2019 27th European Signal Processing Conference (EUSIPCO)

Fingerspelled Alphabet Sign Recognition in Upper-Body Videos

Katerina Papadimitriou and Gerasimos Potamianos
Electrical and Computer Engineering Department, University of Thessaly, Volos 38221, Greece
aipapadimitriou@uth.gr gpotam@ieee.org

***Abstract*—Fingerspelling is a crucial part of sign-based communication, however its recognition remains a challenging and mostly overlooked computer vision problem. To address it, this paper presents a system that recognizes the 24 static fingerspelled alphabet signs of the American Sign Language. The system consists of two algorithmic stages, comprising an efficient preprocessing phase that generates candidate hand-region proposals, followed by their deep-learning based classification. Specifically, the first stage exploits own earlier work on hand detection and segmentation in videos that also contain the signer's face, allowing face detection to drive skin-tone based hand segmentation, with motion further utilized to localize hands, extending it with a peak detection module that yields proposal regions likely to contain the signs of interest. These regions are then classified by a variant of a convolutional neural network that extends traditional convolutions to quadratic operations on the inputs, being, to our knowledge, the first application of such architecture to this task. Both system stages are evaluated on three well-known fingerspelling corpora, significantly outperforming a number of alternative approaches under both multi-signer and signer-independent experimental frameworks.**

***Index Terms*—Fingerspelling, ASL, CNN, detection, classification**

I. Introduction

Sign language recognition constitutes a popular field of research that has attracted increasing interest over the last decade, due to its potential of meeting the communication needs of the speech and hearing impaired [1]–[3]. However, while there have been dramatic breakthroughs in oral speech technologies, less progress has been observed in the domain of sign languages, primarily due to the challenging nature of the underlying computer vision problem, e.g. the visual similarity of specific signs and the hand articulatory complexity.

Among the sign language recognition schemes in the literature, a limited number of works have focused on the problem of fingerspelling recognition. Fingerspelling constitutes a critical component of sign-based communication, as it is commonly used for prominent words lacking unique signs, such as names, technical terms, or foreign words. Example of such works include [4], [5], where convolutional neural networks (CNNs) are used for static American Sign Language (ASL) fingerspelling alphabet recognition from depth-map and color images, the system in [6] that recognizes 20 out of 24 static ASL signs through PCA extracted features, as well as [7] that employs CNNs with multiview augmentation and inference fusion from depth images for this task. Further, in [8] histograms of oriented gradients (HOG) and Zernike moment feature extractors are used in conjunction with a deep belief network classifier, while in [9] a system that exploits hand tracking devices and an SVM classifier is introduced. Additionally, the survey in [10] uses LBP histogram features based on color and depth information with an SVM classifier for recognition, while the fingerspelling ASL recognition systems in [11], [12] rely on semi-Markov conditional random fields and recurrent neural networks using deep neural network-based features.

In this paper, we address the problem of recognizing static signs of the ASL fingerspelling alphabet in video streams, focusing on handshapes rather than motion. Our approach is based on two distinct pillars: handshape extraction and subsequent multi-class classification. Hence, we introduce a hybrid, vision-based, two-stage system for effective handshape extraction through an image processing pipeline and hand-posture classification based on a CNN variant, as also schematically depicted in Figure 1.

Specifically, for the first stage of the proposed system, we exploit the image processing scheme of our earlier work [13], aiming to efficiently detect very few proposal windows likely to contain the signs of interest, which will be further examined by the system's second, classification stage. That work primarily incorporates facial information obtained by a fast off-the-shelf face detector [14] into skin-tone based hand segmentation [15]. For that purpose, it is assumed that the videos under consideration also include relatively frontal head-pose data, which is typically the case in the sign language domain since facial information is crucial to signing. The aforementioned components are complemented by motion-based Kalman filtering [16] for hand tracking, as well as handshape segmentation by Otsu's thresholding [17]. The pipeline of [13] is further extended in this paper by a peak detection module to yield candidate handshapes most likely to "express" alphabet signs.

