Deep Attention Network for Joint Hand Gesture Localization and Recognition Using Static RGB-D Images

Yuan Li, Xinggang Wang, Wenyu Liu, Bin Feng[*]

School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China.

Abstract

This paper presents an effective deep attention network for joint hand gesture localization and recognition using static RGB-D images. Our method trains a CNN framework based on a soft attention mechanism in an end-to-end manner, which is capable of automatically localizing hands and classifying gestures using a single network rather than relying on the conventional means of stage-wise hand segmentation/detection and classification. More precisely, our attention network first computes the weight for each proposal generated from the entire image, in order to judge the probability of the hand appearing in a given region. It then implements a global-sum operation for all proposals, which is influenced by their corresponding weights, in order to obtain a representation of the entire image. We demonstrate the feasibility and effectiveness of our method through extensive experiments on the NTU Hand Digits (NTU-HD) benchmark and the challenging HUST American Sign Language (HUST-ASL) dataset. Moreover, the proposed attention network is simple to train, without requiring bounding-box or segmentation mask annotations, which makes it easy to apply in hand gesture recognition systems. Based on the proposed attention network and taken RGB-D images as input, we obtain the state-of-the-art hand gesture recognition performance on the challenging HUST-ASL dataset.

[*]Corresponding author

Email addresses: yuancoder222@gmail.com (Yuan Li), xgwang@hust.edu.cn (Xinggang Wang), liuwy@hust.edu.cn (Wenyu Liu), fengbin@hust.edu.cn (Bin Feng)

Preprint submitted to Journal of Information Sciences *July 19, 2020*

Keywords: Soft attention mechanism, CNN, hand gesture localization and recognition, RGB-D images.

1. Introduction

Lately, hand gesture recognition has become increasingly important, because of the increasingly frequent interaction between human beings and machines in applications such as remote-control games, virtual reality, and sign language recognition. Even though many studies [25, 8, 41, 14, 21] have contributed to this field, hand gesture recognition still has a long way to go for successful real-environment applications.

In general, the problem can be classified into two branches, namely static and dynamic situations. Dynamic hand gesture recognition attempts to explore spatial-temporal characteristics, while static recognition devotes its attention to the internal information of a single image. This paper focuses on static hand-gesture recognition.

The study of static hand-gesture recognition is meaningful, because different hand shapes convey specific information with no motion cues. In addition, it can help reduce redundant frames in dynamic problems.

The components of static hand gesture recognition always consist of three stages, as shown in Fig. 1(a). These are image acquisition, hand localization, and gesture classification.

First, concerning image acquisition, sensors such as the Microsoft Kinect, ASUS Xtion, and Intel RealSense allow us to collect gesture data easily and conveniently. It must be stressed that in most cases, we can obtain not only color images, but also pixel-wise depth cues.

The second part is called hand localization, and this is indispensable. Most previous approaches [25, 8, 20] take advantage of depth information to solve this problem. They assume that the hand of the user is always in front of the whole body and background, and then set a threshold to segment the hand.

Finally, regarding the task of classification, conventional methods need to

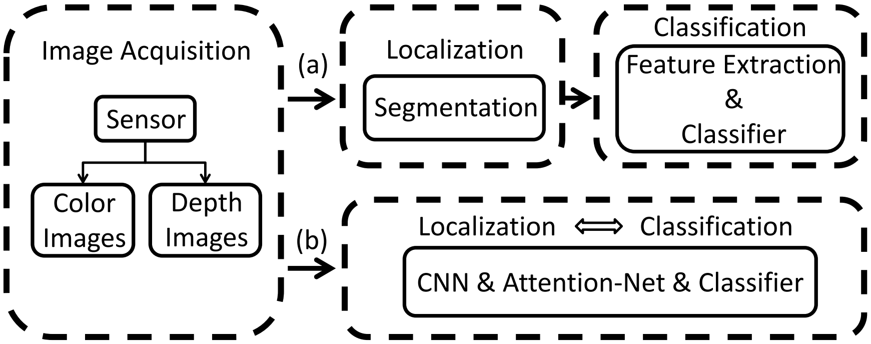


Image Acquisition
Sensor
Color Images
Depth Images
(a)
Localization
Segmentation
Classification
Feature Extraction & Classifier
(b)
Localization ⟺ Classification
CNN & Attention-Net & Classifier

Figure 1: Comparison between the general framework of static hand gesture recognition and ours. The left shared dashed rectangular block shows the process of image acquisition. After this, there are two branches: (a) General framework, in which localization and classification are two independent modules; (b) our framework, in which localization and classification are in a unified network using a soft attention mechanism.

extract features, using the histogram of oriented gradients (HOG) [6], histogram of 3D facets (H3DF) [49], and so on. Once the hand features are extracted, they will be input to a classifier, such as a support vector machine (SVM) or random forest. Recently, Ren *et al.* [45] proposed a direction normalization method based on the multi-scale weighted histogram of contour direction (MSWHCD), which counts the direction of the contour point to focus on the most significant hand features in the first-person view of wearable devices. Feng *et al.* [8] proposed a novel hand-crafted descriptor by extracting bag of contour fragments (BCF) features from depth projection maps (DPM), referred to as BCF-DPM, which captures shape and structure information, and can deal with occlusion, missing parts, and deformation.

However, this traditional framework of static hand-gesture recognition has its limitations: 1) It assumes that the hand has the lowest values in the depth map, which is not robust to noise. Moreover, it is not user-friendly, as users are constrained to place their hand at the nearest place to the sensor. 2) It uses either a color image or depth image, and mostly fails to deeply learn features for gesture recognition. 3) It treats hand localization and gesture recognition as two separate steps. In this case, if hand localization fails, then it is impossible to correctly classify gestures, and the localization feature is not optimized for gesture classification.

Recently, the field of computer vision has been tremendously influenced by the rapid development of deep learning. Convolutional neural networks (CNNs) [16, 42, 48] have demonstrated their formidable power in extracting the discriminative features of images.

Inspired by the potential of CNNs, we propose a novel deep-CNN framework to jointly localize and recognize hand gestures using RGB-D images in an end-to-end manner, as shown in Fig. 1(b). Specifically, our network first computes the feature map of the four-channel RGB-D image, and then it makes use of the region-of-interest (RoI) pooling layer [9], as well as several fully connected (FC) layers, in order to obtain the features of each proposal. Subsequently, these features will be fed into an attention network, in which their weights will be softly computed and assigned to the corresponding proposals. By introducing this kind of attention mechanism, we can gradually assign larger weights to the most pertinent regions. In particular, in the task of hand gesture recognition, it is obvious that the most pertinent regions are those surrounding the hand. Next, the feature of the whole image is computed by implementing a weighted global-sum operation over the vectors of all proposals. The weights are automatically learned in the proposed attention network. It must be clarified that our attention network follows the principle of soft attention mechanisms [5], which means that all subsets of the input are processed, instead of selecting a few subsets to attend. Finally, once the final feature vector of the image is obtained, we can transform this into a vector of length n, where n denotes the number of classes, and then proceed to perform the classification.

