Hand-Over-Face Segmentation

by

SAKHER GHANEM

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

August 2020

Hand-Over-Face Segmentation

The members of the Committee approve the doctoral
dissertation of Sakher Ghanem

Vassilis Athitsos

Supervising Professor _____

Farhad Kamangar _____

Christopher Conly _____

Christopher McMurrough _____

Dean of the Graduate School _____

ACKNOWLEDGEMENTS

First and foremost, thanks to my great Lord for giving me the health and power needed to accomplish this goal of gaining and completing my Ph.D. degree. Thanks to my greatest teacher prophet Mohammad, peace be upon him, who encouraged us to obtain knowledge from birth until death.

I am genuinely thankful to my supervising professor Dr. Vassillis Athitsos, who I am honored to be one of his Ph.D. students. I highly respect and appreciate his consistent help, support, guidance, patience, understanding, encouragement, and insightful advice during my entire time in my doctoral journey. I would also like to thank my academic committee members: Dr. Farhad Kamangar, Dr. Christopher McMurrough, and Dr. Chris Conly, for their interest, assistance, and valuable feedback.

Further, I would like to extend my appreciation to all UTA-CSE faculty/staff members. I am particularly thankful to Dr. Ramez Elamsri, Dr. Bahram Khalili, Camille Costabile, Sherri Gotcher, Pamela Mcbride, and Ginger Dickens for their efforts to ease and smooth all administration process for all graduate students. I also thank all colleagues in the VLM lab for their help, cooperation, and knowledge sharing. I will never forget the kind people who volunteered in the VLM-HandOverFace dataset.

My sincere gratitude goes to the University of Jeddah and the government of Saudi Arabia for funding me the whole doctoral program.

Finally, I am incredibly thankful and grateful to my beloved parents: Fuad Ghanem and Khadijah Alhebaishe, for their unconditional support, guidance, en-

couragement, and prayers. My success will never be achieved without both of them. Many thanks to my brother Basim, my sisters Maysa, Ghofran, and Batool. With my infinite love, I want to give my exceptional thanks to my wife, Sarah, our kids Khadijah, Elyas, and Ouais, for their patience, support, and encouragement. I also want to express my appreciation to my father and mother in law: Captin. Hani Jamaluddin and Fatin Altunsi. Many thanks to Mr. Fahad Aldada, who left before seeing the completion of my degree, for his moral support during my stay in Texas. My deepest and sincere gratitude goes to Professor Sami Halawani, who never stop educating, helping, and guiding me during my academic life. Also, I would like to appreciate all my teachers and mentors, especially: Prof. Nabih Baeshen, Prof. Khalid Thabit, Prof. Osama Abolnaja, Prof. Hassanain Albarhamtoshi, Prof. Khalid Fakeeh, Prof. Osama Abuzinadah, and Shaikh Mohammad Abu Khalil. Lastly, thanks to all my amazing friends, especially: Bander Badrieg, Ayman Damanhouri, Karim Qumosani, Abdullah Bokhary, Bakur AlQaudi, Sami Alesawi, Mousa Almotairi, Tariq Alsahafi, and Ashiq Imran.

July 27, 2020

ABSTRACT

Hand-Over-Face Segmentation

Sakher Ghanem, Ph.D.

The University of Texas at Arlington, 2020

Supervising Professor: Vassilis Athitsos

Accurate hand segmentation is vital in many applications in which the hands play a central role, such as sign language recognition, action recognition, and gesture recognition. A relatively unexplored obstacle to correct hand segmentation is when the hand overlaps the face. The shortage of a dataset for this research area has been one motivation for this work. However, this dissertation investigates and proposes improvements for the hand-over-face segmentation task.

Toward an in-depth study of the hand segmentation problem, the work presented in this dissertation will yield several contributions. First, it introduces a survey on sign language recognition systems using mobile phones, which shows a recent practical example of the need for the hand segmentation dataset and comprehensive research work. Second, following the context of this work, a literature review that covers and summarizes all available hand segmentation datasets will be presented. Besides, I provide a public dataset (VLM-HandOverFace) for hand segmentation task. This newly constructed dataset contains 4384 labeled frames and includes color, depth, infrared streams recorded by Kinect. The performance of the VLM-HandOverFace dataset is evaluated using several state-of-the-art architectures.

Furthermore, this dissertation proposes the Multi-level Pyramid Scene Parsing Network (MPSP-Net) for semantic segmentation. I also provide a thorough discussion and evaluations of the new modeled-solution about the unique characteristics that demonstrate its applicability for the hand-over-face segmentation challenge.

Several experiments were conducted to examine MPSPNet using two object segmentation datasets and two hand segmentation datasets. The results show that the proposed method achieves at least a 6% improvement in mIOU compared with all state-of-the-art methods. Finally, various experiments conducted to measure the impact of including temporal motion information on MPSPNet.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Computer vision has shown an increasingly significant role while assisting computers in extracting and analyzing various pieces of information from images. One of the challenging problems in computer vision is hand segmentation. An additional, yet relatively unexplored, challenge is when hands overlap the face. Hand segmentation is a crucial part of many computer vision applications such as human-computer interaction, gesture recognition, activity recognition, and sign language recognition. In practice, accurate hand segmentation encounters many challenges due to the high variation of lighting, skin colors, and complex backgrounds.

Sign language recognition is a hot topic in computer vision that still desires plenty of effort. Generally, there are two approaches employed in this field. First, is by using gloves sensors, which is costly and not favored by users. Second, is a vision-based technique that utilizes a camera as an input. Nowadays, a camera is embedded with almost all electronic devices, making it a useful sensor in several areas. In the vision-based approach, there are two main steps to complete the task of gesture recognition: (1) hand detection, and (2) classification of hand shape and motion. Therefore, the hand(s) is the dominant element in any such model. According to the American Sign language Dictionary [1], there are numerous signs where hand(s) overlap with the face. The availability of smartphones, which equipped with a high-resolution camera and multi-processor CPU, encouraged many researchers to benefit from mobility advantage [2]. Therefore, a designated chapter of this disser-

tation contributed a survey covering the sign-language recognition-system that uses smartphones.

Research on computer vision topics benefits from the invention of challenging public datasets. Those datasets can be used to benchmark existing methods and to highlight lacks of enhanced performance. Two factors are essential to consider in any hand segmentation dataset: pixel-wise ground truth labels, and the quantity of annotated frames. There are several hand segmentation datasets for egocentric purposes, but it is limited for hand-over-face segmentation. This dissertation addresses and compares all known public hand datasets. Also, it describes the new constructed dataset that fulfills the emerge of shortage. The new dataset, which called VLM-HandOverFace [3], contains more than 4000 labeled images, and it is publicly available for academic purposes.

The process of identifying each pixel in an image belongs to which class is called semantic segmentation. In this dissertation, I draw more attention to the hand-over-face segmentation problem. Generally, two methods are adopted with the semantic segmentation problem: (1) probabilistic approach, and (2) deep learning technique. In recent years, Convolutional Neural Networks (CNN) archived promising results in segmentation task. However, even though there are several hand(s) segmentation attempts for egocentric application, fewer works are published for hand-over-face segmentation. Therefore, several semantic segmentation state-of-the-art networks are tested on the new VLM-HandOverFace dataset. And Multi-level Pyramid Scene Parsing Network (MPSPNet) [4] is introduced to handle hand segmentation challenge. The size cascading configuration of the network, as well as the pyramid scene parsing processing, make it a unique design to enhance the segmentation results.

Since VLM-HandOverFace come up with the video files for advanced research, it is relevant to measure the impact of adding temporal data to the MPSP-Net. Two

types of trails are investigated: (1) using optical flow, and (2) adding previous/next RGB frames. (more information in chapter 5).

1.1 Dissertation Contributions

The focus in this dissertation is on hand-over-face segmentation problem. The work presented in the following chapters will make the following contributions:

1. Reviewing all existing models for sign language recognition using smartphones.
2. Surveying all available hand segmentation datasets and explore the advantages and disadvantages of each one.
3. Enriching the field with a challenging public dataset (VLM-HandOverFace) to address the lack of having appropriate hand-over-face dataset.
4. Introducing the Multi-level Pyramid Scene Parsing Network (MPSP-Net) for semantic segmentation.
5. Demonstrating the unique properties that make (MPSP-Net) suitable for hand-over-face segmentation challenge.
6. Applying MPSPNet on temporal video information from the VLM-HandOverFace dataset and measuring its impact on the segmentation result.

1.2 Dissertation Organization

Chapter 2, aims to cover the most recent techniques in mobile-based sign language recognition systems. It shows a survey on sign language recognition using smartphones. The literature review primary focus is on two main aspects of sign language recognition: feature detection and sign classification algorithms.

In Chapter 3, a survey on hand segmentation datasets is demonstrated. Then, a challenging public dataset for the hand-over-face segmentation problem is presented.

The new dataset contains 4384 annotated frames and includes color, depth, and infrared streams recorded by Kinect.

The Multi-level Pyramid Scene Parsing Network (MPSPNet) for semantic segmentation is proposed in Chapter 4. An evaluation of MPSPNet is performed on two object segmentation datasets (NYUDv2, PASCAL VOC) and two recently published and challenging datasets focusing on scenarios in which the hands overlap the face, VLM-HandOverFace and HOF. Additionally, a comparison between the new method and several state-of-the-art hand segmentation methods such as RefineNet and PSPNet is presented.

Finally, in chapter 5, I examine the impact of adding motion information from the VLM-HandOverFace dataset to MPSPNet.

1.3 Published Papers

As a result of my research, some articles were published during my Ph.D. study. The following are the published papers:

- S. Ghanem, C. Conly, and V. Athitsos, "A survey on sign language recognition using smartphones," in Proceedings of the 10th ACM International Conference on PErvasive Technologies Related to Assistive Environments (PETRA). ACM, 2017, pp. 171-176.
- S. Ghanem, A. Imran, and V. Athitsos, "Analysis of hand segmentation on challenging hand-over-face scenario," in Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments (PETRA). ACM, 2019, pp. 236–242.
- S. Ghanem, A. Dillhoff, A. Imran, and V. Athitsos, "Hand over face segmentation using MPSPNet," in Proceedings of the 13th ACM International Confer-

ence on PErvasive Technologies Related to Assistive Environments (PETRA). ACM, 2020, pp. 257-264.

CHAPTER 2

A Survey on Sign Language Recognition Using Smartphones

2.1 Introduction

According to a report from Gallaudet University, which is a prominent educational institution that serves people who are deaf or are hard of hearing, there are approximately 38 million deaf individuals in the United States [5]. Many of those individuals use a sign language, typically American Sign Language (ASL), as a primary or secondary form of communication. Sign languages (SLs) are necessarily visual in nature. For sign language users, communicating with hearing people can be a challenge. Similarly, important information technology and social connectivity tools are not available to sign language users, unless the users are willing to access such tools using a spoken and written language, such as English, with which they may not be comfortable. Technological innovations in automated sign recognition have the potential to help sign language users overcome such obstacles, by facilitating both communication with hearing people, and human-computer interaction.

Mobile computing has entered a new era where mobile phones are powerful enough to be used in such advanced applications as gesture and sign language recognition. Many of the newly designed smartphones are equipped with multi-core processors, a high-quality GPU, and a high-resolution camera that can reach 12MP and more. These high-tech features allow the devices to execute computationally intensive tasks in less amount of time. In the last decade, many applications of computer vision have been limited to desktops, and now with the availability of advanced processor-

equipped smartphones, computer vision is primed to experience a transformation to provide new experiences via mobile devices.

Research has shown that ASL has four basic manual components: finger configuration of the hands, movement of the hands, orientation of the hands and the location of the hands with respect to the body [1]. Any automated sign recognition system needs two main procedures: the detection of the features and the classification of the input data. With mobile phones, the detection process can be affected by the movement of the phone, which causes extraneous motion around the signer. Some techniques use a sensor-based technology which tracks the gestures via hand movement using embedded sensors. Other techniques utilize vision-based approaches to process images of the captured gesture. Also, several researchers suggest using a client-server architecture to speed up processing time.

This literature review covers existing sign language recognition systems designed to run on smartphones. The lack of a clear overview in this area is the primary motivation to present this work. This survey presents several existing methods and groups them in different categories. The methods are discussed with a focus on the feature detection and classification algorithms.

The rest of the paper is organized as follows. Section 2.2 discusses the datasets used in this area. Section 2.3 describes existing approaches for sign language recognition in portable devices, including sensor-based and vision-based approaches. Finally, conclusions and possible future directions of the technology are discussed in Section 2.4.

2.2 Sign Datasets

In general, there are two types of signs: dynamic and static. Dynamic signs exhibit motion, whereas static signs are characterized by a specific static posture. We

Figure 2.1: ASL signs representing numbers 0-9 and letters of the English alphabet.

did not find any dataset that was designed exclusively for sign language application in portable devices. Some researchers use a static set of gestures, capturing signs for letters of the English alphabet and numbers 0-9, e.g., [6]. Figure 2.1 depicts American Sign Language signs representing numbers and letters. In many implemented methods, a customized dynamic dataset is utilized, e.g., [7]. It is difficult to handle the available datasets that were designed for personal computers due to the limited storage capacity of mobile phones.

