

[18] Z. Liu, X. Chai, Z. Liu, and X. Chen, "Continuous gesture recognition with hand-oriented spatiotemporal feature," in *IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2017, pp. 3056–3064. 2, 4

[19] Z. Hu, Y. Hu, J. Liu, B. Wu, D. Han, and T. Kurfess, "3d separable convolutional neural network for dynamic hand gesture recognition," *Neurocomputing*, vol. 318, pp. 151–161, 2018. 2