Subsequently, these are fed to a modified CNN classifier for static ASL fingerspelled alphabet sign recognition. Most deep learning classifiers are known to require large amounts of labeled data to generalize well, but such data are limited for fingerspelling. To counter the issue, and inspired by the prior work of [18], [19], we propose an altered convolution operation to improve CNN learning capacity. Specifically, focusing on the convolution scheme, instead of it being a linear combination of a filter matrix and the input image elements,

978-9-0827-9703-9/19/$31.00 ©2019 IEEE

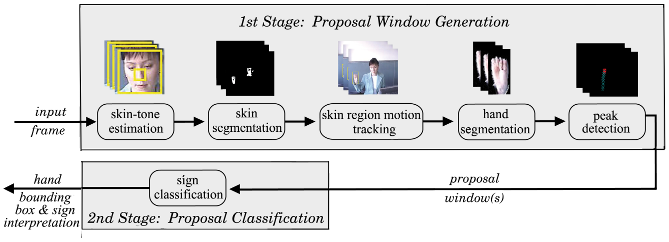

1st Stage: Proposal Window Generation
input frame
skin-tone estimation
skin segmentation
skin region motion tracking
hand segmentation
peak detection
proposal window(s)
sign classification
2nd Stage: Proposal Classification
hand bounding box & sign interpretation

Fig. 1. Block diagram of the proposed two-stage system.

we propose a non-linear form produced by a quadratic function of the inputs. Transforming the inputs through such a function contributes to the non-linear behavior of the network output, aiming to provide better generalization. At the same time, inclusion of the preprocessing stage improves accuracy and yields low computational cost, as the CNN classifier is fed with very few regions that are likely to contain hand-signs.

Further details of the proposed system are provided in Section II. The approach is benchmarked on three video fingerspelling datasets, as presented in Section III, where both system stages are evaluated. In particular, both stages are found to significantly outperform a number of alternative approaches under two experimental frameworks: multi-signer and signer-independent, the latter particularly demonstrating the successful generalization achieved. Finally, the paper conclusions are summarized in Section IV.

II. Proposed model

This section describes in detail the proposed methodology for performing ASL fingerspelled alphabet recognition from video data. As already discussed, it comprises two main stages: (i) a preprocessing pipeline, slightly modified from [13], and (ii) a classification framework based on a CNN variant, both detailed next.

A. Preprocessing stage

1) Skin-tone estimation based on face detection: To best capture the skin color range, the proposed pipeline, as described in [13], commences with face detection by means of the Viola-Jones algorithm [14]. After successful face detection, the central rectangular region of the facial bounding box (nose area) is extracted (see also Figures 2(a)-(b)) and converted to the YCbCr color space, in order to drive the skin segmentation step that follows. In case of face detection failure, the image frame is subjected to skin segmentation in the YCbCr color space based on specific threshold values [15], instead of the skin segmentation step below.

2) Skin segmentation: Based on the skin-tone information provided by the extracted nose region above, skin region segmentation in the YCbCr color space is applied. Specifically, after the image frame transformation to the YCbCr color space (see Figure 2(c)), skin pixels are classified based on the range of the corresponding YCbCr values of the extracted nose region. Subsequently, morphological operations are applied for noise elimination, and a binary image is generated (see Figure 2(d)).