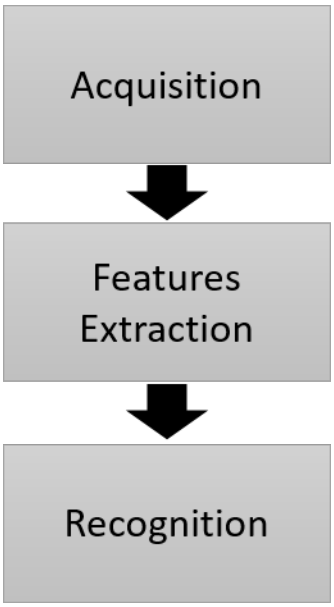


Figure 2.2: Basic System Architecture.

2.3 Sign Language Recognition Using Smartphones

In sign language recognition, the motion and posture of the human hand can observed via different approaches. In the sensor-based approach, the movement of the hand is tracked via sensors attached to wireless gloves or sensors embedded in smartphones, and appropriate techniques are used to process the responses from the sensors. In the vision-based approach, the gestures are observed via a mobile camera, and multiple processing steps are applied to recognize the signs that appear in the video stream.

Any sign recognition system contains three major steps; see Figure 2.2 for an overview. First, the input data is acquired, for example via the phone camera or from some sensor. The next step is to extract the features from the input data. Finally, the sign is classified using some appropriate algorithm that is compatible with the extracted features. For each method we examine, we take a close look in how that method approaches the problems of feature extraction and recognition/classification.

Table 2.1: A Comparison of Available Sensors Based Systems

System	Sensors	Classification Method	Gesture Type	Processing	Voc. Size	Dependency
Kau 2015 [7]	Gloves	Template Matching	Dynamic	Local	5	user-independent
Preetham 2013 [8]	Gloves	Minimum Mean Square Error Algorithm	Static	Local	-	-
Seymour 2015 [6]	Gloves	SVM	Static	Local	31	user-independent
Choe 2010 [9]	Phone Internal Sensors	DTW	Dynamic	Local	20	user-independent
Gupta 2016 [10]	Phone Internal Sensors	DTW	Dynamic	Local	6	user-independent
Joselli 2009 [11]	Phone Internal Sensors	HMM+forward -backward algorithm	Dynamic	Local	10	user-independent
Niezen 2008 [12]	Phone Internal Sensors	DTW	Dynamic	Local	8	user-independent
Wang 2012 [13]	Phone Internal Sensors	own statistical method	Dynamic	Local	21	user-dependent

2.3.1 Sensor-Based Approach

The usage of sensors simplifies the detection process and makes it faster. At the same time, sensor-based systems can be expensive and cumbersome to use, and these factors discourage adoption by a large number of users. Table 2.1 demonstrates a comparison between existing sensor-based models that use the phone as a platform. Sensor-based approaches can be broadly categorized based on whether they use external sensors, such as gloves, or internal sensors built into the smartphone. The following two subsections discuss these two categories.

2.3.1.1 Using Gloves

Glove-based approaches have been implemented using sensors that track hand gestures. Multiple sensors embedded in the gloves are used to track the fingers, palm and their location and motion. Such an approach provides coordinates of the palm and fingers for further processing. These devices may be connected wirelessly via Bluetooth.

The detection of hand parameters in this approach relies on a customized glove [7, 8] that contains ten flex sensors to track the posture of each finger. Moreover, a

G-sensor is used to monitor the orientation of the hand. Hand motion is detected by using a gyroscope sensor that calculates angles of the hands in space. These sensors continuously trace the signal to get hand data. The data are transferred wirelessly to the mobile device. From the gathered data, the state of the hand is estimated. This state can be decomposed into four independent components: hand posture, position, orientation, and motion.

The recognition methods vary by the available input data and the dataset used. Template matching was used in [7] as a classification method using five dynamic sign classes. In [6], a comparison is made between SVM and neural network methods using two different activation functions: log-sigmoid and symmetric Elliott functions. The experiment was done using static hand gestures, representing letters and numbers. In the results, SVM produced better accuracy, but it required 16 times more time for classification, compared to Log-sigmoid neural network and symmetric Elliott neural network. The advantage of this method was memory usage: only 4 MB of memory were required, which makes this method usable even with low-end smartphones.

2.3.1.2 Smartphone Internal Sensors Approach

Recently, new smartphones have been embedded with sensors that help to detect the posture and motion of the device. Numerous researchers utilize this feature to create gesture recognition models. The main issue with this approach is the limitation of signs details provided by the sensors.

Gestures recognized using such sensors can be decomposed into sequences of two simpler gesture types [13]. Turn gestures correspond to a change in the 3D orientation of the device. An example is rotating the device from the face up to face down position. Translation gestures correspond to the phone moving in 3D space. Moving the phone up and down is an example of such a gesture. Segmentation of

the motion is performed to detect the start and end point of the movement. Since the accelerometer continuously reads data of the three axis point in space, a vector containing the sum of derivatives of the current axis with the previous axis can be used to detect motion, as done in [11, 12, 13, 9]. To speed up the calculation time, **Gupta 2016** [10] change mean floating values to integer values by using a probability density function.

One of the better-known classification methods is the Dynamic Time Warping (DTW) algorithm, which is applied to measure the cost of a selected gesture compared with training data [14, 15]. One of the main advantages of this algorithm is that it does not need large amounts of training data, as it can be used even when only a single training example per class is available. DTW is used by [12, 10, 9] to achieve high accuracy, under the assumption that the start and end times for every sign are known. **Joselli 2009** [11] adapted foreword-backward algorithm to classify dynamic input signs using Hidden Markov Models (HMM), and using a database containing ten classes with a total of 400 samples. **Wang 2012** [13] process the data from the sensors to develop a sinusoid-like curve that can be used to extract the pattern of the movement. The axis of the largest variance between peak and valley is the movement direction.

2.3.2 Vision Based Approach

In recent years, the availability and simplicity of smartphones has encouraged researchers to utilize them in vision-based sign language recognition applications. The vision-based approach uses the phone camera to capture the image or the video of the hand performing signs. These frames are further processed to recognize the signs, so as to produce text or speech output. Vision-based approaches risk producing relatively low accuracy compared to sensor-based approaches, due to multiple chal-

Table 2.2: A Comparison of Available Vision Based Systems

System	Features Extraction	Classification Method	Processing	Voc. Size	Dependency
Elleuch 2015 [17]	Skin detection HSV, convexity defects	SVM	Local	5	user-independent
Gandhi 2015 [18]	Background subtraction	Template matching	Local	-	-
Hakkun 2015 [19]	Viola-Jones Haar Filters	KNN	Local	8	user-dependent
Hays 2013 [20]	Skin detection YCrCb, canny edge	SVM	Local, Client-Server	32	user-independent
Jin 2016 [21]	Skin detection RGB, canny edge, SURF	SVM	Local	16	user-dependent
Joshi 2015 [22]	PCA	SVM	Local	5	user-independent
Kamat 2016 [23]	Skin detection RGB	Template Matching	Local	4	user-dependent
Lahiani 2015 [24]	Skin detection RGB, convexity defects	SVM	Local	10	user-dependent
Prasuhn 2014 [25]	Skin detection HUV, HOG	Brute-force Matching	Client-Server	26	user-dependent
Raheja 2015 [26]	Sobel Edge Filter, PCA	Template Matching	Client-Server	10	user-dependent
Rao 2016 [16]	Gaussian and Sobel Edge Filter + PCA	MDC	Local	18	user-independent
Saxena1 2014 [27]	Sobel Edge Filter	Backpropagation Algorithm	Client-Server	5	user-dependent
Saxena2 2014 [28]	Skin detection RGB, PCA	Template Matching	Client-Server	10	user-dependent
Warrier 2016 [29]	Skin detection RGB	Geometric Matching	Client-Server	11	user-dependent

lenges in image processing, like light variations, dependency on the skin color of the user, complex backgrounds in the image, etc. Table 2.2 shows a comparison between currently existing vision based methods. It is important to note that all approaches listed in this table use static signs, except **Rao 2016** [16] which includes dynamic signs.

Extracting accurate hand features is a major challenge for the vision-based approach. Extraction is affected by many factors, such as lighting condition and background noise. The more accurate the detection and extraction is, the better the recognition results become. Orientation and position of the hand can be detected in different ways, for example using skin detection or Viola-Jones cascades of boosted rectangle filters [30]. Detecting the position and orientation of the hand at each frame accurately also allows us to detect the motion of the hand for dynamic signs.

Skin segmentation algorithms, which often depend on specifying thresholds [31], are widely used in Computer Vision applications. The researchers either specify skin thresholds manually or automatically by taking a skin color sample before the experiment. Several available models use RGB color space, e.g., [21, 23, 24, 28, 29]. To solve brightness and lighting problems, [20] use YCrCb color space, [17] employ HSV color space, and [25] benefit from HUV color space.

The Viola-Jones detection method [30], which uses cascades of boosted rectangle filters, is a well-known method, that is commonly used for detecting hands. Some researchers [19, 17] implement the Viola-Jones method on portable platforms, as Viola-Jones is relatively easy to implement and has low hardware requirements. Another alternative, used by [22, 26, 16, 28], is Principal Component Analysis (PCA).

Additional hand details are also extracted by various methods. Examples of such details are the number of open fingers (measured by finding contours), finding the palm area (by finding the largest circle that fits in the hand region), detecting the convex hull, and getting convexity defects [17, 20, 24]. Canny edge detection [32] can also be used to identify the hand area [21]. Likewise, a Sobel Edge filter, which measures the changes in value in the highest moving direction, has been used [26, 16, 27]. **Prasuhn 2014** [25] apply a Histogram of Orientation Gradients (HOG) method, which is sensitive to the angle of the object, to extract the features from the input image. Another method, used by [18], is background subtraction using a motion detection method. In **Jin 2016** [21], Speeded-Up Robust Features (SURF) is used as an extra feature to improve accuracy.

Once the features describing a sign have been extracted, there are numerous recognition procedures that can be applied. Support Vector Machines (SVM) define decision boundaries between classes, which are linear in some transformed feature space, but can be highly nonlinear in the original feature space [33]. Several papers use

SVMs, e.g., [17, 20, 21, 22, 24]. **Hakkun 2015** [19] use K-Nearest Neighbor (KNN) for classification. Another simple technique for classification is template matching, used by [18, 23, 29, 26, 22]. The Backpropagation algorithm [34] can lead to very efficient classification timewise, but it needs more training data to minimize error rate. Backpropagation is used by [27] as the recognition method. In **Rao 2016** [16], because the speed of processing in portable devices is a major factor, a minimum distance classifier (MDC) was chosen as a classification method. The experiments use sentences of signs as training and test data.

Some systems assume that the only visible object in the captured image is the hand [19, 17], while the more advanced models manage to capture both hands and face. One way to remove the confusion between a face and hand area is to subtract or isolate the face, so that the detection of hand details will be more precise [17]. Another issue that can be considered is hand angle and hand distance from the mobile device. In tests conducted in [19], optimal results were achieved with no more than 50 cm distance between the hand and the camera, and the hand being in the upright state.

Due to slow processing time in some models, a client-server framework is used. In such a framework, the phone is connected to a regular computer via wireless network. Such an approach was implemented in [25, 26, 27, 28, 29]. A cloud service can be used to execute part of recognition operations, as done in [20]. Moreover, **Elleuch 2015** [17] implement a multithreading technique by running face subtraction and hand pre-processing at the same time, thus decreasing the processing time by half.

2.4 Conclusion

In this paper, we have provided a survey of existing techniques for sign language recognition in smartphones. We discussed sensor-based approaches, which track hand

motion and/or posture using hardware-based trackers installed in a glove or inside a smartphone. We also discussed vision-based approaches, which use the phone camera for observing the hand. In discussing both types of approaches, we focused on the detection and feature extraction module as well as the classification module of each approach.

Regarding vision-based methods, significant challenges remain to be overcome by future research, regarding accuracy of hand detection and articulated hand pose estimation, as well as classification accuracy. Most existing vision-based methods only recognize static gestures, and we expect new methods to be proposed for handling dynamic gestures. Similarly, existing methods typically cover no more than a few tens of signs, and there is significant room for improvement until methods can cover the several thousands of signs that users of a sign language employ in their daily usage. Extending vision-based recognition systems to cover dynamic gestures and thousands of signs may strain the hardware capabilities of smartphones. While smartphone hardware specs are expected to continue to improve rapidly, cloud processing could push the boundaries further ahead by alleviating the hardware requirements on the mobile device. However, maintaining interactivity and low latency while using cloud processing can also be challenging, and these are also issues that we expect future research to focus on.