RGB is a method of encoding color, whereas depth images contain abundant geometric information. Although RGB images and depth images present information from different angles, our holistic network is still capable of locating approximately the same hand position in both, because the attention mechanism is designed to find the place that contributes most to the specific task.

Moreover, the CNN itself has strong capabilities in feature extraction and representation, which can help the above two types of information to work together complementarily.

In summary, this paper provides the following contributions:

- We propose a deep end-to-end CNN framework for static hand gesture recognition based on a soft attention mechanism, which is capable of automatically localizing the hand and classifying the gesture with an excellent performance.
- Thanks to the soft attention mechanism, we perform gesture localization in a weakly supervised manner, which does not require bounding-box or segmentation annotations in training images. Thus, the proposed method is easy to deploy in hand gesture-recognition systems.
- Our network can robustly handle color images and depth cues in the field of static hand gesture recognition. More importantly, with the fusion of these two kinds of information, namely RGB-D images, the performance will be further improved.

The remainder of this paper is organized as follows. Section 2 reviews related work from recent years. Section 3 introduces our method in detail. Section 4 presents the experimental results and some analyses. The paper closes with conclusions and a discussion in Section 5.

2. Related Work

Most previous work dealt with gesture localization and classification separately. Concerning hand localization, also known as hand detection, [25, 8, 20] simply segmented the hand out of the other parts by configuring a threshold, which was estimated from particular circumstances using depth cues. The authors of [18, 36] utilized skin color maps without using depth information, and [22, 33] achieved better segmentation results by combining color-based skin detection and depth thresholding. Regarding classification, many conventional approaches relied on hand-crafted features [6, 25, 49, 8, 38, 1, 2], which may capture silhouette, shape, and structure information.

Recently, CNNs have demonstrated excellent performance in many vision tasks, such as object detection [10, 9], image classification [31, 28], and semantic segmentation [19, 32]. It is pleasing that many challenging human-related vision tasks benefit significantly from this technique. For example, for human pose recovery[34, 11, 12, 47], Hong *et al.* [12] used non-linear mapping with a multi-layered deep neural network based on feature extraction, with multi-modal fusion and back-propagation optimization. Yu *et al.* [47] proposed a novel pose recovery framework by simultaneously learning the tasks of joint localization and detection. For hand gesture recognition, Koller *et al.* [14] exploited the training of a CNN with an expectation maximization (EM) algorithm on 1 million hand images, and achieved a state-of-the-art performance on two large public sign language datasets. Molchanov *et al.* [21] presented a recurrent 3D CNN for the online detection and classification of hand gestures. However, these studies focus on the dynamic hand gesture recognition task, and use cropped data after tracking hands, i.e., separating the procedures of localization and classification. There has been relatively less work regarding static hand gesture recognition, while Yamashita *et al.* [41] seem to be the first to exploit CNNs and treat localization and classification together. They proposed a deep CNN with a bottom-up structure, incorporating a special layer for binary image extraction that can segment the hand. We believe it is advantageous to address localization and classification jointly, as they are highly interdependent. Therefore, we attempt to integrate them into a single network in a different manner, and train it end-to-end.

Recently, the attention mechanism has been grafted onto the deep learning framework with considerable success. Many typical computer vision tasks, such as image classification [39], object detection [46], and semantic segmentation [3] have made tremendous advances under the influence of attention. Furthermore, [40, 44] utilized attention models for image and video captioning. Yang *et al.* [43] adapted stacked attention networks (SANs) to solve image question answering problems. Lee *et al.* [17] presented recursive recurrent neural networks with attention modeling for lexicon-free optical character recognition (OCR) in the wild. Kuen *et al.* [15] also made use of recurrent attentional networks for saliency detection. In addition, in the scope of machine translation, speech recognition, and natural language processing, researchers have extensively explored the value of attention models. Hand recognition is difficult, due to the variation in hands and complicated backgrounds, and there is no previous work employing an attention mechanism for this task. Here, we propose an attention network that can softly weight the proposal features in order to judge which region is most likely to enclose the hand.

Another factor that contributes to our success is the use of RGB-D images. Benefitting from the widespread of commodity depth cameras, more and more RGB-D data is available. RGB-D images carry abundant information, with not only color data but also depth cues, which interests many researchers [27, 4]. For example, [37] encodes depth using three channels (HHA), which makes it possible to treat depth data like normal RGB images, which can be directly fed into a pre-trained CNN model. Concerning hand-related work, most approaches focus on hand pose estimation [30, 26] using RGB-D images. Unlike these methods, we simply treat depth cues as a normal channel of input to the CNN.

It should be emphasized that our method is related to fast region-based convolutional neural networking (Fast R-CNN) [9], which shares the time-consuming convolutional computation of different proposals by an RoI pooling layer, and is applied to object detection. However, our method is quite different from Fast R-CNN: 1) The objective of Fast R-CNN is to detect objects in images, whereas we are concerned with gesture recognition. 2) The inputs of Fast R-CNN are color images, whereas our inputs are RGB-D images. 3) The training groundtruths of Fast R-CNN include bounding boxes for objects, whereas we only take gesture categories as supervisions.

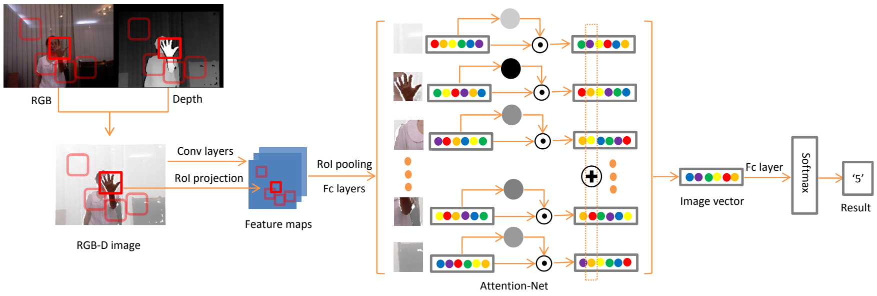


Figure 2: Architecture of our network. The red rectangles in the image represent the proposals generated by the sliding window. Here, we only choose five proposals/windows as representatives. We deepen the color of the window that encloses the hand to convey that it is the most anticipated one, while it is not in fact special. In the attention network, the shades of filled circles represent the values of weights. The darker the circle, the more attention the network will pay. A circle with a point in it indicates an element-wise operation, while one with a plus sign indicates the global-sum operation.