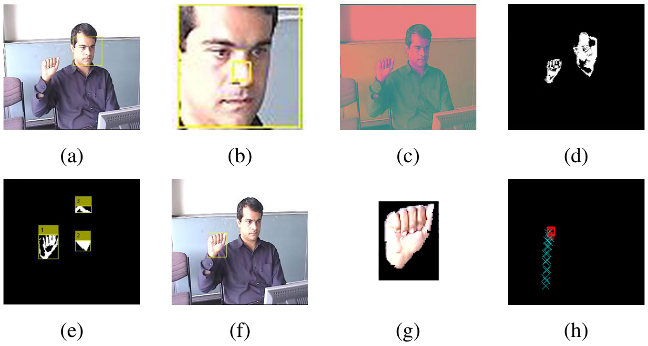


Fig. 2. Preprocessing pipeline example: (a) Input image marked with a rectangular box enclosing the detected face; (b) the central square of the detected face region (zoomed-in); (c) input image converted to the YCbCr color space; (d) segmented skin region; (e) binary image with detections encompassed with rectangular bounding boxes; (f) resulting image with the yellow rectangular box illustrating the moving object (hand); (g) segmented hand; and (h) hand trajectory (sequence of upper-left hand coordinates) with red squares denoting handshapes retained after peak detection.

3) Skin region motion tracking: To avoid incorrect skin region detection, in case of areas of similar skin-tone but containing no skin, after skin segmentation and under the assumption that hands are moving objects, we employ Kalman filtering [16]. As described in detail in [13], the main idea is to forward the location of the skin regions previously extracted (excluding the face region) to the Kalman filter, in order to associate each of them with a related track, thus rejecting detections that do not correspond to moving objects (see Figures 2(e)-(f)).

4) Hand segmentation: This step focuses on the subtraction of the background from the rectangular bounding boxes generated in the previous step, employing Otsu's thresholding method [20]. This process is critical, since it generates bounding boxes including only the target objects to be fed to the classification stage, which constitute the so-called proposal windows. An example of Otsu's thresholding is shown in Figure 2(g), with pixels not satisfying the optimal threshold changed to zero.

5) Peak detection: The number of proposal windows provided to the classification stage is determined using peak detection. Specifically, a peak is defined as the point where the hand bounding box upper y-coordinate remains "stable" for more than one frame, and only proposal windows of two adjacent frames with upper y-coordinates Euclidean distance less than 8 are marked as a peak (see Figure 2(h)). The peak frames for each letter tend to be characterized by limited motion, which is estimated to be less than 8 pixels between hand position transitions. This peak is commonly the point where the handshape configuration most closely resembles the canonical handshape and, by extension, the letter sign. Note that, in case of multiple tracks of hand bounding boxes, this step is individually applied to each track.

B. Classification stage

Following the proposal window selection, a number of rectangular regions are returned as hand candidates, excluding of course any face bounding boxes. To yield the static hand gesture recognition, a CNN is employed, adopting the AlexNet architecture [21] after appropriate modification and training. It should be noted that the developed CNN is multi-class, corresponding to the 24 static letters of the ASL alphabet, excluding "J" and "K" because they are formed by hand movements, and including a "no hand" class. In more detail, each proposal window is resized to the fixed size of the CNN input layer (227×227 pixels) and fed to it in order to predict its label. As already mentioned, the CNN follows the AlexNet architecture [21], based on its wide adoption by the computer vision community and high accuracy achieved on the ImageNet benchmark [22]. The network consists of five convolutional and three fully-connected layers, and it is pretrained on the ImageNet corpus. The only modification made is on its final fully-connected layer, so that it has the same size as the number of classes of interest (25).

In the literature, in the case of CNNs, non-linearities have been mainly deployed through activation functions and pooling operations, with limited only attention paid to the filtering mechanics. Here, motivated by the non-linear nature of image data and inspired by recent works [18], [19], we consider an alternative convolutional operation that outputs the feature map based on a non-linear quadratic function. More precisely, our model follows the regular CNN layer pipeline (convolution, pooling, activation function, etc.). However, instead of performing the traditional linear operation

$$F(x) = w^\top x + b \,, \tag{1}$$

with x being the input vector, w representing the weight or filter vector, and b denoting the bias value, we employ the non-linear quadratic based scheme

$$F(x) = w^\top (x \odot x) + b \,, \tag{2}$$

where $\odot$ denotes element-wise vector product. The rationale lies on the quadratic function curvature that improves network learning by representing non-linear complex functional mappings from inputs to outputs, while its gradient remains smooth, depending on x during backpropagation as