CHAPTER 3

Hand Over Face Dataset

3.1 Introduction

In recent years, computer vision has been playing an increasingly important role in assisting computers extract and analyze a variety of information from images. One of the challenging problems in computer vision is hand segmentation, particularly when hands are placed in front of a face. Hand segmentation can be defined as the problem of determining, given a picture containing a hand, for each pixel in the image whether it is part of the hand or not. Hand segmentation is a useful operation for numerous applications such as sign language recognition, action/ activity recognition, and recognition of objects that hands interact with. In the past, probabilistic methods such as Conditional Random Fields (CRF) [35] were used in image segmentation problem. Deep learning architectures have dominated the research in the field in recent years. A key advantage of deep neural networks is the ability to automatically extract expressive features from a dataset. In egocentric application, where the hand-over-face problem does not appear, there are numerous research works about hand segmentation using CNN algorithms[36, 37, 38]. Hand over face segmentation research did not capture enough attention, although a few methods have been proposed [39, 40]. Even though some of the methods used for egocentric applications can be utilized to solve hand segmentation in a normal scene, the similarity of skin color between hand(s) and face make it a challenging problem that needs further consideration.

Work on various computer vision topics typically benefits from the creation of challenging public datasets, that can be used to benchmark existing methods and

to highlight needs for improved performance. Two factors are important to consider in any hand segmentation dataset: pixel-wise ground truth data, and the quantity of annotated frames, which is an important attribute for CNN-based algorithms. For hand segmentation, there are many egocentric datasets, but there is a lack for datasets where hand(s) appear in front of or near to the face. To the best of our knowledge, HOF [37] is the only available dataset that can be utilized for hand-over-face segmentation model, but the amount of annotated frames is small.

The shortage of a dataset for this research area is the major motivation for the work described in this paper. In summary, our first contribution is to enrich the field with a challenging dataset (VLM-HandOverFace) to address hand over face problem. Secondly, we manually annotate hand(s) at a pixel level over more than 4300 video frames taken in normal environments condition. Finally, the performance of our new dataset is evaluated using two state-of-the-art methods.

The rest of the article is arranged as follows. Section 3.2 discusses the related work including used methods and related datasets. Section 3.3 describes the new dataset (VLM-HandOverFace) in details and in section 3.4 we present our analysis of the new dataset using recent state-of-the-art methods. Conclusions and future works are discussed in Section 3.5.

3.2 Related Work

Recently, several methods have been proposed to solve the hand segmentation problem. To discuss the previous works in hand segmentation, we review it into two sections: related methods and related datasets.

3.2.1 Related Methods

Hand segmentation can be considered as a semantic segmentation problem, where the goal is to assign a single label from a target set of class labels to each pixel. Semantic segmentation is important for understanding the content of images and finding target objects.

In recent years, very deep residual networks have been showing promising performance for semantic segmentation. RefineNet [41] exploits activation maps at different levels to produce high-resolution semantic maps. RefineNet proposes a multi-path refinement network that feeds all available input data towards the down-sampling procedure to enable the prediction of a high-resolution result by applying long-range residual joints. RefineNet has demonstrated the usefulness of models based on encoder-decoder architecture on several semantic segmentation benchmarks.

SegNet [42] proposes a simple encoder-decoder based architecture for semantic segmentation. It was originally designed for road scene understanding, but SegNet can be used for any pixel-wise semantic segmentation task. It consists of an encoder network which is a standard CNN like VGG-16 [43] and a corresponding decoder network which is used for up-sampling the output from the encoder [44, 45, 46, 47]. In the end, there is a pixel-wise classification layer. It applies unpooling operations to un-sample the low-resolution features and learns deconvolutional layers to improve the up-sampling process.

U-Net [48] is a very popular bio-medical imaging segmentation method and it is generally useful for the semantic segmentation problem. The architecture of U-Net involves a contraction path and expansive path. The contraction path contains 4 blocks of (a) two 3x3 convolution layers, (b) ReLU layer, and (c) 2x2 max pooling layer. Then, there is an intermediate downsampling step that contains 2 simple convolution layers. The expansive path starts with upsampling of the features using 4

blocks and each block contains (a) 2x2 de-convolutional layer, (b) two 3x3 convolution layers, and (c) ReLU layer. In the end, one convolution layer is used to map the feature vector to output classes.

The above methods mostly perform one-shot segmentation. In other words, the source image is passed only once through the network, which directly outputs the segmentation map.

3.2.2 Related Datasets

Several datasets have been proposed for the hand segmentation task. The list of datasets is shown in table 3.1 with brief information about each one. The only dataset that addresses the hand-over-face problem is HOFdataset [37], which is created from random images collected from the internet. The images contain hands in front of the face or near the face. The people in the images are from different ethnic backgrounds, colors, gender, and ages. The size of the dataset is 300 frames, which might not be enough to train deep neural network. On the other hand, our dataset contains more than 4300 pixel-level annotated frames, which are extracted from videos that represent random hand movements in front of or near the face. Also, our dataset includes depth frames that may help in future research.

In the next paragraphs we discuss some datasets that are to an extent related to the dataset we propose in this paper. The key differences and advantages of the proposed dataset will be clarified in the next section, where we provide a detailed description of our dataset.

The NYU Hand pose dataset [40] was created using 3 Kinect cameras (front and two sides views), resulting in a total number of 81000 frames. To simplify the annotation step, the hands of the performer were painted with red paint. Training data were recorded from a single subject. That same user and another one are the

only two subjects in the test data. Each frame may contain one hand, two hands, or no hand. Hand locations are diverse and around 10% are in front of the face or near the face. The main advantages of our new dataset compared to the NYU dataset for the purposes of hand-over-face segmentation are, first, that in our dataset almost all frames show the hand over the face, and, second, that our dataset contains videos from 42 subjects, and thus allows user-independent evaluations, where the subjects in the test set are different from the subjects in the training set.

The Caltech Occluded Faces in the Wild (COFW) dataset [49] is mainly for detection of face parts in cases where the face is occluded by hands, objects, and other faces. The images were collected from the internet and face landmarks were annotated manually. This dataset does not contain annotation of hand locations at the pixel level, so it does not provide the required ground truth information for evaluation of hand segmentation methods.

The ICVL Hand posture dataset [50] and ICVL Big Hand dataset [51] are used for hand pose estimation. It contains labels for the 21 joints of the hand, and no pixel-level segmentation ground truth.

The Video Corpus HandOverFace dataset [52] was recorded for 6 participant (3 male, 3 female). It contains 138 videos with about 450 frames each. Each frame was annotated by dividing it into 9 regions, and each region was labeled with 1 if the hand was present in that region, or labeled with 0 otherwise. The dataset contains facial expressions and head motion gestures including hand(s) in front of the face or near the face. This annotation method does not provide pixel-level information, which is important for evaluating segmentation accuracy.

The authors of Hand2Face dataset [53] created the dataset following these steps: first, selecting an existing face dataset (the LFPW dataset [56] was chosen). Second, extracting hand and other objects such as glasses, hat, scarves, etc. from a group of

Table 3.1: A Comparison of Available Related Datasets

	Dataset Name	Availability	Number of frames	Hand over face	Egocentric	Pixel level	Depth Info.	User Independent
1	VLM-HandOverFace	Public	4384	Yes	No	Yes	Yes	Yes
2	HandOverFace (HOF) [37]	Public	300	Yes	No	Yes	No	Yes
3	NYU Hand pose dataset [40]	Public	81000	Yes, 10% of the dataset	No	Yes	Yes	No
4	Caltech Occluded Faces in the Wild (COFW) [49]	Public	1852	Yes but also other objects	No	No	No	Yes
5	ICVL Hand Posture Dataset [50]	Public	180K	Some	Yes	No	Yes	Yes
6	Video Corpus HandOverFace [52]	Public	138 Videos * 450 Frames	Yes	No	No	No	Yes
7	Hand2Face [53]	Private	9912	Yes	No	Yes	No	Yes
8	ICVL Big Hand Dataset [51]	Public	2.2M	Some	Some	No	Yes	Yes
9	MSRGesture3D [54]	Public	12 gesture by 10 participant	Some	No	No	Yes	Yes
10	LSF Dicta-Sign corpus [39]	Private	50	Yes	No	Yes	No	Yes
11	Cam3D corpus [55]	Public	108 videos	25%	No	No	Yes	Yes
12	EgoHands [36]	Public	4800	No	Yes	Yes	No	Yes
13	EgoYouTubeHands (EYTH) [37]	Public	1290	No	Yes	Yes	No	Yes
14	GTEA [38]	Public	663	No	Yes	Yes	No	Yes

images taken from the internet. Third, generating new images that include a face and one of the occlusions. The choice of suitable occlusion was measured by many factors including color illumination of face and occluder, quality of both face and occluder, region to be inserted in, the scale of the occluder, and pose of the face. An advantage of our dataset compared to Hand2Face is that our dataset consists of real images, as opposed to combinations of unrelated real images, which is the case in Hand2Face.

The MSRGesture3D dataset [54] is a small dataset with only 12 classes. The number of frames is not reported. The depth information is available. The authors of LSF Dicta-Sign corpus dataset [39] created a small subset of this dataset where the hand(s) are in front of face with only 50 manually annotated frames. The Cam3D corpus dataset [55] made for emotion description application. They found out that 25% of the data is hand over face. It was done by 7 participants and 12 emotional

expressions. Our new dataset is 6 times bigger in the number of participants and focused only on hand over face problem.

EgoHands [36] is an egocentric dataset that focuses on playing activity. The recording region of interest is around the object of the game, which is not near to the face. EgoYouTubeHands (EYTH) [37] is an egocentric dataset created from YouTube videos that record real-life activities. This dataset focuses on hand activities and the hands do not appear in front of a face. The GTEA [38] is well known egocentric dataset recorded in a static background, doing 7 daily activity, performed by 4 people. It is mainly used for activity recognition.

3.3 The New Hand Over Face dataset

In this work, we introduce a new dataset, that we call the VLM-HandOverFace dataset, that targets the hand segmentation problem where hands are in front of or near the face. Our new dataset was recorded inside a lab. A Kinect v2 camera was used for, and RGB, depth, and skeleton streams were recorded. This camera was attached to a stand in front of the subject. Also, a Leap Motion sensor was used to register hand(s) parts as an additional resource that can be used in future. This sensor was placed on a table to record any hand(s) movement in the range of it. SenseCap [57] was used as a tool for recording both sensors simultaneously. The frame size of RGB stream is 1920x1080 and the frame size of depth and label streams is 512x424. Moreover, the frame size of the leap motion right and left streams is 640x240. The Kinect skeleton information, that contains the position of 25 body joints in the 3D space, and Leap motion hand joints data, which include the 3D positions of all bones in each finger, are saved as text files. Figure 3.1 shows an example of recorded streams.

There are 42 participants with a variety of skin colors, gender, and races. In addition, all contributors are free to wear any accessories that they normally use such

(a) KINECT RGB Stream

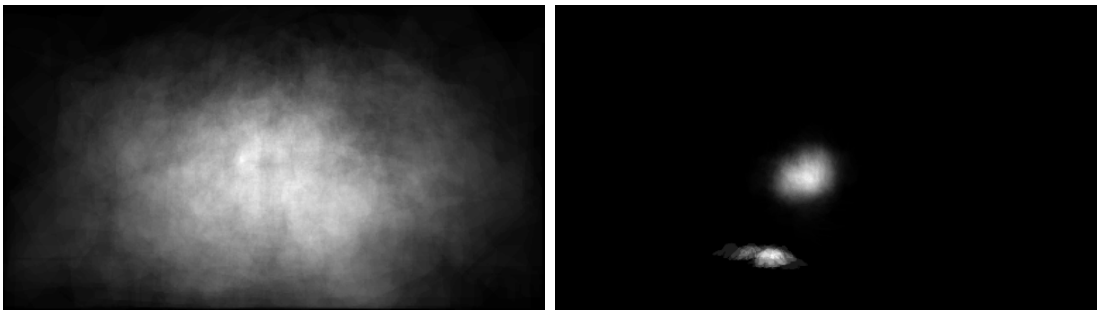
(b) KINECT Depth Stream

(c) KINECT Label Stream

(d) Leap Motion Left Stream

(e) Leap Motion Right Stream

Figure 3.1: example of recorded streams.



(a) HOF (b) VLM-HandOverFace

Figure 3.2: Heatmaps for HOF dataset and VLM-HandOverFace dataset

as eyeglasses, watch, ring, ...etc. For each volunteer, two videos were recorded. First, with a white wall background. Second, with complex lab background. In each video, the subject mimics a video of random hand movements and shapes where hand(s) are in front of or close to the face. Each video contains these parts: (1) some hand shapes and movement with one hand in front of the face but away from it. (2) numerous hand shapes and movements with one hand touching the skin of the face. (3) a collection of hand shapes and movement with two hands in front of the face but away from the face. (4) Two hands touching the face and performing some actions. Furthermore, to make it a more challenging dataset, we include some hand shapes that cover the whole face. The dataset includes many cases with hand(s) touching the side of the face, which makes it a hard task to distinguish between face skin and hand skin. Moreover, our dataset includes occlusion between right and left hand. The total number of frames for all videos is (317764) frames. Figure 3.2 shows the heatmap of hands locations in our VLM-HandOverFace dataset and HOF dataset. Clearly, in our dataset, hands locations are within the center of frames where the face are normally located. The small white area at the bottom represent the nondominant hand in the case of one hand only in front of or near to the face.

(a) RGB frame (b) Hands (c) Right/Left Hand

Figure 3.3: An example of dataset masks.