3. Method

In this section, we will first concisely describe the whole framework, and then introduce each step in detail. The overall architecture of our network is shown in Fig. 2. First, the entire input image goes through a pre-trained CNN model to extract image-level features, and then we utilize an RoI pooling layer and several fully-connected (FC) layers to obtain the representation of each proposal generated by a sliding window. Second, we construct the attention network to acquire the weight of each proposal, in order to decide which part we will focus on. Furthermore, we need to aggregate all the feature vectors of the proposals using a global-sum operation. Third, we can obtain the image-level features after applying our attention network, and just need to classify these.

It is worth noting that when training the network, we only give image-level annotations, with no region-related labels, which contributes to the weakly supervised learning method for hand localization.

In order to introduce our approach in a digestible manner, we divide the whole framework into four modules, which are an input layer with aligned RGB-D data, feature extraction via the RoI pooling layer, the attention network for gesture localization, and gesture classification using the softmax loss.

3.1. Input layer with aligned RGB-D data

The input of our network consists of two components. The first is RGB-D images, which constitute the fusion of color images and depth cues. Here, we treat the depth cues as a normal channel, similar to other color channels, and align it to the primary RGB images, which means we need to restrict the depth value to a range of [0,255]. Specifically, we formulate the conversion function as Eq. (1)

$$D_\sigma(i,j) = \begin{cases} \left\lfloor \frac{max(D)-D(i,j)}{max(D)-min(D)} \times 255 \right\rfloor & if \quad D(i,j) \neq 0 \\ 0 & if \quad D(i,j) = 0, \end{cases} \tag{1}$$

where D denotes the set of the depth values in a single image, and (i,j) represents the index of each pixel. Furthermore, $max(\bullet)$ and $min(\bullet)$ denote the functions obtaining the maximum and minimum values in the depth map, respectively. Consequently, D_σ is the set of depth values after conversion. In addition, there exist some bad pixels (value 0) resulting from the depth camera failing to collect information, and these can be deemed as the farthest points. The second component consists of proposals, which are a set of generative bounding boxes. There exist many effective proposal generating approaches, such as sliding window, selective search [35], and edge boxes [50]. In our approach, we should generate multiple proposals in order to make sure that the hand is included, so that the subsequent work makes sense. We choose the sliding window here, for its simplicity and stability. The sizes and intervals of the windows can easily be controlled, and as a result the hand can be chosen precisely. We also utilize multiple sizes of windows to cope with various scales of hands.

Parameter setting information will be detailed in Section 4.

3.2. Feature extraction via RoI pooling layer

Fine-tuning is a universal trick in training a deep network. We train the target dataset using of a pre-trained CNN model, which has been trained on a large-scale dataset such as ImageNet [7].

In order to extract representative features using the CNN, we fine-tune our network on a widely used deep CNN model named VGG16 [29], which has 13 convolutional layers, five max pooling layers, and three FC layers. The network processes the entire image using these convolutional and max pooling layers to produce a feature map that can represent the image abstractly.

However, some alterations are necessary because of changes in the input data. First, we need to increase the channel of the first convolutional layer from three to four, to fit the RGB-D condition. Second, we replace the last max pooling layer by an RoI pooling layer, inspired by Fast R-CNN [9], because a set of proposals will also be fed into our network.

The RoI is a rectangular window within an entire feature map, defined by a four-tuple (x, y, h, w), where (x, y) represents the top-left corner and (h, w) represents the height and width. Each RoI window will be divided into an $H \times W$ grid, with sub-windows of approximate size $h/H \times w/W$. After this, we can obtain the value of each grid cell by taking the maximum value in the corresponding sub-window. It should be noted that the RoI pooling operation described above is applied independently to each feature map channel. In essence, the RoI pooling layer is a variant of the max pooling layer, and is simplified from the spatial pyramid pooling layer used in SPPnets [10] with multiple pyramid levels that can output to a fixed-size feature vector.

More details of the employed CNN architectures will be given in Section 4.

3.3. Attention network for gesture localization

The core part of the framework is our attention network. After implementing the input layer and feature extraction introduced above, we obtain the representation of each proposal. These features will be fed into our attention network, in which the weights that decide which part receives special attention will be softly computed. This process can be deemed to judge the probability that a hand lies within each region. Next, the network will proceed to successive element-wise and global-sum operations. We will present a more detailed mathematical expression in the following paragraph.

As mentioned earlier, the input of our attention network is a proposal feature matrix $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_N]^T \in \mathbb{R}^{N \times M}$, where N denotes the number of proposals and M is the output number of fully connected neurons. Next comes the core part, where we compute a weight vector $\mathbf{W} = [w_1, w_2, ..., w_N]^T \in \mathbb{R}^{N \times 1}$ for the densely-collected proposals. This computational procedure can be deemed as regressing a value w_i using a vector $\mathbf{v}_i$ realized by a fully connected layer. After determining the weight vector $\mathbf{W}$, we need to further normalize it using the softmax function:

$$w_i = \frac{e^{w_i}}{\sum_{k=1}^{N} e^{w_k}}, \tag{2}$$

where $e^{(\bullet)}$ is the natural exponential function. This process intuitively reflects the importance of the features of each proposal, and can be further seen as a probability computation for all proposals to estimate the presence of the hand in a corresponding region.

The Hadamard product would then involve acting on the matrix $\mathbf{V}$ with the weight $\mathbf{W}$, *i.e.*,

$$\mathbf{U} = \mathbf{V} \odot \mathbf{W} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_N]^T \in \mathbb{R}^{N \times M}, \tag{3}$$

The vector u_i in the matrix $\mathbf{U}$ is the new feature vector of the i-th proposal.

Considering that we only have image-level annotations, we should obtain the feature vector of the whole image by aggregating these vectors. Thus, we globally sum the matrix $\mathbf{U}$ to be an M-dimensional vector $\mathbf{F} = [f_1, f_2, ..., f_M]^T \in \mathbb{R}^{M \times 1}$, which can be deemed as the final feature vector of the whole image, where

$$f_i = \sum_{j=1}^{N} u_{j,i}. \tag{4}$$

We can imagine that if the network has a pair of eyes, then the vector $\mathbf{F}$ is the feedback from viewing the entire image, and the region with the highest weight will receive the most attention. The network can iteratively adapt the weight to decide which proposal will be paid special attention.