$$\frac{\partial F}{\partial x} = 2w^\top x \,. \tag{3}$$

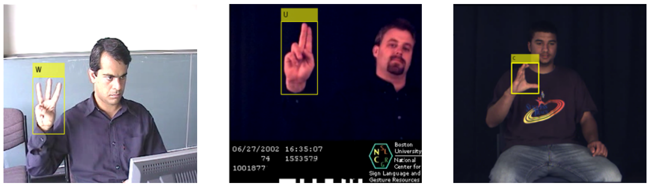
To further enhance the sparsity and efficiency of the activations, we replace the rectified linear units (ReLU), which often result in weight updates that cause inactive neurons, with LeakyReLU ones.

III. Experiments

A. Video datasets

Our experiments are conducted on three publicly available databases: the RWTH German fingerspelling dataset [23], the National Center for Sign Language and Gesture Resources (NCSLGR) handshapes corpus [24], and the American Sign Language Lexicon Video Dataset (ASLLVD) [25].

The RWTH German Fingerspelling dataset comprises video sequences with fingerspelling letters of the German sign language, including among others the signs "A" to "Z", which are identical to the corresponding letters of ASL, except for letter "T" that is different. The data are organized in sets of 44 sequences for every letter, recorded by 20 different users. Videos are available at a 30 Hz frame rate and a frame resolution of 352×288 pixels. The body parts, like hands and face, for 4,416 image frames, corresponding to 44 videos for each letter, apart from letters "J" and "K" which include motion, were manually annotated as part of this work.


W
U
C
06/27/2002 16:35:07
74 1553579
1001877
Boston University National Center for Sign Language and Gesture Resources

(a) (b) (c)

Fig. 3. Hand detection and letter classification for (a) the RWTH German fingerspelling database, (b) the NCSLGR corpus, and (c) the ASLLVD dataset.

The NCSLGR handshapes corpus consists of 87 ASL videos including both letters and numbers, generated and linguistically labeled at Boston University. The database consists of single-signer videos for every letter, showing the signing from three front cameras including the face region, and a hand closeup view containing only hands. Image frames are available at a resolution of 312×324 pixels. A manual ground-truth hand and face annotation for 1,168 frames recorded by the front-view cameras arising from 72 videos was conducted for the purposes of this work.

The American Sign Language Lexicon Video Dataset (ASLLVD) is a large dataset of video sequences of almost 3000 isolated ASL signs produced by 6 native signers of ASL for a total of almost 9,800 tokens, including number signs, fingerspelled signs, and lexical ones. Signing is captured simultaneously by four cameras, providing a side view, two frontal views, and a face closeup view. For the side view, the first frontal view, and the face closeup, videos are available at 60 Hz and 640×480-pixel resolution. Additionally, for the second frontal view, video is captured at 30 Hz, non-interlaced, with a frame resolution of 1600×1200 pixels. Hand and face ground-truth annotations of every frame are publicly available. Frontal-view camera recordings and 30% of all image frames corresponding to 3,700 images are employed in this work.

B. Algorithm implementation details

The evaluation of the algorithm was run on a CPU architecture (i7-6700HQ, 2.60 GHz processor), whereas CNN training was carried out on a Nvidia GTX 1050 Ti GPU. Both training and evaluation were implemented within the Matlab environment.

Network training employed stochastic gradient descent with momentum with an initial learning rate of 0.004 (decreasing by a factor of 0.5), performing 60 complete passes over the data. The mean squared error loss function and a mini-batch size of 128 images are used.