As a part of our contribution, we annotate a total of (4384) frames randomly extracted from the dataset videos. There are several annotation tools that can be used for creating a ground-truth data. To the best of our knowledge, Ratsnake [58] is the suitable pixel-wise annotation tool because it is easy to use, open source, and fulfill our needs. For each frame, a binary hand-background mask was created indicating if a pixel belongs to a hand or not. In addition, a three-label mask was created for each frame, where each pixel was annotated as belonging to the left hand, to the right hand, or to the background. For all these masks, "background" simply means "not hand". Figure 3.3 demonstrate examples of hand(s) masks.

3.4 Analysis and Experiments

Since neural network algorithms run much faster on an advanced graphical processor, we use a computer that contains a NVIDIA GeForce GTX 1080 GPU

that can perform matrix operation faster. To analyze our new dataset, we made two experiments. First, segmentation of hand(s) from an input image. Second, pixel level segmentation of right and left hands from an input picture. Both experiments are performed in a user-independent fashion, where humans appearing in training data do not appear in test data. In the two experiments, we use RefineNet and SegNet as segmentation methods. Before applying any method, all images are resized to 480x272. To evaluate the segmentation result, we report three metrics: pixel-wise mean Intersection Over Union (mIOU), mean Recall (mRec), and mean Precision (mPrec).

3.4.1 Hand-No Hand Experiment

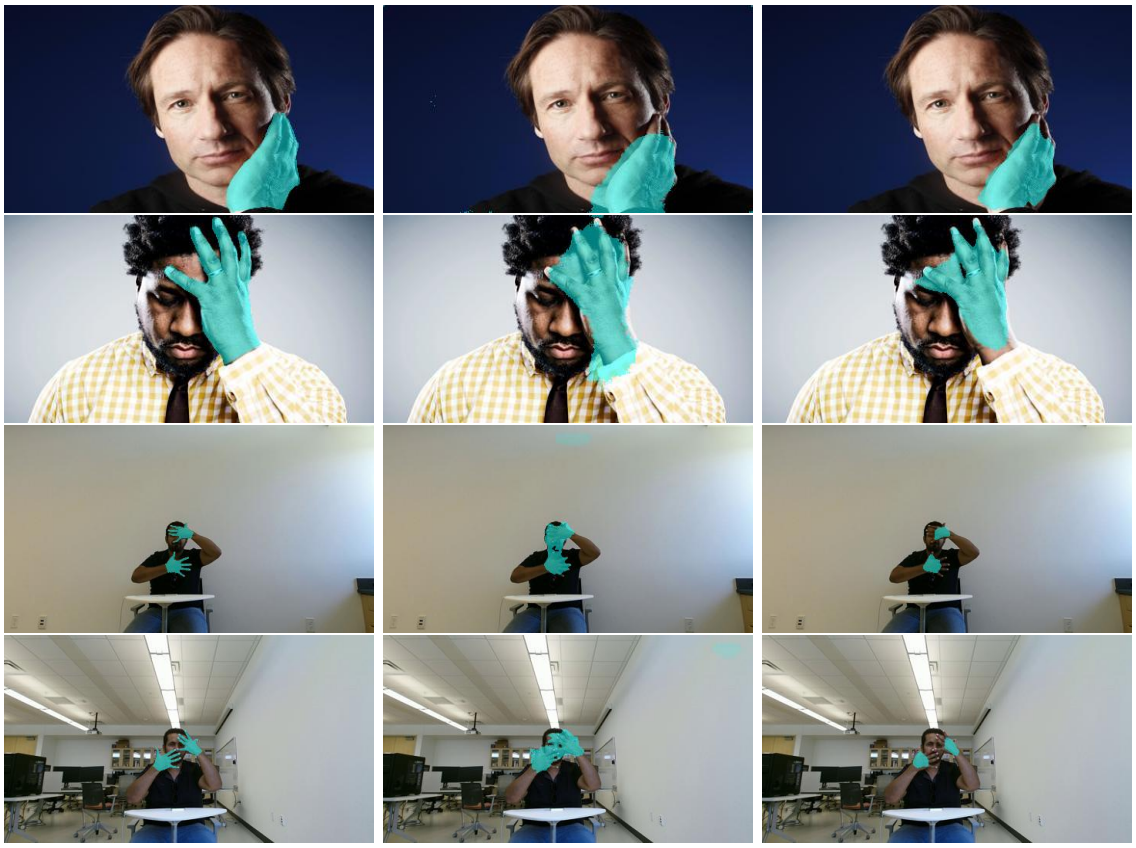
The Hand-No Hand experiment aims to predict hand pixels in an input image. We show results obtained both on our dataset and on the HOF dataset of Urooj and Borji[37]. In HOF, the data is split to 80% as training data and 20% as testing data. Also, they used 5e-5 as a learning rate. In our experiment, we use the same data split ratio and the same learning rate. We perform RefineNet and SegNet on both datasets, our new VLM-HandOverFace dataset, and the HOF dataset. The training for each method stopped in epoch 200. Table 3.2 shows the segmentation results for both experiments. From results metrics, RefineNet is 19% mIOU and 27% mPrecision better than SegNet. Indeed, the ratio still low, which increase the challenges to solve this segmentation problem using our new dataset. Figure 3.4 demonstrates examples of prediction results.

3.4.2 Right Hand-Left Hand-No Hand Experiment

Another experiment was done for right and left hands segmentation. Our new dataset equipped with the labeling of right and left hands. Many applications can

Table 3.2: Hand-Background experiment using RefineNet and SegNet on VLM-HandOverFace and HandOverFace2018 Datasets

	VLM-HandOverFace			**HandOverFace2018**		
	mIOU	**mRec**	**mPrec**	**mIOU**	**mRec**	**mPrec**
RefineNet	0.7951	0.8993	0.8338	0.7676	0.8832	0.8559
SegNet	0.4398	0.9790	0.4442	0.4902	0.7248	0.6076

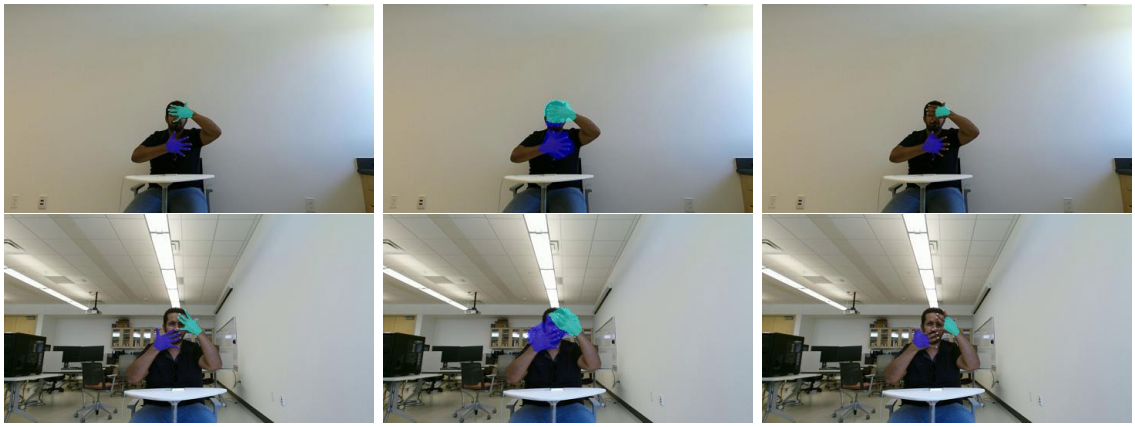


(a) Ground Truth (b) SegNet (c) RefineNet

Figure 3.4: Examples of predicted images after performing semantic segmentation methods on HandOverFace2018 (first two rows) and VLM-HandOverFace datasets.

Table 3.3: Right hand-Left Hand-Background experiment using RefineNet and SegNet

	mIOU	**mRec**	**mPrec**
RefineNet	0.6984	0.8100	0.8491
SegNet	0.4318	0.9813	0.4352



(a) Ground Truth (b) SegNet (c) RefineNet

Figure 3.5: Examples of predicted images after performing semantic segmentation methods on VLM-HandOverFace datasets.

benefit from this type of experiment, including sign language recognition. The same data split was used, 80% training and 20% for testing, as in the first experiment. All models were trained until epoch 200. SegNet and RefineNet were applied to our VLM-HandOverFace dataset. Table 3.3 displays the result matrices of the experiment. Again, RefineNet outperformed SegNet by more than 14% mIOU and 34 % mPrecision. However, mRecall in RefineNet is 30% less than SegNet. Figure 3.5 shows examples of right and left hand segmentation result.

3.5 Conclusions

In this paper, we have proposed a new annotated video dataset for segmenting hands appearing in front of faces. We provided a review of several existing datasets

related to hand segmentation research, to motivate the need for the new dataset. Our VLM-HandOverFace dataset includes more than 4300 annotated frames with RGB, depth, infrared streams, captured by a Kinect camera. Moreover, the provided dataset includes hand(s) finger coordinates recorded by Leap Motion sensor. Performance on our new dataset was measured for two state-of-the-art methods: SegNet and RefineNet, which are used for semantic segmentation. In our experiments, the RefineNet algorithm achieved better results than the SegNet algorithm. However, the attained accuracy is far below from the human-eyes regular ability distinguishing different objects, and thus these results illustrate that there is wide room for improvement of the state of the art for hand-over-face segmentation.

Although recent research associated with hand segmentation primarily focused on egocentric applications, the hand-over-face segmentation problem remains challenging. Directions that should be covered in related future research include improving the accuracy of hand segmentation, as well as handling occlusions, hand(s) size, and the lighting condition. The availability of abundant and in-depth information can help overcome occlusion problem in future methods. Also, we believe research can benefit from Leap motion sensor to accurately recognize hand(s) shapes. Finally, distinguishing between right and left hand is an important task for applications such as sign language recognition systems. Also, analyzing the appearance of many hands appearing together in the scene and possibly overlapping is an interesting challenge in this field of research.

CHAPTER 4

Hand Over Face Segmentation using MPSPNet

4.1 Introduction

Hand segmentation is an essential part of many computer vision applications, such as human-computer interaction, gesture recognition, activity recognition, and sign language recognition. Accurate hand segmentation is challenging, due to the high variation of lighting and skin color as well as complex backgrounds. An additional, yet relatively unexplored, challenge is the case of hands overlapping the face. The similarity of the skin color results in inaccurate segmentation, especially with RGB-based segmentation methods.

In this work, we address the problem of segmenting the hands, with a special emphasis on examples where a hand overlaps with the face. Hand segmentation is commonly formulated as classifying each pixel in an image as belonging to a hand or not. Methods for solving hand segmentation can generally be categorized into two tracks: (1) probabilistic approaches and (2) deep learning approaches. Probabilistic methods, such as Conditional Random Fields [35], were dominant in earlier work. Recent approaches utilize Convolutional Neural Networks (CNNs) for hand segmentation in egocentric scenes [36, 38, 37]. Despite their contributions on challenging standard benchmarks, few approaches have focused on the hand over face setting [39, 40].

RefineNet [41] and PSPNet [59] are two well-known semantic segmentation architectures. RefineNet can extract the core and in-depth features of the object of interest while PSPNet is a lightweight framework that investigates global and local

context features. Although RefineNet exhibits state-of-the-art performance on PASCAL VOC 2012, it has a high computational cost and struggles with determining edge features. The pyramid pooling technique, introduced in PSPNet, successfully detects the edges of the target object. However, PSPNet falls short with noisy samples. In this paper we introduce a novel approach, that combines the benefits of the above-mentioned state-of-the-art frameworks. This new model, which we call the Multi-level Pyramid Scene Parsing Network (MPSPNet), relies on two main ideas: (1) the cascading of multi-level tuned features and (2) pyramid-pooling encapsulated blocks, introduced in PSPNet [59].

Our work is motivated by the challenging scenario in which the hands overlap the face. We empirically evaluate MPSPNet on two recently published datasets for hands-over-face segmentation [3, 37]. These evaluations test the network under two settings. In the first setting, each image pixel is labeled as hand or non-hand. In the second setting, there is a separate label for the left hand and a separate label for the right hand. Also, we evaluate our MPSPNet on two standard object segmentation datasets: PASCAL VOC [60] and NYUDv2[61].

The remainder of the paper is organized as follows. Section 4.2 surveys prior work related to hand segmentation. Section 4.3 reviews RefineNet and PSPNet, as both of them are strongly related to our work. Section 4.4 details our proposed architecture (MPSPNet). In section 4.5, we present our results on four semantic segmentation datasets including two challenging hand over face datasets: VLM-HandOverFace and HOF. Discussion and conclusion are in Section 4.6.

4.2 Related Work

In general, there are two broad categories of methods to handle the hand segmentation problem. The first category comprises probabilistic approaches, which can

be classified into four sub-categories; (1) relying on local appearance features, for example, skin color [62, 63, 64]. (2) based on global appearance features such as hand template matching [65, 66, 67]. (3) tracking the motion of the hand(s) [38, 68, 69]. (4) derived from the combination of skin color and edge features from the hand(s) and face when tracking hand motions [39]. In general, more work is needed in this area to address the broad possible variations in illumination and skin color. Many papers adopt depth information to segment the hand(s) [40, 70, 71, 72], but in many real-world scenarios (for example, translating sign language in YouTube videos) depth information is simply not attainable.