3.4. Gesture classification using softmax loss

After implementing our attention network, we need to perform a normal classification task. The image feature vector is transformed to a length C, where C is the number of hand gesture classes that need to be classified. Narrowly, let the classifier be $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_C] \in \mathbb{R}^{M \times C}$. Then, we can calculate the predicted score vector as follows:

$$\mathbf{S} = [s_1, s_2, ..., s_C]^T = \mathbf{X}^T \mathbf{F} \in \mathbb{R}^{C \times 1}, \tag{5}$$

$$o = \arg\max_j \frac{e^{s_j}}{\sum_{k=1}^{C} e^{s_k}}, \tag{6}$$

$$L(S, Y) = -\log \frac{e^{s_Y}}{\sum_{k=1}^{C} e^{s_k}} = \log \sum_{k=1}^{C} e^{s_k} - s_Y. \tag{7}$$

We can then obtain the predicted result based on the softmax function, as in Eq. (6). When training the network, we formulate the softmax loss function as $L(S, Y)$ in Eq. (7), where Y is the input label of each image.

3.5. Computational complexity analysis

The measurement of floating-point operations (FLOPs) can reflect the computational complexity of a network. For example, the Alexnet has 725M FLOPs and the VGG16 has 15484M FLOPs, which means that VGG16 is much more complex than Alexnet. More than 90% of the FLOPs are caused by their convolutional layers. However, the dominating computational cost of our holistic network comes from our proposed attention network, because we need to regress the weight of each proposal simultaneously. This implies that the computational complexity of our network is proportional to the number of proposals. Consequently, the eventual outcome is about 900B FLOPs if we input 2000 proposals.

4. Experiments

In this section, we will concisely introduce the two datasets, and then present the details of implementing the experiments. Then, the localization and classification results will be presented, demonstrating the effectiveness of our method. Moreover, further analyses of our network will be provided. In addition, codes and trained models for reproducing the results will be made available upon acceptance.

4.1. Datasets

In order to demonstrate that the proposed network can jointly localize and classify hand gestures in an end-to-end deep network without explicit hand region segmentation, the chosen dataset ought to meet the following two requirements: 1) It is a static dataset, which means that each gesture is represented by a single image, not based on video. 2) The input image has a complicated background, and the hand only occupies a small portion of the image. If the attention network cannot lead the network to pay special attention to the window enclosing the hand, then the classification result will be greatly impacted by background noises.

After careful selection, our method will be evaluated on a benchmark dataset called the NTU Hand Digits (NTU-HD) dataset [25], and a challenging dataset called the HUST American Sign Language (HUST-ASL) dataset, which is derived from [8]. More details regarding the two datasets are given below.

4.1.1. NTU Hand Digits dataset

The NTU-HD dataset is a small dataset, which contains a total of 1000 images. There are 10 subjects, each of which performs 10 different gestures and repeats each of these 10 times. Color and depth images can both be obtained.

4.1.2. HUST American Sign Language dataset

The HUST-ASL dataset was generated by Media and Communication Lab in Huazhong University of Science and Technology, and was first used in [8] in

2016. It was collected using Microsoft Kinect with 5440 color images and their corresponding 5440 depth maps. Ten participants are involved, and 34 hand gestures are performed to imitate the digits 0 to 9 and the 24 English letters other than j and z. These are imitated following the samples from the ASL Finger Spelling dataset [23]. Each gesture is repeated 16 times with different degrees of deflection in different orientations. In particular, the performers need to revolve their hands around their wrists or elbows within a certain degree. It is difficult to localize the hand, because the hand only occupies less than about two percent of pixels in a single image.

4.2. Implementation details

In order to exhibit the experimental configurations more clearly, we first present them in Table 1, and will explain each part in detail.

4.2.1. Data preparation

As mentioned in Section 3, we need to fuse the color and depth images to create RGB-D images, and then generate proposals using sliding windows. We set three sizes of windows, for the sake of having different sizes and distances to performers in front of the sensor when collecting data. These are 64×64, 96×96, and 128×128. Moreover, the interval size is set to 32, ensuring that it encloses all the hands in the datasets.

Our datasets are not very large, so we take some effective strategies to avoid over-fitting to some extent.

One is a known method called data augmentation. Specifically, image pyramids [10] are taken with five scales (480, 576, 688, 864, and 1200). During training, we randomly sample a pyramid scale each time an image is sampled. At the test time, pyramids of each scale will be evaluated, and the scores will be averaged to obtain the final result.

Another strategy employed is leave-one-subject-out, which is a means of cross validation. If a dataset is performed by N subjects, then $N-1$ subjects are

Table 1: Experimental configurations.

Data Preparation	
Image Pyramids setting (short side)	480,576,688,864,1200
Sliding window size	64*64,96*96,128*128
Sliding window interval	32
SGD Hyper-parameter Setting	
Weight initialization	Gaussian(0, 0.01)
Bias initialization	0
Batch size	8
Learning rate(new added layers)	0.001
Learning rate(original layers)	0.0001
Momentum	0.9
Decay for weight&bias	0.0005
Platform and Device	
Platform	Caffe
GPU	NVIDIA GTX TitanX

selected for training, while the remaining one is used for testing. This procedure is repeated for every subject, and an average accuracy is calculated.

4.2.2. Network architecture modifications

As stated in Section 3, we fine-tuned our network based on VGG16 [29] with some transformations.

First, we need to accommodate filters in the first convolutional layer, considering the circumstances of RGB-D images. Under normal conditions, the channels of the filters in this layer correspond to the three classical R, G, and B channels separately. However, RGB-D images have four channels. We adopt the simplest method of taking average of the R, G, and B channels for the newly added depth channel.

Second, the RoI pooling layer, which converts the features inside each region of interest into a small feature map with a fixed length of $H \times W$, where we set $H = W = 7$, is the substitute for the last max pooling layer.

Third, we introduce our attention network between several fully connected layers immediately after the RoI pooling layer and the last fully connected layer for classifying. More details are presented in Section 3.

Lastly, the number of outputs of the last fully connected layer is changed into the number of categories of hand gestures according to the datasets, which is 10 for NTU-HD and 34 for HUST-ASL.

4.2.3. Hyper-parameters for training

When training the network, the batch size is set to eight. Newly added layers are initialized from zero-mean Gaussian distributions with standard deviations of 0.01, and the biases are initialized as 0. Newly added layers have a learning rate of 0.001, which is ten times greater than that of layers loaded from the pre-trained VGG16 weights. The learning rates of all layers will decrease after every 10k iterations. A momentum of 0.9 and decay parameter of 0.0005 (on weights and biases) are employed.

4.2.4. Platform and device

Our experiments are implemented based on the deep learning platform Caffe [13], and the GPU is the NVIDIA GTX Titan X, with 12 GB of memory.

In the following, we will demonstrate the feasibility of our method through the experimental results on the two datasets.

4.3. Localization results

Localizing the hand is indispensable, as the hand always occupies a limited space within an image, which results in a large background area. Most conventional approaches need to localize the hand using a segmentation model, and then perform the classification task. The performance of the latter step is immensely influenced by the former. In the worst case, the hand cannot be segmented out at all, due to many uncontrollable factors.