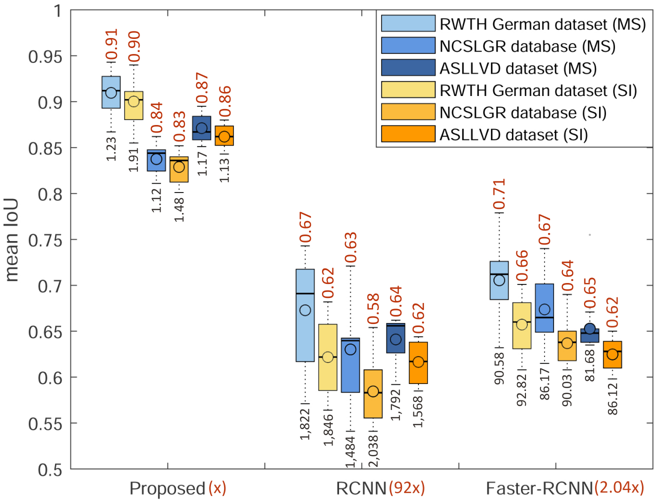


Fig. 4. Boxplot for comparing the detection performance of the proposed method against two alternatives in both multi-signer (MS) and signer-independent (SI) cases on all three evaluation datasets in terms of mean IoU. The average number of proposal windows per frame is listed in green, the mean IoU in red, and the running time is shown inside parentheses.

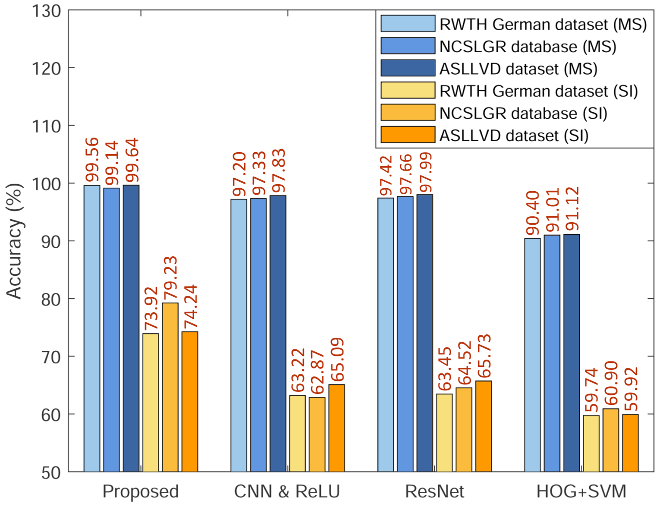


Fig. 5. Performance comparison of the proposed method against three alternatives in both multi-signer (MS) and signer-independent (SI) cases on all three evaluation datasets in terms of classification accuracy (listed in red).

C. Experimental setup

We follow two experimental paradigms in our evaluation concerning the training and test set split, as detailed next.

1) Multi-signer: In this case, we fine-tuned the pretrained CNN of Section II-B on data from all three corpora of Section III-A, using the parameter setup of Section III-B. Specifically, for training we utilized 2,279 handshapes from the RWTH German fingerspelling dataset, 606 samples from the NCSLGR database, and 2,188 images of ASLLVD, corresponding to 50% of their available videos (containing manually extracted handshapes). For testing, sets were generated after preprocessing the remaining videos with the pipeline of Section II-A, resulting in 2,137 frames from the first, 562 from the second, and 1,512 from the third database.

2) Signer-independent: To validate the generalization ability of our model, we trained the CNN of Section II-B on two datasets at a time (thus resulting in three models), and tested on 30% of all videos of the third database each time (after preprocessing). Specifically, we tested the respective CNN on 1,367 image frames from the RWTH German dataset, 350 from the NCSLGR database, and 1,110 from ASLLVD.

D. Detection efficiency

The performance of the proposed multi-class scheme, including the preprocessing phase and the nonlinear model, was initially evaluated in terms of detection efficiency and compared against two well-established alternatives: the R-CNN approach [26], which was considered with five CNN layers comprised of linear convolutions and ReLU activations, but employing selective search for proposal generation [27], as well as its efficient implementation known as the Faster R-CNN [28] that is a region proposal network sharing convolutional layers with the Fast R-CNN [29]. For comparison purposes, both models were provided with the frames extracted by the peak detection step including only canonical handshapes of the letter signs. Regarding the detection efficiency, the aforementioned models were compared in terms of the mean Intersection-over-Union (mean IoU) [30], which is a standard metric to measure the overlap ratio between ground-truth and predicted bounding boxes.