The second category of hand segmentation methods utilizes deep learning. Several well-known architectures that are based on convolutional neural networks (CNNs) have been proposed in the field of semantic segmentation. FCN [73] is a popular network, that first encodes and merges features from different stages, and then applies a deconvolutional operation to get the maps of the upsampled semantic features. The drawback of FCN is the long processing time, and the loss of some feature information during the transition within the network layers. In U-Net [48], upsampling and downsampling layers are combined via a skip-connection technique to concatenate features from the base and developed paths. Furthermore, this network requires more memory usage due to the entire feature map being transferred between encoders and the corresponding decoders.

SegNet [74] is a simple encoder-decoder architecture, which varies by the design of the decoder. More specifically, the decoder in SegNet consists of a group of upsampling and convolution layers followed by a softmax layer at the end to label each pixel in the output. The accuracy of SegNet tends to be lower than that of other existing approaches. AdapNet [75] adds a convolution layer before ResNet, which allows the architecture to learn high-resolution features more quickly. Also,

the Convoluted Mixture of Deep Experts (CMoDE) was introduced in AdapNet to fuse multiple modalities and spectra in order to learn deeper robust kernels. Bilateral Segmentation Network (BiSeNet) by Yu *et al.*[76], contains two parts: a spatial part to extract deep semantic information and a context part to produce a sufficient receptive field. DeepLabv3 [77] is a cascading atrous convolution that collects multi-scale context by utilizing multiple atrous rates. DeepLabv3+ [78] is an extended version of DeepLabv3 by adding a decoder module, where the feature is upsampled by four rather than 16 and concatenated with the corresponding low-level features from the network. Also, depth-wise separable convolution is applied to decrease computational complexity. RefineNet [41] is another encoder-decoder based architecture. It introduces a multi-path refinement network that loads all input data across the downsampling process to enable the prediction of high-resolution output by executing long-range residual joints. PSPNet [59] uses spatial pyramid pooling to collect global and local feature maps from four different bin sizes before upsampling and concatenating them to obtain the final prediction output.

The number of publicly available datasets for the hand over face problem is limited. The size and the variety of recorded samples are a vital attribute in any deep learning dataset. Also, pixel-wise annotation is the most crucial element in segmentation datasets. Ghanem *et al.* [3], provided a detailed discussion about hand segmentation datasets. Also, they created a VLM-HandOverFace dataset and made it available to the public for future use by the research community.

For our research project, the most interesting and relevant work was done by Urooj and Borji [37]. They fine-tune the RefineNet architecture to do segmentation of hand(s) in egocentric and hand over face applications. They adopted RefineNet-Res101, which pre-trained on Pascal-Person-Parts. Also, Urooj and Borji [37] introduced a small hand over face dataset that contains 300 frames.

4.3 Background

MPSPNet is inspired by two well-known hand segmentation networks, RefineNet and PSPNet. Additionally, our proposed method is adapted to the challenging hands-over-face scenario. Due to the strong relationship between our work and these two models, a brief overview of each one is highlighted as follows:

RefineNet [41] is based on an encoder-decoder architecture consisting of two main components. The first component is the RefineNet block which includes three units: residual convolution unit, multi-resolution fusion unit, and chained residual unit. The second is multi-path refinement where four different sizes of feature maps are downsized. In each path there is a RefineNet block which receives the input in the current path and the output of the RefineNet block in the previous path. In this way, all blocks are cascaded to predict high-resolution semantic maps.

PSPNet [59] is a promising lightweight model for pixel-wise segmentation. Its architecture can be summarized in three stages. (1) ResNet [79] is employed to extract visual features from the input image. (2) The visual features are passed to the pyramid pooling module which joins features within four different pyramid dimensions. The idea of applying the pyramid pooling procedure is to capture both local context features at different scales (using the receptive field of 1x1, 2x2, 3x3 and 6x6 respectively) and the global context features (the entire image as a receptive field). The generated features from all levels are upsampled to match the size of the input feature map and concatenated. (3) The last stage applies a convolutional layer to get the final prediction result. Despite the fact that the final feature map contains valuable semantic information, the object boundary information is still missing.

RefineNet and PSPNet use ResNet [79] as a feature extractor. Their feature networks are pre-trained on the Pascal Person-Part dataset. After pre-training, the

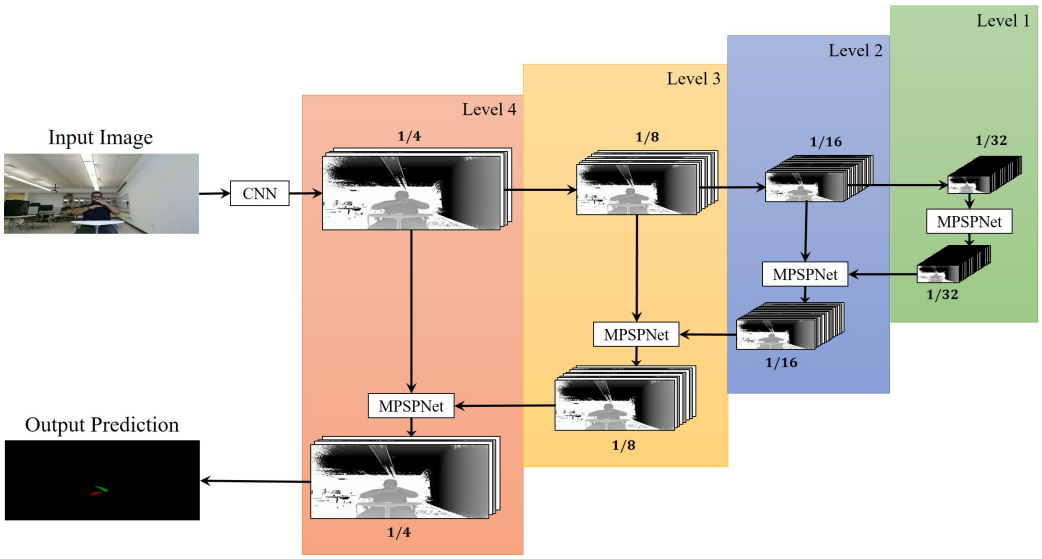


Figure 4.1: MPSPNet architecture.

entire hand segmentation network, including the pre-trained feature extractors, is fine-tuned in an end-to-end fashion.

4.4 Proposed Method

MPSPNet incorporates two basic concepts. The first is multi-level processing and fusion of different sizes of feature maps, discussed in Section 4.4.1. The second is the MPSPNet block that extracts global and local context attributes from each level, described in Section 4.4.2.

4.4.1 Multi-level MPSPNet

A multi-level processing hierarchy yields successful outcomes in the pixel-wise hand segmentation [41, 37]. Figure 4.1 shows the design of our proposed architecture. First, we use ResNet to generate four sets of feature maps of the original image scaled by 1/4, 1/8, 1/16, and 1/32. Each feature map is handled through a single level of processing using an MPSPNet block. The workflow of the network is starting from

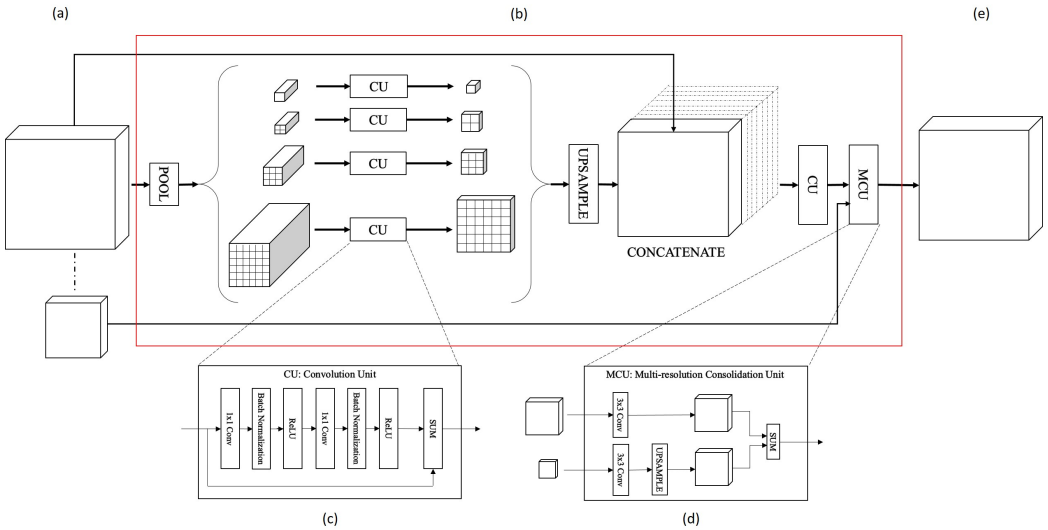


Figure 4.2: MPSPNet block and the details of each individual component. (a) is the input feature map. (b) the red box is the modified pyramid pooling module. (c) is the convolution unit. (d) is the multi-resolution consolidation unit. (e) is the feature map after processing.

the first level in the right, which examines the smallest size of feature map (1/32). The MPSPNet block in the first level receives only one feature map input and the output in this level are considered as the initial weights of the network. The remaining levels 2,3, and 4 are attached to feature maps 1/16, 1/8, and 1/4, respectively. Each MPSPNet block in those three levels acquires two inputs. The first came from the output of the MPSPNet block in the previous level, which is considered as a low-resolution feature map. And the second is the feature map in the current level which is rated as a high-resolution feature map. By applying this technique, the MPSPNet block will produce a high-resolution feature map by cultivating the two input maps. Finally, the resulting feature map from the last level is forwarded to a Softmax layer for prediction.

4.4.2 MPSPNet Block

The design for a single MPSPNet block is shown in Figure 4.2. Depending on which level the block occurs at, the input for the module is either one or two feature maps as described in section 4.4.1. Following the same procedure of the Pyramid Pooling Module in [59], we execute a max-pooling step to generate four different bin sizes of feature representations. Each bin passes through a Convolution Unit (CU), which is described in Section 4.4.2.1. The resulting feature maps are upsampled to match the original size of the input feature map. The four feature maps produced by the previous step are then concatenated with the original input feature map before being processed by an additional CU. Finally, the concatenated feature maps are used as input to a Multi-resolution Consolidation Unit (MCU), described in section 4.4.2.2. The final result from the block is a high-resolution feature map with the same dimensions of the high-resolution input.

4.4.2.1 Convolution Unit (CU)

The main goal of this unit is to preserve the quality of the global features in the input map. To do that, we apply two sets of the following three layers: a 1x1 convolution layer, a batch-normalization layer, and a ReLU activation function. In the end, the outcome map is added to the input map in the unit. The design of this unit is shown in figure 4.2 (c).

4.4.2.2 Multi-resolution Consolidation Unit (MCU)

The fusion of different sizes of feature plans is the target of the MCU. This unit receives two feature maps as inputs: (1) the high-resolution map that came from the pyramid pooling module, and (2) the low-resolution map from the previous level (if

available). Both are processed by 3x3 convolution layers. Then the low-resolution plan is upsampled to the size of the higher resolution map. Finally, a summation of the two feature maps is done to complete the consolidation procedure. Figure 4.2 (d) demonstrates the structure of MCU.

4.5 Experiments

To show the validity of our new proposed architecture, we use four public datasets to perform two types of segmentation experiments: object and hand(s) segmentation. In each experiment, we compare our network with several state-of-the-art architectures. Also, in hand(s) experiments, we make ablation studies for the two added units (CU and MCU).

To evaluate the quality of our segmentation, we provide three metrics. The first and most popular value is the **mean Intersection Over Union (mIOU)**, which presents the overlap between the prediction mask and the ground truth mask.The second reported measurement is the **mean Precision (mPrec)**, which represents the quality of object pixel detection with respect to the ground-truth label. The last value to report is the **mean Recall (mRecall)**, which illustrates the quantitative value of correct pixel prediction.

4.5.1 Object Segmentation

4.5.1.1 PASCAL VOC 2007

PASCAL VOC 2007 [60] is a popular segmentation dataset which comprises 20 object classes and a background. The pictures were taken from a variety of places with different lighting conditions, and each image includes a random number of objects. In

Table 4.1: Results on PASCAL VOC 2007 testing set.

Network	mIOU	mRecall	mPrec
AdapNet[75]	0.2365	0.3545	0.3732
BiSeNet[76]	0.3273	**0.4803**	0.4685
DeepLab-v3[77]	0.2594	0.3822	0.4212
DeepLab-v3Plus[78]	0.2870	0.4115	0.4604
RefineNet[41]	0.2876	0.4226	0.4677
PSPNet[59]	0.2630	0.4063	0.4066
MPSPNet (our)	**0.3341**	0.4660	**0.5188**

this work, we adopt the same splitting criteria used in the PASCAL VOC challenge; 209 training, 213 validation, and 210 testing sets.

The experiments on PASCAL VOC 2007 are arranged to be independent. To examine our proposed architecture, we compare MPSPNet with six state-of-the-art networks. We use ResNet-101 in all networks for fine-tuning. The learning rate for the training is set as 1e-4, and each network is trained until convergence. As presented in Table 4.1, MPSPNet achieved 33% in terms of mIOU, which is the highest result using the same settings in all architectures.