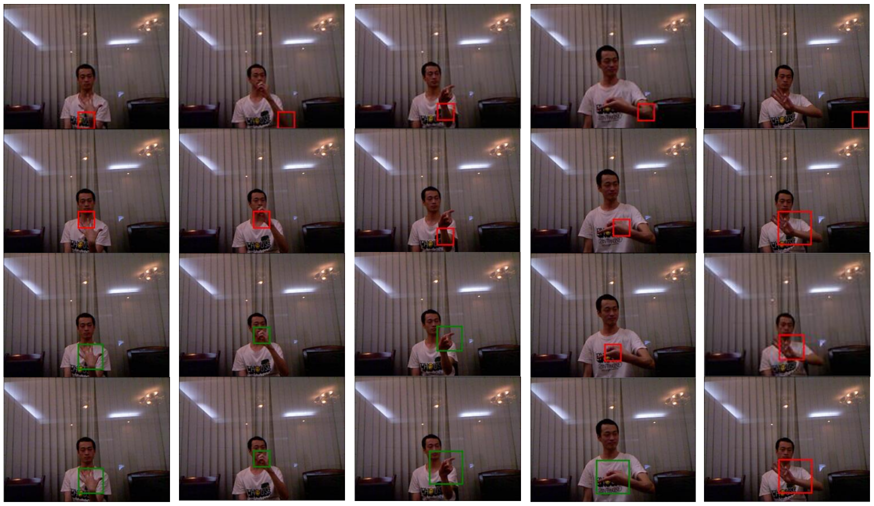


Figure 3: Examples of localization results. There are five columns in total, which represent five different hand gestures we randomly chose in one subject from HUST-ASL. Each line represents the results at different iterations, which are 200, 400, 2000, and 4000, respectively. Green/red rectangles indicate the highest weighted proposals computed by our attention network. Green represents good localization results, while red represents unsatisfactory results.

In our method, we design the attention network to make our network gradually focus on the hand, thereby bypassing the segmentation process. More importantly, the localization process is optimized for the classification task.

Fig. 3 shows some examples of the localization procedure. Each window corresponds to the proposal with the highest-weight computed by our attention network. As the iteration increases, we can observe that the highest weighted proposal can gradually enclose the hand very accurately in most cases.

Let us observe the movement of the attention of our network precisely. At iteration 200, all the different hands are out of our attention. Then, at iteration 400 our attention has already covered a relatively small part of the hand. As expected, our method can roughly localize the hands by iteration 2000, although improvement is still needed. Satisfactorily, the hands have been precisely localized at iteration 4000, except for the last column. Unfortunately, part of the index and middle fingers of the performer are outside of the window in

Table 2: Results on the NTU-HD dataset.

Method	Segmentation	Mean Accuracy(%)
Contour-Matching [24]	✓	93.9
HOG [6]	✓	93.1
H3DF [49]	✓	95.1
Dominant Line [38]	✓	91.1
DPM-BCF [8]	✓	**100.0**
Our method	✗	98.5

the last picture. However, this is not a major cause for concern, because the actual receptive field of the feature map is a little larger than the size of the corresponding window. Therefore, despite the deviation in the localization, our method may still classify the gesture effectively.

We can also observe that the different hand gestures have different scales, while our highest-weighted window can adapt to the scale of the hand gesture, because our sliding window strategy is multi-scale when generating proposals.

We emphasize again here that this localization process is in a weakly supervised learning framework, because we do not provide any region-related labels for this task. We optimize it using only image-level annotations, in an end-to-end manner.

4.4. Classification results

In this section, we compare our method with others on both the NTU-HD and HUST-ASL datasets.

4.4.1. Results on NTU-HD

As there are only 900 training samples in the NTU-HD dataset, we simply compare our method with some conventional approaches. As can be seen in Table 2, the benchmark methods using contour-matching [24], conventional hand-crafted descriptors such as HOG [6], H3DF [49], dominant line [38], and

DPM-BCF [8] are compared with our approach on the NTU-HD dataset. The mean accuracies of these methods are taken from [8].

The NTU-HD dataset is not very challenging, and most of the conventional methods listed above perform well on it. All of the results are higher than 93.0%, and DPM-BCF [8] even achieves a 100.0% accuracy. The accuracy of our method is 98.5%, which outperforms the benchmark method by 4.6%, and is comparable to the state-of-the-art performance. It is important to note that when organizing the data, we find that there exists an apparent deviation between the color images and depth cues, which may have a negative impact on the data. More importantly, we find that most subjects can achieve an excellent performance, such as 100% or 99%, while subject 4 only achieved 94%, which affects our mean accuracy by a large margin. By analyzing the errors, we observe that this is because the fourth performer places his hand too close to the face when performing some gestures, which causes confusion between some fingers and parts of the face.

We deem that NTU-HD dataset is too small, and it cannot adequately train the network. As a result, the powerful performance of deep CNN cannot be fully exerted. Hence, we place more significance on the HUST-ASL dataset.

4.4.2. Results on HUST-ASL

Regarding the challenging HUST-ASL dataset, we first provide experimental evidence to demonstrate the effectiveness of the proposed attention network. We try to train a model without computing the weight for each proposal, i.e., we directly aggregate all the features of the proposals by the global-sum operation after obtaining them. However, after training several times, we find that the model cannot converge. In contrast, our proposed attentional model converges very well. We infer that our model aids in convergence, because it can gradually suppress the weight of the noisy proposals (without hands) and pay special attention to the most important part (hands) in order to obtain the discriminative features of the entire image during training.

We further choose two baseline methods using the pre-trained CNN network

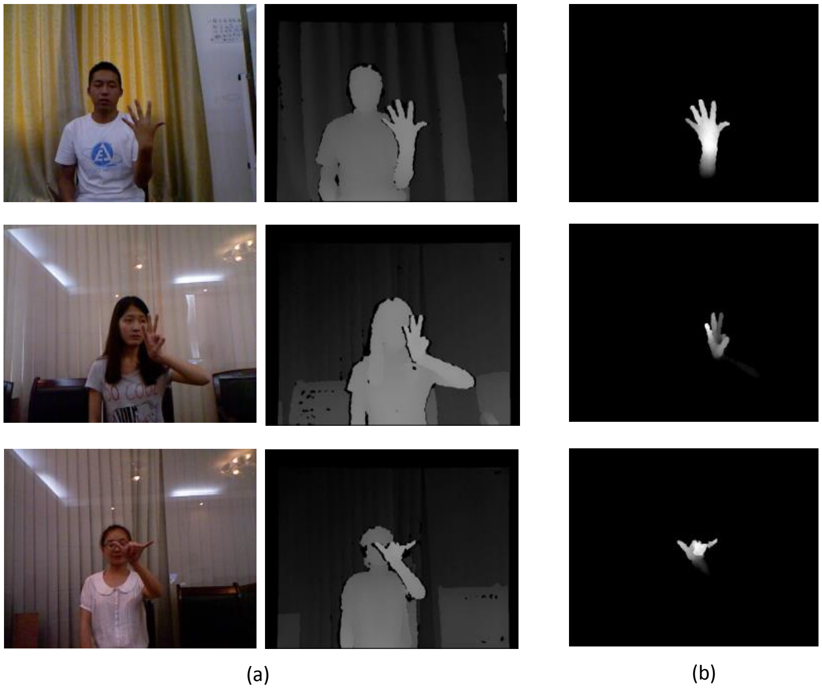


Figure 4: Examples of different inputs of the models. (a) RGB-D images which consists of RGB images and depth images. (b) The depth images of the segmented hands using a depth thresholding algorithm.