As can be readily observed in Figure 4, the proposed method reaches the highest IoUs on all three datasets and under both experimental paradigms, attaining a small interquartile range (spread) of values. This is primarily due to the smaller number of proposal windows that reduces the risk of false detections, while not missing hand candidate regions due to the robust design of the first stage of the algorithm. Indeed, on the average, the proposed algorithm yields 1.34 proposals per frame, versus 1,758 of the R-CNN and 87.9 of the Faster-RCNN. As a result, the R-CNN runs significantly slower than the proposed by approximately 92 times, while the Faster R-CNN manages to reduce the latter, remaining though still at 2.04 times slower. Note that in its current Matlab-based implementation (see Section III-B), the proposed model (including both its preprocessing and classification stages) runs at approximately 0.28 sec per frame.

E. Classification accuracy

Further, the classification accuracy of the proposed approach was evaluated against two variations based on the preprocessing pipeline for hand segmentation and a CNN classifier with a linear convolutional operation and a ReLU layer [4] (“CNN & ReLU”), as well as a ResNet-50 classifier [31] (“ResNet”), using the training options described in Section III-B. Moreover, it was also compared against a “HOG+SVM” system, employing a HOG feature extractor (64-dimensional features) and an SVM classifier that was provided with frames including only canonical handshapes.

It can be readily observed from Figure 5 that the proposed model turns out superior to the considered alternatives in both multi-signer and signer-independent cases and on all evaluation datasets. This can be attributed to the non-linear operations of (2) and the effective activations of the LeakyReLU layer, yielding accuracies ranging within 99.14% and 99.64% for the three sets in the multi-signer case and between 73.92% and 79.23% in the signer-independent case. It is worth noting that the largest absolute accuracy improvements occur in the signer-independent case. Specifically, such improvements range between 9.15% and 16.36% against the "CNN & ReLU" system across the three datasets, between 8.51% and 14.71% compared to the "ResNet", and even larger, between 14.18% and 18.33%, against the "HOG+SVM" system.

IV. Conclusions

This paper presents a hybrid approach to effectively solve the problem of automatic detection and classification of static ASL fingerspelled alphabet signs in video data. The method combines deep learning with traditional image processing methods, assuming visibility of the face to guide skin-tone based segmentation, assisted by motion-based tracking of the segmented skin regions. The substitution of the linear convolutional operation by a quadratic function in the CNN architecture is shown to improve performance significantly, especially in the mismatched signer-independent case, showcasing its better generalization ability.

References

[1] H. Cooper, E.-J. Ong, N. Pugeault, and R. Bowden, "Sign language recognition using sub-units," in *Gesture Recognition*, S. Escalera, I. Guyon, and V. Athitsos, Eds., pp. 89–118. Springer, 2017.

[2] R. Cui, H. Liu, and C. Zhang, "Recurrent convolutional neural networks for continuous sign language recognition by staged optimization," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1610–1618.

[3] P. Kumar, P. P. Roy, and D. P. Dogra, "Independent Bayesian classifier combination based sign language recognition using facial expression," *Information Sciences*, vol. 428, pp. 30–48, 2018.

[4] B. Kang, S. Tripathi, and T. Q. Nguyen, "Real-time sign language fingerspelling recognition using convolutional neural networks from depth map," in *Proc. IAPR Asian Conference on Pattern Recognition (ACPR)*, 2015, pp. 136–140.

[5] S. Ameen and S. Vadera, "A convolutional neural network to classify American Sign Language fingerspelling from depth and colour images," *Expert Systems*, vol. 34, no. 3, 2017.

[6] S. Upendran and A. Thamizharasi, "American Sign Language interpreter system for deaf and dumb individuals," in *Proc. International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, 2014, pp. 1477–1481.