4.5.1.2 NYUDv2

The NYU-Depth V2 dataset [61] includes 1449 RGB-D images captured form interior scenes of commercial and residential structures in multiple US cities. We apply the segmentation labels presented in [80], where all labels are mapped to 40 classes instead of 894. In our work, we only use RGB frames with the standard training/validation/testing split with 381, 414, and 654 images, respectively.

Table 4.2: Segmentation results on NYUDv2 test set.

Network	mIOU	mRecall	mPrec
AdapNet[75]	0.2072	0.3629	0.4467
BiSeNet[76]	0.3119	0.4505	**0.5832**
DeepLab-v3[77]	0.2315	0.3580	0.4900
DeepLab-v3Plus[78]	0.2541	0.3837	0.5173
RefineNet[41]	0.2415	0.3673	0.5006
PSPNet[59]	0.2689	0.3917	0.4697
MPSPNet (our)	**0.3332**	**0.4585**	0.5788

To check the performance of our MPSPNet on NYUDv2, we compare it with several state-of-the-art networks. We employ ResNet-101 in all architectures for fine-tuning. The learning rate applied for training is 1e-4, and each network trained until it converged. As shown in Table 4.2, MPSPNet reached 33% in terms of mIOU, which outperformed BiSeNet, which is the second-highest network, by 2%.

4.5.2 Hand(s) Segmentation

Since our main interest is in hand over face segmentation, we have conducted two experiments on a challenging hand over face pixel prediction problem. A single class (hand(s)) vs. background segmentation, and dual-class (right hand, left hand) vs. background segmentation, as discussed in Section 4.5.2.2 and Section 4.5.2.3 respectively. It is worth noting that both experiments are user-independent. Two public hand datasets are utilized in this work, as described in section 4.5.2.1.

4.5.2.1 Hand Datasets

To the best of our knowledge, there are two datasets designated for the hand over face problem. We use both of them in our work, and the following are the details of each one:

HOF HandOverFace (HOF) dataset by [37] has 300 pictures collected from the internet. All images contain hand(s) occlusion with the face in different shapes, sizes, and locations. The people in the dataset are from a variety of ethnicities, ages, and genders. Each image has a pixel-wise mask that are labeled as hand or background entity. Similar to Urooj and Borji [37], we choose the ratio of data split as 70% training, 10% validation, and 20% testing in our experiments.

VLM-HandOverFace The Vision Learning Mining Hand Over Face (VLM- HandOverFace) dataset was created by Ghanem *et al.* [3]. There are 42 subjects from different ethnicity's, genders, and ages. The recording was in a lab scene with diverse lighting conditions. The dataset contains 4384 frames with pixel level annotations. They provide two types of masks: (1) binary hand /background mask which denotes for each pixel whether it belongs to a hand(s) or not. (2) three classes of masks where each pixel is labeled as the right hand, left hand, or background. In our experiments, the data divided into 70%-10%-20% for training, validation, and testing, respectively.

4.5.2.2 Hand(s) Experiment

A one-class (hand(s)) segmentation experiment is performed to show the performance of our new proposed network (MPSPNet) along with several state-of-the-art architectures. VLM-HandOverFace and HOF datasets are used in this experiment. ResNet-101 pre-trained on Pascal Person-Parts employed in all networks for fine-tuning. The learning rate used for training is 1e-4, and each network trained until

Table 4.3: Hand(s) segmentation results on VLM-HandOverFace and HOF Datasets including the ablation experiments for newly added unites in our proposed architecture

Network	VLM-HandOverFace			HOF		
	mIOU	**mRecall**	**mPrec**	**mIOU**	**mRecall**	**mPrec**
AdapNet[75]	0.7617	0.8393	0.8463	0.6109	0.6828	0.7601
BiSeNet[76]	0.7837	0.9068	0.8292	0.7306	**0.8791**	0.7928
DeepLab-v3[77]	0.7478	0.8308	0.8292	0.6374	0.6997	0.8093
DeepLab-v3+[78]	0.7983	0.9261	0.8369	0.6974	0.8049	0.8030
RefineNet[41]	0.7951	0.8993	0.8338	0.7676	0.8832	0.8559
PSPNet[59]	0.8141	0.9294	0.8534	0.6543	0.7237	0.8115
PSPNet+CU	0.8254	0.9338	0.8648	0.6747	0.7422	0.8342
PSPNet+MCU	0.8355	0.9335	0.8760	0.7866	0.8486	**0.8976**
MPSPNet (PSPNet+CU+MCU)	**0.8560**	**0.9482**	**0.8898**	**0.8044**	0.8783	0.8933

convergence. Table 4.3 shows the pixel prediction results for hand(s) experiments. For VLM-HandOverFace dataset, MPSPNet achieves a ratio of 85% in terms of mIOU, which is 6% better than RefineNet and 4% better than PSPNet. Moreover, mPrec in MPSPNet improved by 5% and 3% over RefineNet and PSPNet, respectively. In the HOF dataset, the mIOU metric of MPSPNet has 4% improvement when compared to RefineNet and a 15% better than PSPNet. Also, MPSPNet improved by at least 4% in terms of mPrec more than the other networks. In Figure 4.3, the first three rows show examples of the prediction results using RefineNet, PSPNet, and MPSPNet on HOF dataset, and the rest of rows are for VLM-HandOverFace dataset. From the experiment, we notice that PSPNet performs better in a large dataset while RefineNet handles the smaller dataset. Our approach, MPSPNet successfully manages both sizes of datasets.

Ablation study for newly added units To evaluate our proposed architecture,

we conduct experiments on two added units: CU and MCU. We consider PSPNet as a baseline, and we perform (1) PSPNet+CU, (2) PSPNet+MCU, and (3) PSPNet+CU+MCU (which is our MPSPNet). As shown in Table 4.3, each unit gained at least 1% in all evaluation matrices on VLM-HandOverFace. Moreover, in HOF dataset, each added unit increased (mIOU, mRecall, mPrec) attributes by at least 2%.

4.5.2.3 Right/Left Hands Experiment

As a matter of fact, the detection and segmentation of the right and left hand is important information for many applications such as sign language recognition. We evaluate our MPSPNet by performing two classes of (right hand and left hand) pixel-level segmentation and compare it with multiple state-of-the-art networks. Since the VLM-HandOverFace dataset contains labeling information for the right and left hand, we employ it in this experiment. Similar to the hand(s) experiment, we tune the network using ResNet101 pre-trained on Pascal Person-Parts and adopted the same learning rate, 1e-4.

As shown in Table 4.4, MPSPNet outperformed RefineNet by 11% and PSPNet by 4% in regard to mIOU. Also, mRecall improved using our network by 9% compared with RefineNet and by 3% contrasted with PSPNet. Figure 4.4 presents detection examples using the three architectures.

Ablation study for newly added units To show the effect of CU and MCU, we performed experiments on each. We place PSPNet as a baseline, and we execute (1) PSPNet+CU, (2) PSPNet+MCU, and (3) PSPNet+CU+MCU (MPSPNet). As presented in Table 4.4, mIOU, mRecall, mPrec improved by at least 1% in each unit. Addition of the two units improved the overall results.

Table 4.4: Right Hand-Left Hand segmentation results using VLM-HandOverFace dataset including the ablation experiments for newly added unites in MPSPNet

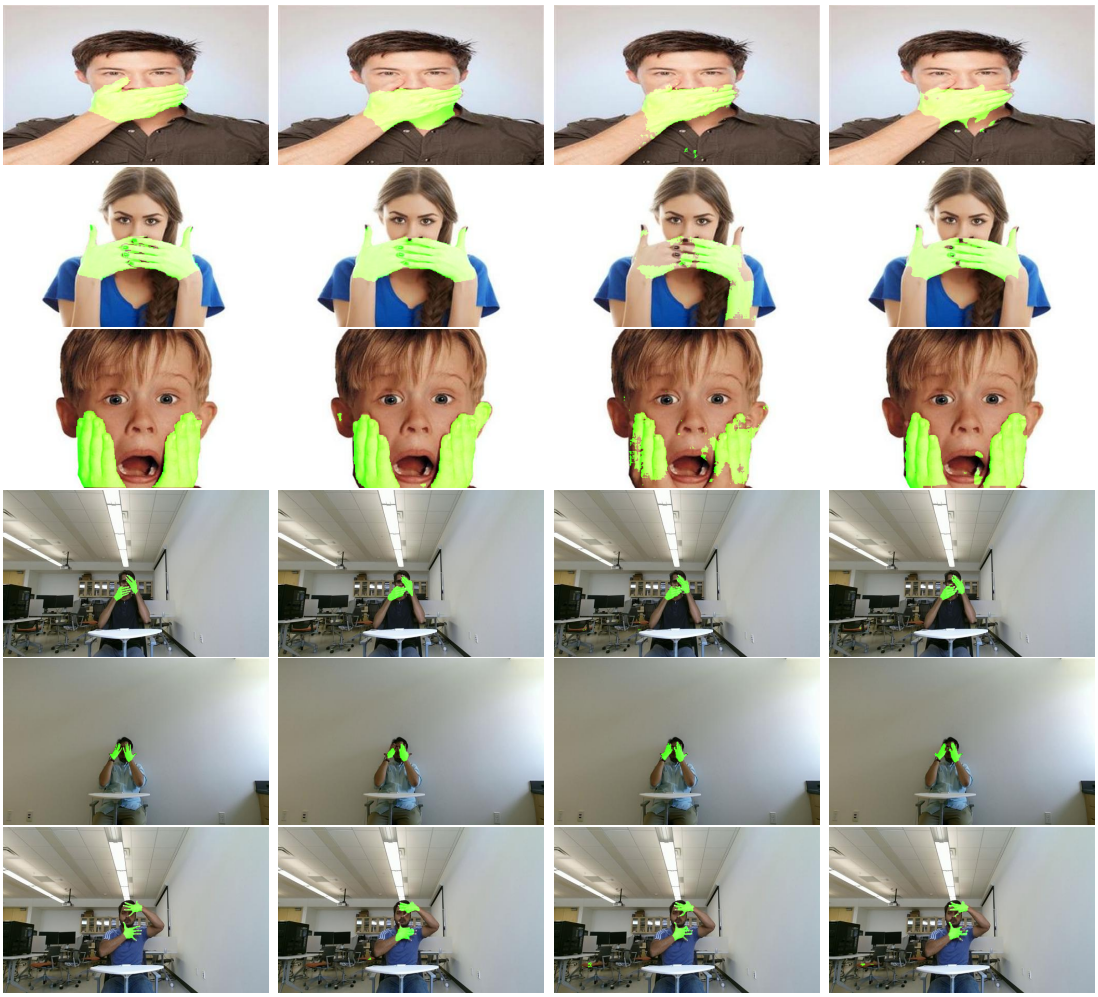
Network	**mIOU**	**mRecall**	**mPrec**
AdapNet[75]	0.6503	0.7433	0.7529
BiSeNet[76]	0.7191	0.8440	0.7912
DeepLab-v3[77]	0.6612	0.7575	0.7592
DeepLab-v3Plus[78]	0.7421	0.8627	0.8025
RefineNet[41]	0.6984	0.8100	**0.8491**
PSPNet[59]	0.7556	0.8707	0.8171
PSPNet+CU	0.7632	0.8839	0.8140
PSPNet+MCU	0.7879	0.8965	0.8399
MPSPNet (PSPNet+CU+MCU)	**0.8009**	**0.9082**	0.8458

4.6 Discussion and Conclusion

Accurate hand segmentation is a crucial task for several human interaction applications. In this work, we have addressed the challenging scenario of segmenting hands overlapping with the face, and we have introduced the Multi-level Pyramid Scene Parsing Network (MPSP-Net) for semantic segmentation. The Multi-level integration successfully extracts high-level features that help to predict the core region of the target. The pyramid pooling module was utilized to obtain global and local features that help to recognize the edges of the object of interest. MPSP-Net was evaluated and compared with RefineNet and PSPNet, both of which are among the state-of-the-art frameworks for semantic segmentation. Two types of experiments were conducted. First, a single class (hand) pixel-wise prediction was performed using two datasets: HOF and VLM-HandOverFace. The second evaluation considered two-class (right/left hands) segmentation using VLM-HandOverFace. In both exper-

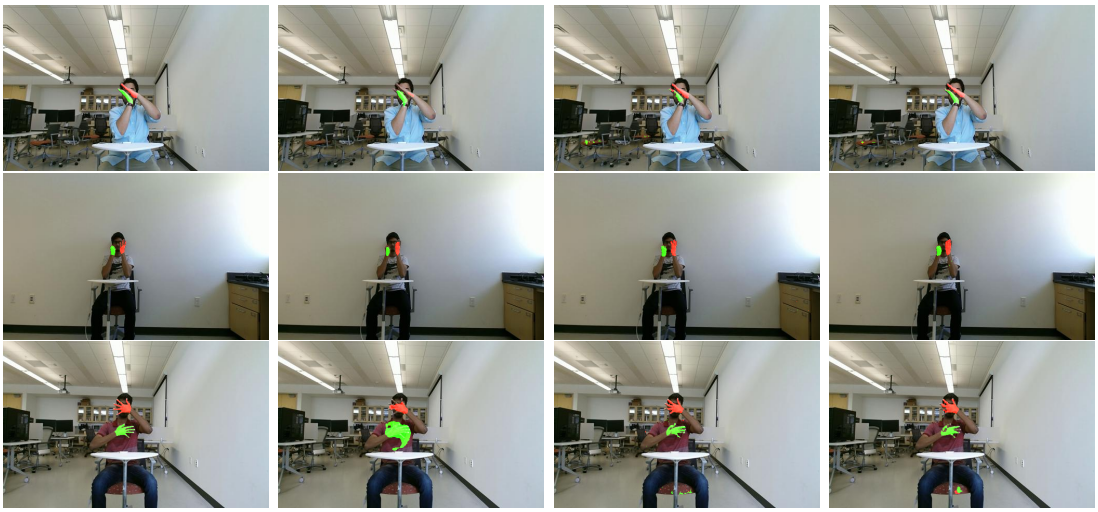
iments, our model achieved better outcomes in all metrics as compared to RefineNet and PSPNet.