VGG16 [29], and fine-tune them using different input data, as shown in Fig. 4. The first baseline method is called VGG16_R, which takes the raw images as its input. The second baseline method is called VGG16_S, in which we first segment the hand by setting the threshold to be 100 mm, as in [8], to obtain the images and feed them into the network to fine-tune the parameters.

Comparisons are presented in Table 3. As mentioned above, the HUST-ASL dataset is very challenging, due to the intricacy of its collection. Although DPM-BCF [8] can achieve a state-of-the-art performance with a 100% mean accuracy on NTU-HD, it sharply declines to 58.0% when faced with the HUST-ASL dataset. Moreover, the accuracy of the contour-matching is 10.8%, which seems to almost lose the capability of classifying the hand gestures. It is gratifying that our method outperforms these methods by a large margin. Concretely, the result of our method is 73.4%, which is 62.6% higher than contour-matching [24], 39.2% higher than HOG [6], and 15.4% higher than BCF-DPM [8]. In addition, there are clear improvements of 40.9% and 6.4%, respectively, compared to the two baseline methods. It is easy to identify that the weak performance of the first baseline method is because it utilizes the features of the entire image with a large background area, while our proposed method exploits the soft attention mechanism to determine the discriminative features. The second baseline method VGG16_S exploits the power of CNNs, compared with hand-crafted features. It should be pointed out that it is unfair to compare this with our method, because it requires segmented images as its input, and will lose efficacy if the hand is not in front of any objects. However, our proposed method is still superior, because we utilize the attention mechanism and treat the localization and classification together.

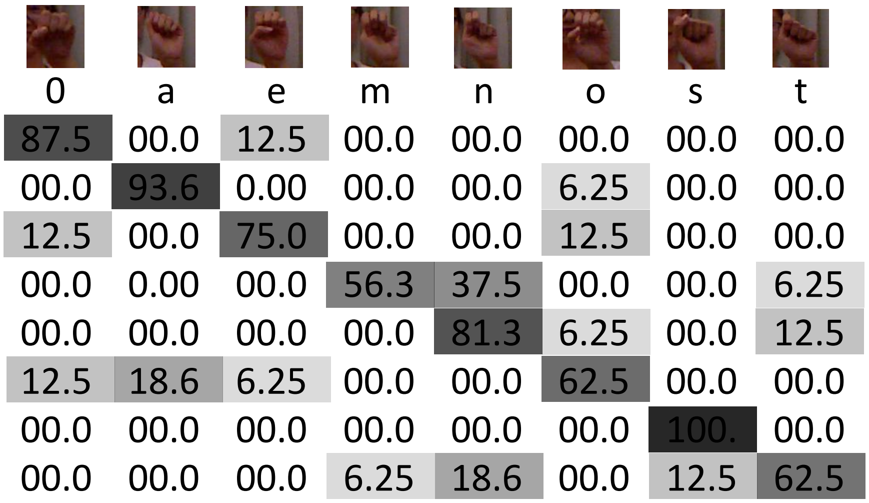
Our method is much faster than the conventional methods in terms of the time consumption. The state-of-the-art method DPM-BCF [8] needs 2.128 seconds, which is too slow for a practical system, while our method only needs 0.726 seconds. The reason why ours is slower than the baseline method VGG16_R is that the structure of VGG16_R is very simple, being constructed by several convolutional and fully connected layers. It is not necessary to compute the features of proposals using the RoI pooling layer, and that method does not utilize the soft attention mechanism, which is the most important part of our method. We deem that it is worth sacrificing some computational simplicity to acquire a tremendous increase in accuracy, from 32.5% to 73.4%.

4.4.3. Error analysis

By analyzing the types of errors on the HUST-ASL dataset, we find that most mistakes result from similar gestures, *e.g.*, the number ***6*** and the letter ***W*** are almost the same, which is even difficult for human beings to distinguish. Moreover, there are a group of hand gestures without any fingers held out, which can be interpreted as a fist. The confusion matrix is shown in Fig. 5. We note

Table 3: Results on the HUST-ASL dataset.

Method	Segmentation	Accuracy(%)	Test Time(s)
Contour-Matching [24]	✓	10.8	4.001
HOG [6]	✓	34.2	N/A
DPM-HOG [8]	✓	36.6	N/A
DPM-BCF [8]	✓	58.0	2.128
VGG16_R	✗	32.5	**0.012**
VGG16_S	✓	67.0	0.013
Our method	✗	**73.4**	0.726



0	a	e	m	n	o	s	t
87.5	00.0	12.5	00.0	00.0	00.0	00.0	00.0
00.0	93.6	0.00	00.0	00.0	6.25	00.0	00.0
12.5	00.0	75.0	00.0	00.0	12.5	00.0	00.0
00.0	0.00	00.0	56.3	37.5	00.0	00.0	6.25
00.0	00.0	00.0	00.0	81.3	6.25	00.0	12.5
12.5	18.6	6.25	00.0	00.0	62.5	00.0	00.0
00.0	00.0	00.0	00.0	00.0	00.0	100.	00.0
00.0	00.0	00.0	6.25	18.6	00.0	12.5	62.5

Figure 5: Confusion matrix of hand gestures without any fingers held out.

that half of these cases are less than 80.0%, and only the letter ***A*** and the letter ***S*** achieve a relatively satisfactory result.

Some possible reasons are listed below. 1) The color images and depth cues are collected by two different kinds of cameras in a Microsoft Kinect, which means that there exists little offset when performing a fusion. 2) The location of the hand is gradually learned by our network itself, with no related annotations.