[7] W. Tao, M. C. Leu, and Z. Yin, "American Sign Language alphabet recognition using convolutional neural networks with multiview augmentation and inference fusion," *Engineering Applications of Artificial Intelligence*, vol. 76, pp. 202–213, 2018.

[8] Y. Hu, H.-F. Zhao, and Z.-G. Wang, "Sign language fingerspelling recognition using depth information and deep belief networks," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 32, no. 06, 2017.

[9] L. Quesada, G. López, and L. Guerrero, "Automatic recognition of the American sign language fingerspelling alphabet to assist people living with speech or hearing impairments," *Journal of Ambient Intelligence and Humanized Computing*, vol. 8, no. 4, pp. 625–635, 2017.

[10] C. S. Weerasekera, M. H. Jaward, and N. Kamrani, "Robust ASL fingerspelling recognition using local binary patterns and geometric features," in *Proc. International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2013.

[11] B. Shi and K. Livescu, "Multitask training with unlabeled data for end-to-end sign language fingerspelling recognition," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 389–396.

[12] T. Kim, J. Keane, W. Wang, H. Tang, J. Riggle, G. Shakhnarovich, D. Brentari, and K. Livescu, "Lexicon-free fingerspelling recognition from video: Data, models, and signer adaptation," *Computer Speech and Language*, vol. 46, pp. 209–232, 2017.

[13] K. Papadimitriou and G. Potamianos, "A hybrid approach to hand detection and type classification in upper-body videos," in *Proc. European Workshop on Visual Information Processing (EUVIP)*, 2018.

[14] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[15] K. B. Shaik, P. Ganesan, V. Kalist, B. S. Sathish, and J. M. M. Jenitha, "Comparative study of skin color detection and segmentation in HSV and YCbCr color space," *Procedia Computer Science*, vol. 57, pp. 41–48, 2015.

[16] J.-M. Jeong, T.-S. Yoon, and J.-B. Park, "Kalman filter based multiple objects detection-tracking algorithm robust to occlusion," in *Proc. SICE Annual Conference (SICE)*, 2014, pp. 941–946.

[17] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[18] N. Tsapanos, A. Tefas, N. Nikolaidis, and I. Pitas, "Neurons with paraboloid decision boundaries for improved neural network classification performance," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 1, pp. 284–294, 2019.

[19] G. Zoumpourlis, A. Doumanoglou, N. Vretos, and P. Daras, "Non-linear convolution filters for CNN-based learning," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 4771–4779.

[20] P.-s. Liao, T.-s. Chen, and P.-c. Chung, "A fast algorithm for multilevel thresholding," *Journal of Information Science and Engineering*, vol. 17, pp. 713–727, 2001.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS) 25*, 2012, pp. 1097–1105.

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

[23] P. Dreuw, T. Deselaers, D. Keysers, and H. Ney, "Modeling image variability in appearance-based gesture recognition," in *Proc. ECCV Workshop on Statistical Methods in Multi-Image and Video Processing (ECCV-SMVP)*, 2006, pp. 7–18.

[24] C. Neidle, "SignStream: A database tool for research on visual-gestural language," *Sign Language and Linguistics*, vol. 4, no. 1, pp. 203–214, 2001.

[25] V. Athitsos, C. Neidle, S. Sclaroff, J. Nash, A. Stefan, A. Thangali, H. Wang, and Q. Yuan, "Large Lexicon Project: American Sign Language video corpus and sign language indexing/retrieval algorithms," in *Proc. LREC Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, 2010, pp. 11–14.

[26] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142–158, 2016.

[27] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[28] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NIPS) 28*, 2015, pp. 91–99.

[29] R. Girshick, "Fast R-CNN," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[30] M. A. Rahman and Y. Wang, "Optimizing Intersection-Over-Union in deep neural networks for image segmentation," in *Proc. International Symposium on Visual Computing (ISVC)*, 2016, pp. 234–244.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.