Our proposed network employs the pyramid pooling module, which helps to extract the edge features of the hand palm and fingers. Our ablation studies experimentally show that the additional unit CU and PSPNet (as a baseline of our method), which represents the pyramid pooling module, improves the segmentation accuracy by at least 1%. Furthermore, the usage of the multi-path cascading technique, expressed by MCU in our architecture, assists in differentiating between hand and face as well as other objects in the scene. The implementation of MCU increases the segmentation accuracy in term of mIOU by more than 3%, as shown in Tables 4.3 and 4.4. The combination of these ideas leads to an overall improvement of over 4%.



(a) Ground Truth (b) RefineNet (c) PSPNet (d) MPSPNet

Figure 4.3: Examples of hand(s) predicted images after performing semantic segmentation methods on HOF (first three rows) and VLM-HandOverFace datasets.



(a) Ground Truth (b) RefineNet (c) PSPNet (d) MPSPNet

Figure 4.4: Examples of predicted images after performing right/left hands semantic segmentation experiment on VLM-HandOverFace dataset.

CHAPTER 5

Evaluation of MPSPNet with Motion Information

Video data can be useful for improving the segmentation result, where motion information adds an extra layer of worthy features. Recently, several attempts have been made to implement an unsupervised or semi-supervised semantic segmentation model based on the motion clue [81]. Many applications handle time sequence input data, such as sign language recognition, where the segmentation process is an essential step that influences the overall results. In this chapter, I discuss my observations about the impact of including motion information on MPSPNet, where optical flow and temporal RGB frames are included as an additional input in the conducted experiments.

The rest of this chapter will be organized as follow: related work is reviewed in section 5.1. Then I demonstrate the case of using FlowNet in section 5.2. The temporal frames experiment is presented in section 5.3. Finally, in section 5.4, I discuss the experimental results and the possible track of future work.

5.1 Related Work

One of the advantages of the VLM-HandOverFace dataset [3] is the availability of the source videos. Therefore, this valuable data can be used to study and improve the segmentation results using any Motion Segmentation techniques. We can define Motion segmentation as the process of classifying each pixel (or superpixel) in an image as a static or a dynamic point within the associated dimensions [82, 83]. The efforts in the motion segmentation domain can be categorized into three groups.

The first and the simplest, among other categories, is the image difference technique, where the pixel-wise difference between two frames is computed. The noises and lighting condition are challenging problems in this group. Examples of this category are employed in [84, 85]. The second group of motion segmentation techniques is the statistical approach. Several principles are utilized in this category, such as Maximum A posteriori Probability (MAP) [86, 87, 88], Particle Filter (PF) [89], and Expectation-Maximization (EM) [90]. The third category in motion segmentation tactics is the optical flow, which defined as the motion vector for each pixel in an image based on the brightness pattern from two consecutive frames. Optical flow is an old principle that was introduced by Horn and Schunck in 1981 [91]. Subsequently, many researchers enhanced the optical flow algorithm, but the most popular one was presented by Tomasi and Kanade in 1992 using the factorization technique [92]. Consequently, optical flow gained more attention, and it was utilized in several applications, such as identifying moving objects using the motion vector [83] and using the 3D motion vector [93]. Since the invention of Deep Neural Network, researchers applied it in various fields of studies, including the optical flow. Dosovitskiy et al. [94] introduce FlowNet for optical flow. The design of FlowNet includes two network architectures. First is FlowNetS, which is built by stacking two chronological frames as an input for the model. The second network architecture is FlowNetC that takes the two feature maps from the first architecture as input and contrasts them using a correlation layer. In [95], FlowNet 2.0 was proposed by stacking FlowNetS, and FlowNetC then runs them in a deeper network.

5.2 FlowNet

The latest state-of-the-art DNN optical flow algorithm is FlowNet 3.0 [96]. It was designed on the base of FlowNet 2.0 with some modifications. The new network

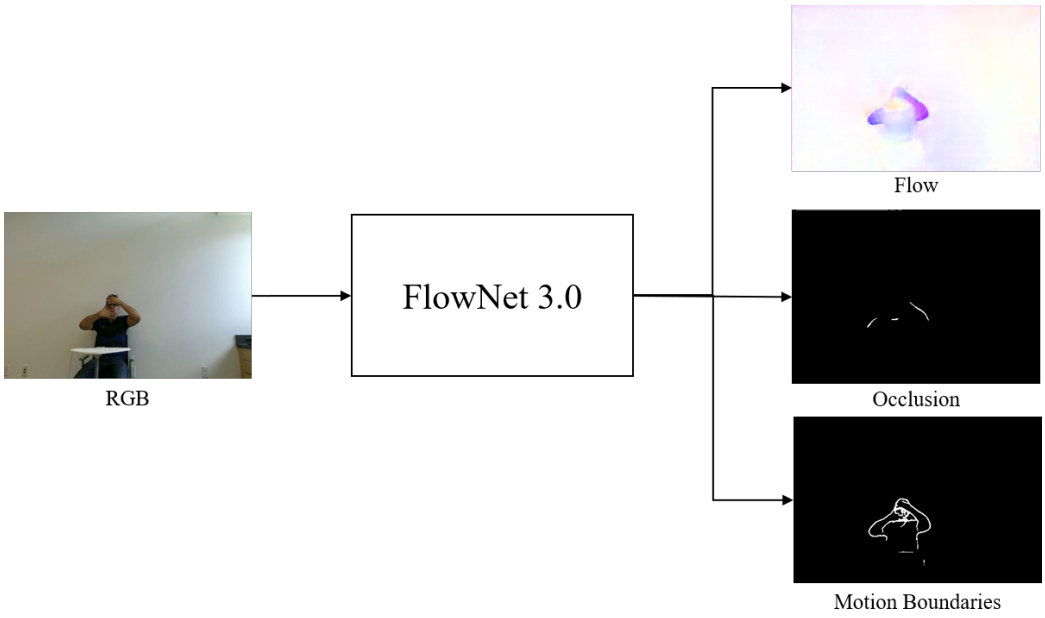


Figure 5.1: FlowNet 3.0 input/ output example.

architecture is consists of three stacked networks: FlowNetC and two FlowNetS. In the beginning, a fusion layer is utilized as a refinement of the input images. The first network (FlowNetC) is modified to perform dual forward and backward estimation with warping. In this stage, flows and occlusions are jointly detected by adding a two warping correlation between the forward and backward networks. The second network in the stack is the same as FlowNet 2.0, but it is redesigned to be dual as in the previous step. The last stacked network will benefit from the dual architecture to estimate the motion boundaries giving the flows and occlusions from the second stage. Figure 5.1 shows an example of the input and output of the FlowNet 3.0.

The outputs from FlowNet 3.0 (flow, occlusion, and motion boundaries) are included as an additional input to my work. In MPSPNet, ResNet-101 is used for fine-tuning where the input images are the three-channels (RGB) only. There is no fixed solution to handle more channels, and it depends on the problem. There are several workaround solutions, as follow:

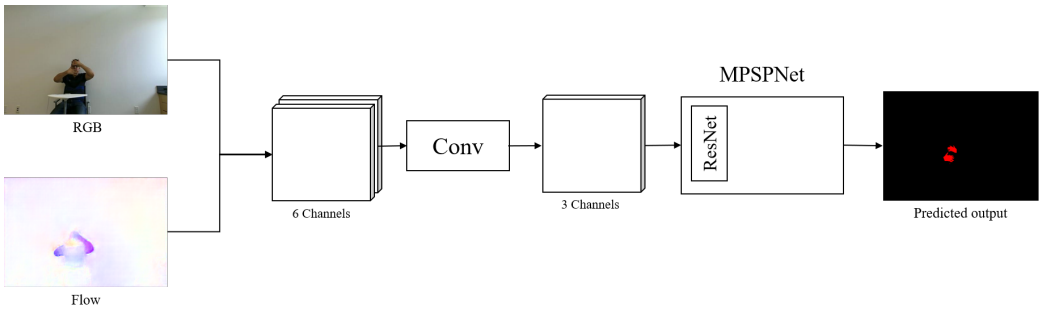


Figure 5.2: Modifying MPSPNet to handle additional channels input (method 1).

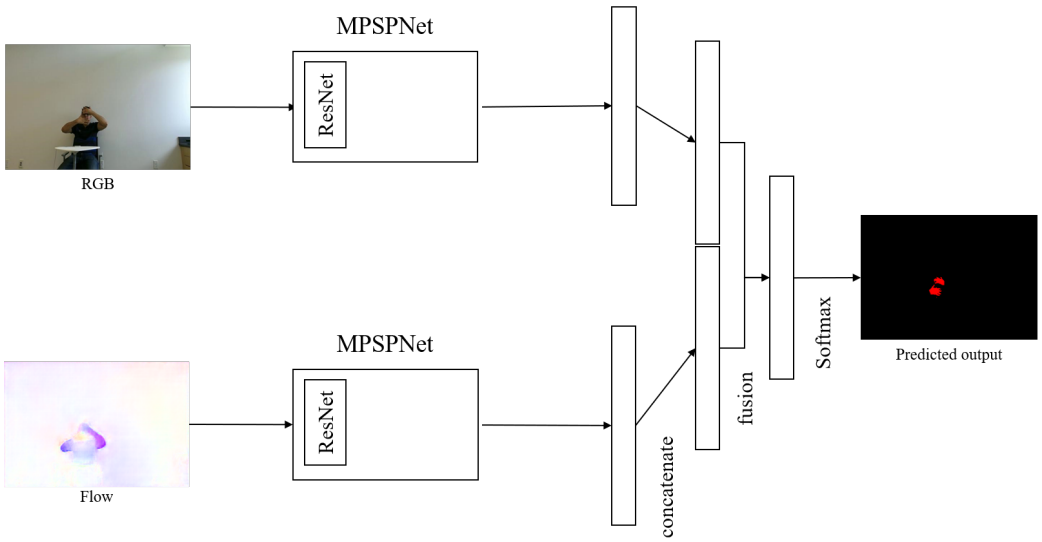


Figure 5.3: Modifying MPSPNet to handle additional channels input (method 2).

- Method 1: The most straightforward approach is to add one convolutional layer before ResNet to change the input image from x channels to a three channels. The design is shown in Figure 5.2
- Method 2: The idea in this method is to run parallel networks, each with a three channels as input, and before the softmax layer, we add a fusion layer to concatenate and merge both network's weights [97, 98] as in Figure 5.3.
- Method 3: This method aims to copy the weights from the regular three channels ResNet and duplicate it as additional channels (as needed). By using this technique, all other inputs are initialized with weights from the RGB image

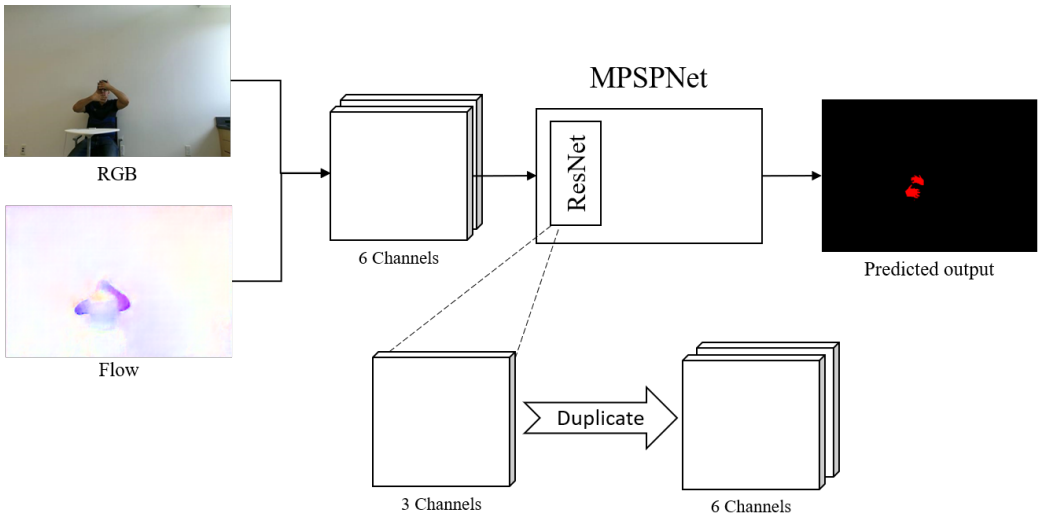


Figure 5.4: Modifying MPSPNet to handle additional channels input (method 3).

only. Figure 5.4 show the design of this method. This tactic was presented in [99].