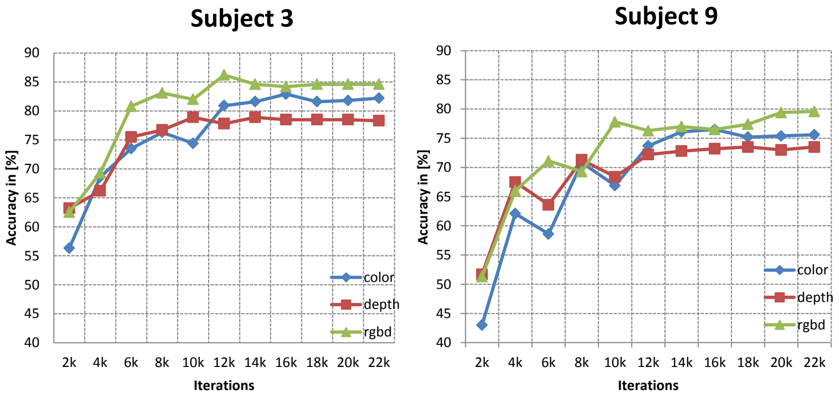


Figure 6: Accuracy curves of two subjects from HUST-ASL for three kinds of inputs, which are color images, depth cues, and RGB-D images.

This process lacks a refinement mechanism, and may lead to unsatisfactory results. 3) Some hand gestures are highly similar, and coupled with the angular diversity in HUST-ASL. This may cause the occlusion of essential features of the gesture and arouse confusion.

4.4.4. Robustness to input

Our method can robustly deal with color images and depth cues. However, the fusion of these types of information will lead to a better performance. Fig. 6 shows the evolving accuracy of two subjects chosen randomly from the HUST-ASL dataset as the number of iterations increases. We can observe that all the curves representing color, depth, and RGB-D images gradually converge after 10000 iterations (about 16 epochs). Above all, the green curves with triangular dots, representing RGB-D images, are clearly better than the others, owing to the fusion of the information.

4.4.5. Robustness to noise

As shown in Fig. 1(a), the most traditional framework of static hand-gesture recognition must segment the hand out from the raw image by assuming that the

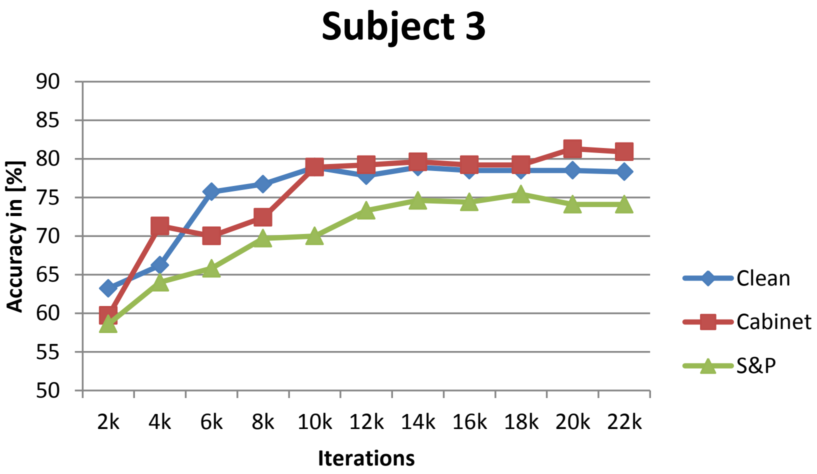


Figure 7: Accuracy curves of subject three from HUST-ASL. *Clean* represents raw depth inputs without deliberately added noise. *Cabinet* indicates that we add a cabinet in depth maps, while *S&P* represents salt and pepper noise.

depth cues of the hand are always in front of the other parts. If this process fails, then it would be impossible to correctly classify gestures. In real-environment applications, this requirement is unreasonable, because real scenes are always complicated, and it is too demanding to ask users to put their hand in front of anything when facing a camera.

In order to demonstrate that our method is robust without segmentation, we introduce two different kinds of noise in depth maps, which can invalidate conventional methods based on segmentation. First, we simulate a circumstance in which there exists a flat cabinet in the foremost position. Therefore, the hand cannot be segmented out by setting a threshold, because the hand is supposed to be in front of the background. Second, we introduce salt-and-pepper noise that presents itself as sparsely occurring white and black pixels. It is obvious that the white pixels can be viewed as the nearest points when we restrict the values in the depth map to [0,255], as shown in Eq. (1), which will damage the segmentation process. However, these two kinds of noise will not influence our method significantly, because we do not need to localize the hand by segmentation. The results of our method are shown in Fig. 7. We observe that the two curves representing *Clean* and *Cabinet* almost overlap, while salt-and-pepper noise drops slightly (by about five percentage points). We conclude that regardless of whether the hand is in front of the background, our method has a considerable discriminative capability for hand gestures.

5. Conclusion

In the course of this work, we have presented a novel static hand gesture recognition method, based on an end-to-end trainable CNN framework with a soft attention mechanism. Our network can automatically localize the hand without any regional annotations, and achieve an excellent performance in classifying gestures. The demand of segmentation for performers' hands to be in front of a specific background is removed. Using this approach, we achieve a comparable mean accuracy result with the previous state-of-the-art method on the NTU-HD dataset, and outperform it on the HUST-ASL dataset by a large margin. Moreover, our method can robustly deal with color images and depth cues. In particular, RGB-D images contribute to the improved performance. To the best of our knowledge, there is no previous work that exploits the discriminative power of CNNs with an attention network for joint hand gesture localization and recognition.

In future work, we will explore how to utilize this CNN framework based on the attention mechanism for dynamic gesture recognition. In addition, we will try to aggregate more information into our network. For example, inferring hand pose for more accurate hand gesture recognition.

Acknowledgements

This research is supported by the Fundamental Research Funds for the Central Universities (NO. 2017KFYXJJ185) and NFSC (NO. 61773176, NO. 61572207 and NO. 61503145).

References

[1] Bai, X., Bai, S., Zhu, Z., Latecki, L.J., 2015. 3d shape matching via two layer coding. IEEE transactions on pattern analysis and machine intelligence 37, 2361–2373.

[2] Bai, X., Latecki, L.J., 2008. Path similarity skeleton graph matching. IEEE transactions on pattern analysis and machine intelligence 30, 1282–1292.

[3] Chen, L.C., Yang, Y., Wang, J., Xu, W., Yuille, A.L., 2016. Attention to scale: Scale-aware semantic image segmentation, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[4] Chen, Y., Pan, D., Pan, Y., Liu, S., Gu, A., Wang, M., 2015. Indoor scene understanding via monocular rgb-d images. Information Sciences 320, 361–371.

[5] Cho, K., Courville, A., Bengio, Y., 2015. Describing multimedia content using attention-based encoder-decoder networks. IEEE Transactions on Multimedia 17, 1875–1886.

[6] Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), IEEE. pp. 886–893.

[7] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE. pp. 248–255.

[8] Feng, B., He, F., Wang, X., Wu, Y., Wang, H., Yi, S., Liu, W., 2017. Depth-projection-map-based bag of contour fragments for robust hand gesture recognition. IEEE Transactions on Human-Machine Systems 47, 511–523.

[9] Girshick, R., 2015. Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448.

[10] He, K., Zhang, X., Ren, S., Sun, J., 2014. Spatial pyramid pooling in deep convolutional networks for visual recognition, in: European Conference on Computer Vision, Springer. pp. 346–361.

[11] Hong, C., Yu, J., Tao, D., Wang, M., 2015a. Image-based three-dimensional human pose recovery by multiview locality-sensitive sparse retrieval. IEEE Transactions on Industrial Electronics 62, 3742–3751.

[12] Hong, C., Yu, J., Wan, J., Tao, D., Wang, M., 2015b. Multimodal deep autoencoder for human pose recovery. IEEE Transactions on Image Processing 24, 5659–5670.

[13] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T., 2014. Caffe: Convolutional architecture for fast feature embedding, in: Proceedings of the 22nd ACM international conference on Multimedia, ACM. pp. 675–678.

[14] Koller, O., Ney, H., Bowden, R., 2016. Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[15] Kuen, J., Wang, Z., Wang, G., 2016. Recurrent attentional networks for saliency detection, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[16] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278–2324.

[17] Lee, C.Y., Osindero, S., 2016. Recursive recurrent nets with attention modeling for ocr in the wild, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[18] Liu, K., Gong, D., Meng, F., Chen, H., Wang, G.G., 2017. Gesture segmentation based on a two-phase estimation of distribution algorithm. Information Sciences 394, 88–105.

[19] Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440.

[20] Mo, Z., Neumann, U., 2006. Real-time hand pose recognition using low-resolution depth images., in: CVPR (2), pp. 1499–1505.

[21] Molchanov, P., Yang, X., Gupta, S., Kim, K., Tyree, S., Kautz, J., 2016. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[22] Oikonomidis, I., Kyriazis, N., Argyros, A.A., 2011. Efficient model-based 3d tracking of hand articulations using kinect., in: BmVC, p. 3.

[23] Pugeault, N., Bowden, R., 2011. Spelling it out: Real-time asl fingerspelling recognition, in: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, IEEE. pp. 1114–1119.

[24] Ren, Z., Yuan, J., Meng, J., Zhang, Z., 2013. Robust part-based hand gesture recognition using kinect sensor. IEEE transactions on multimedia 15, 1110–1120.

[25] Ren, Z., Yuan, J., Zhang, Z., 2011. Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera, in: Proceedings of the 19th ACM international conference on Multimedia, ACM. pp. 1093–1096.

[26] Rogez, G., Khademi, M., Supančič III, J., Montiel, J.M.M., Ramanan, D., 2014. 3d hand pose detection in egocentric rgb-d images, in: Workshop at the European Conference on Computer Vision, Springer. pp. 356–371.

[27] Shao, L., Cai, Z., Liu, L., Lu, K., 2017. Performance evaluation of deep feature learning for rgb-d image/video classification. Information Sciences 385, 266–283.

[28] Shi, C., Pun, C.M., 2017. 3d multi-resolution wavelet convolutional neural networks for hyperspectral image classification. Information Sciences 420, 49–65.

[29] Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. ICLR .

[30] Sinha, A., Choi, C., Ramani, K., 2016. Deephand: Robust hand pose estimation by completing a matrix imputed with deep features, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[31] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9.

[32] Tan, J.H., Fujita, H., Sivaprasad, S., Bhandary, S.V., Rao, A.K., Chua, K.C., Acharya, U.R., 2017. Automated segmentation of exudates, haemorrhages, microaneurysms using single convolutional neural network. Information Sciences 420, 66–76.

[33] Tang, M., 2011. Recognizing hand gestures with microsofts kinect. Palo Alto: Department of Electrical Engineering of Stanford University:[sn] .

[34] Toshev, A., Szegedy, C., 2014. Deeppose: Human pose estimation via deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1653–1660.

[35] Uijlings, J.R., van de Sande, K.E., Gevers, T., Smeulders, A.W., 2013. Selective search for object recognition. International journal of computer vision 104, 154–171.

[36] Vaillant, R., Darmon, D., 1995. Vision based hand pose estimation, in: Proc. Intl Workshop on Automatic Face and Gesture Recognition, pp. 356–361.

[37] Wang, A., Cai, J., Lu, J., Cham, T.J., 2016. Modality and component aware feature fusion for rgb-d scene classification, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[38] Wang, Y., Yang, R., 2013. Real-time hand posture recognition based on hand dominant line using kinect, in: Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on, IEEE. pp. 1–4.

[39] Xiao, T., Xu, Y., Yang, K., Zhang, J., Peng, Y., Zhang, Z., 2015. The application of two-level attention models in deep convolutional neural network for fine-grained image classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 842–850.

[40] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R.S., Bengio, Y., 2015. Show, attend and tell: Neural image caption generation with visual attention. JMLR .

[41] Yamashita, T., Watasue, T., 2014. Hand posture recognition based on bottom-up structured deep convolutional neural network with curriculum learning, in: 2014 IEEE International Conference on Image Processing (ICIP), IEEE. pp. 853–857.

[42] Yang, T., Asanjan, A.A., Faridzad, M., Hayatbini, N., Gao, X., Sorooshian, S., 2017. An enhanced artificial neural network with a shuffled complex evolutionary global optimization with principal component analysis. Information Sciences 418, 302–316.

[43] Yang, Z., He, X., Gao, J., Deng, L., Smola, A., 2016. Stacked attention networks for image question answering, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[44] Yao, L., Torabi, A., Cho, K., Ballas, N., Pal, C., Larochelle, H., Courville, A., 2015. Describing videos by exploiting temporal structure, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 4507–4515.

[45] Yiyi, R., Xie, X., Li, G., Wang, Z., 2016. Hand gesture recognition with multi-scale weighted histogram of contour direction (mswhcd) normalization for wearable applications. IEEE Transactions on Circuits and Systems for Video Technology .

[46] Yoo, D., Park, S., Lee, J.Y., Paek, A.S., So Kweon, I., 2015. Attentionnet: Aggregating weak directions for accurate object detection, in: Proceedings of the IEEE International Conference on Computer Vision, pp. 2659–2667.

[47] Yu, J., Hong, C., Rui, Y., Tao, D., 2017. Multi-task autoencoder model for recovering human poses. IEEE Transactions on Industrial Electronics .

[48] Yu, J., Yang, X., Gao, F., Tao, D., 2016. Deep multimodal distance metric learning using click constraints for image ranking. IEEE transactions on cybernetics .

[49] Zhang, C., Yang, X., Tian, Y., 2013. Histogram of 3d facets: A characteristic descriptor for hand gesture recognition, in: Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on, IEEE. pp. 1–8.

[50] Zitnick, C.L., Dollár, P., 2014. Edge boxes: Locating object proposals from edges, in: European Conference on Computer Vision, Springer. pp. 391–405.