The first way loses a lot of features in the first convolution layer, leading to a drop in the result. The second approach consumes more time without any improvement. The best method for our problem is the third one.

I conduct various experiments using FlowNet3 outcomes. Table 5.1 shows the experiment results using different orientations of flow inputs. As a result, the addition of FlowNet3 frames did not improve segmentation accuracy. The utilization of RGB + Flow frames in MPSPNet reduce the mIOU by 2% compared with the base RGB only model. Also, the usage of all FlowNet3 outcomes (flow + occlusion+ motion boundaries) drop the mIOU by 4%. Further, the joining of previous flow frames did not increase the evaluation matrices.

Table 5.1: Results of applying MPSPNet on different FlowNet/RGB input configuration

# Channels	Input Configuration	mIOU	mRecall	mPrec
3	RGB (base)	**0.8572**	0.9265	**0.8977**
6	RGB+Flow	0.8383	**0.9296**	0.8716
12	RGB+Flow+Occlusion+Motion Boundaries	0.8174	0.9052	0.8620
9	Previous_Flow+RGB+Flow	0.8230	0.9234	0.8591
12	Previous_RGB+Previous_Flow+RGB+Flow	0.7945	0.8818	0.8539

5.3 Temporal frames

The second technique I tried to improve segmentation results by combining different arrangements of temporal frames as an input to MPSPNet. While the adding of one previous RGB frame slightly decreases the mIOU, mRecall, and mPrec, adding two previous frames reduces all evaluation metrics by 1%. Another common arrangement for time series input is by using the following equation:

$$2K\ Prev\_RGB + K\ Prev\_RGB + RGB + K\ Next\_RGB + 2K\ Next\_RGB$$

I did several experiments where K ranges from 1 to 5. The outcome was a significant drop in mIOU by at least 11%. Moreover, a nine channels input was used as:

$$5\ Prev\_RGB + RGB + 5\ Next\_RGB$$

and

$$10\ Prev\_RGB + RGB + 10\ Next\_RGB$$

Table 5.2: Results of applying MPSPNet on different input configuration using FlowNet data

# Channels	Input Configuration	mIOU	mRecall	mPrec
3	RGB (base)	0.8572	**0.9265**	0.8977
6	Prev_RGB+RGB	0.8538	0.9245	0.8950
9	Prev_Prev_RGB+Prev_RGB+RGB	0.8456	0.9220	0.8864
9	Prev_RGB+RGB+Next_RGB	**0.8582**	0.9247	0.8999
9	5Prev_RGB+RGB+5Next_RGB	0.8324	0.9111	0.8784
9	10Prev_RGB+RGB+10Next_RGB	0.8324	0.8801	**0.9087**
15	2K_Prev_RGB+K_Prev_RGB+RGB+K_Next_RGB+2K_Next_RGB			
15	where K=1	0.7404	0.7640	0.9028
15	where K=2	0.7348	0.8649	0.7687
15	where K=3	0.7430	0.7693	0.8922
15	where K=4	0.7225	0.7307	0.8991
15	where K=5	0.7229	0.7654	0.8421

The mIOU of both input combinations tests was reduced by 2%. One arrangement that gets a tiny improvement is by combining the current frame with the previous frame and the next frame.

$$Prev\_RGB + RGB + Next\_RGB$$

This input pattern increases the mIOU from 84.72% to 84.82%. Table 5.2 presents the experiment results using the addressed input patterns.

5.4 Discussion

In contrast to our expectation, adding temporal motion information to MPSP-Net did not yield a noticeable improvement to the segmentation results. The usage of ResNet in MPSPNet appeared to behave as a limiting factor in the challenge. Optical flow affords valuable details about the motion in the image. Also, temporal frames can be used to match and extract additional features. Experimentally, both of them did not strengthen the segmentation process toward providing relatively tangible outcomes. Modifying the design of MPSPNet to handle video motion information can be considered a possible path for future work.

CHAPTER 6

CONCLUSION

This dissertation investigated hand-over-face segmentation challenges from many perspectives: (a) an essential example application that reveals the need for a reliable solution was discussed. (b) a review of all available hand segmentation datasets was reported. (c) the creation of a challenging hand-over-face segmentation was presented, and (d) an adequate solution was proposed. The contributions in this work are as follow:

1. I presented a survey of all existing sign language recognition applications built on mobile phones.
2. All existing public hand segmentation datasets was reviewed and analyzed in terms of pros and cons.
3. I created a new public dataset for the hand-over-face segmentation.
4. A Multi-level Pyramid Scene Parsing Network (MPSP-Net) for semantic segmentation problem was proposed. The unique characteristics that make this model proper for a hand-over-face segmentation challenge were discussed.
5. I provided a study on the consequence of utilizing video motion information from the VLM-HandOverFace dataset on the (MPSP-Net).

REFERENCES

[1] E. Costello, *American sign language dictionary*. Random House Reference &, 2008.

[2] S. Ghanem, C. Conly, and V. Athitsos, "A survey on sign language recognition using smartphones," in *Proceedings of the 10th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, ser. PETRA '17. New York, NY, USA: Association for Computing Machinery (ACM), 2017, pp. 171–176. [Online]. Available: https://doi.org/10.1145/3056540.3056549

[3] S. Ghanem, A. Imran, and V. Athitsos, "Analysis of hand segmentation on challenging hand over face scenario," in *Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, ser. PETRA '19. New York, NY, USA: Association for Computing Machinery (ACM), 2019, p. 236–242. [Online]. Available: https://doi.org/10.1145/3316782.3321534

[4] S. Ghanem, A. Dillhoff, A. Imran, and V. Athitsos, "Hand over face segmentation using mpspnet," in *Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, ser. PETRA '20. New York, NY, USA: Association for Computing Machinery (ACM), 2020, pp. 257–264. [Online]. Available: https://doi.org/10.1145/3389189.3397970

[5] S. Hamrick, L. Jacobi, P. Oberholtzer, E. Henry, and J. Smith, "Libguides. deaf statistics. deaf population of the us." *Montana*, vol. 16, no. 616,796, pp. 2–7, 2010.

[6] M. Seymour and M. Tšoeu, "A mobile application for south african sign language (sasl) recognition," in *AFRICON, 2015*. IEEE, 2015, pp. 1–5.

[7] L.-J. Kau, W.-L. Su, P.-J. Yu, and S.-J. Wei, "A real-time portable sign language translation system," in *2015 IEEE 58th International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 2015, pp. 1–4.

[8] C. Preetham, G. Ramakrishnan, S. Kumar, A. Tamse, and N. Krishnapura, "Hand talk-implementation of a gesture recognizing glove," in *India Educators' Conference (TIIEC), 2013 Texas Instruments*. IEEE, 2013, pp. 328–331.

[9] B. Choe, J.-K. Min, and S.-B. Cho, "Online gesture recognition for user interface on accelerometer built-in mobile phones," in *International Conference on Neural Information Processing*. Springer, 2010, pp. 650–657.

[10] H. P. Gupta, H. S. Chudgar, S. Mukherjee, T. Dutta, and K. Sharma, "A continuous hand gestures recognition technique for human-machine interaction using accelerometer and gyroscope sensors," *IEEE Sensors Journal*, vol. 16, no. 16, pp. 6425–6432, 2016.

[11] M. Joselli and E. Clua, "grmobile: A framework for touch and accelerometer gesture recognition for mobile games," in *2009 VIII Brazilian Symposium on Games and Digital Entertainment*. IEEE, 2009, pp. 141–150.

[12] G. Niezen and G. P. Hancke, "Gesture recognition as ubiquitous input for mobile phones," in *International Workshop on Devices that Alter Perception (DAP 2008), in conjunction with Ubicomp*. Citeseer, 2008, pp. 17–21.

[13] X. Wang, P. Tarrío, E. Metola, A. M. Bernardos, and J. R. Casar, "Gesture recognition using mobile phone's inertial sensors," in *Distributed Computing and Artificial Intelligence*. Springer, 2012, pp. 173–184.

[14] H. Wang, A. Stefan, S. Moradi, V. Athitsos, C. Neidle, and F. Kamangar, "A system for large vocabulary sign search," in *European Conference on Computer Vision*. Springer, 2010, pp. 342–353.

[15] J. Kruskall and M. Liberman, "The symmetric time warping algorithm: From continuous to discrete. time warps, string edits and macromolecules," 1983.

[16] G. A. Rao and P. Kishore, "Sign language recognition system simulated for video captured with smart phone front camera," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 6, no. 5, pp. 2176–2187, 2016.

[17] H. Elleuch, A. Wali, A. Samet, and A. M. Alimi, "A static hand gesture recognition system for real time mobile device monitoring," in *Intelligent Systems Design and Applications (ISDA), 2015 15th International Conference on*. IEEE, 2015, pp. 195–200.

[18] P. Gandhi, D. Dalvi, P. Gaikwad, and S. Khode, "Image based sign language recognition on android," *International Journal of Engineering and Techniques*, vol. 1, no. 5, pp. 55–60, 2015.

[19] R. Y. Hakkun, A. Baharuddin *et al.*, "Sign language learning based on android for deaf and speech impaired people," in *Electronics Symposium (IES), 2015 International*. IEEE, 2015, pp. 114–117.

[20] P. Hays, R. Ptucha, and R. Melton, "Mobile device to cloud co-processing of asl finger spelling to text conversion," in *Image Processing Workshop (WNYIPW), 2013 IEEE Western New York*. IEEE, 2013, pp. 39–43.

[21] C. M. Jin, Z. Omar, and M. H. Jaward, "A mobile application of american sign language translation via image processing algorithms," in *Region 10 Symposium (TENSYMP), 2016 IEEE*. IEEE, 2016, pp. 104–109.

[22] T. J. Joshi, S. Kumar, N. Tarapore, and V. Mohile, "Static hand gesture recognition using an android device," *International Journal of Computer Applications*, vol. 120, no. 21, 2015.

[23] R. Kamat, A. Danoji, A. Dhage, P. Puranik, and S. Sengupta, "Monvoix-an android application for hearing impaired people," *Journal of Communications Technology, Electronics and Computer Science*, vol. 8, pp. 24–28, 2016.

[24] H. Lahiani, M. Elleuch, and M. Kherallah, "Real time hand gesture recognition system for android devices," in *Intelligent Systems Design and Applications (ISDA), 2015 15th International Conference on.* IEEE, 2015, pp. 591–596.

[25] L. Prasuhn, Y. Oyamada, Y. Mochizuki, and H. Ishikawa, "A hog-based hand gesture recognition system on a mobile device," in *2014 IEEE International Conference on Image Processing (ICIP).* IEEE, 2014, pp. 3973–3977.

[26] J. L. Raheja, A. Singhal, and A. Chaudhary, "Android based portable hand sign recognition system," *arXiv preprint arXiv:1503.03614*, 2015.

[27] A. Saxena, D. K. Jain, and A. Singhal, "Hand gesture recognition using an android device," in *Communication Systems and Network Technologies (CSNT), 2014 Fourth International Conference on.* IEEE, 2014, pp. 819–822.

[28] ——, "Sign language recognition using principal component analysis," in *Communication Systems and Network Technologies (CSNT), 2014 Fourth International Conference on.* IEEE, 2014, pp. 810–813.

[29] K. S. Warrier, J. K. Sahu, H. Halder, R. Koradiya, and V. K. Raj, "Software based sign language converter," in *Communication and Signal Processing (ICCSP), 2016 International Conference on.* IEEE, 2016, pp. 1777–1780.

[30] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[31] S. L. Phung, A. Bouzerdoum, and D. Chai, "Skin segmentation using color pixel classification: analysis and comparison," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 1, pp. 148–154, 2005.

[32] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[33] J. Weston and C. Watkins, "Multi-class support vector machines," Citeseer, Tech. Rep., 1998.

[34] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, no. 3, p. 1, 1988.

[35] J. D. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning*. Williamstown, MA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 282–289.

[36] S. Bambach, S. Lee, D. J. Crandall, and C. Yu, "Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions," in *2015 IEEE International Conference on Computer Vision (ICCV)*. Santiago, Chile: IEEE, 2015, pp. 1949–1957.

[37] A. Urooj and A. Borji, "Analysis of hand segmentation in the wild," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, UT: IEEE, 2018, pp. 4710–4719.

[38] A. Fathi, X. Ren, and J. M. Rehg, "Learning to recognize objects in egocentric activities," in *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Colorado Springs, CO, USA: IEEE, 2011, pp. 3281–3288.

[39] M. Gonzalez, C. Collet, and R. Dubot, "Head tracking and hand segmentation during hand over face occlusion in sign language," in *European Conference on Computer Vision*. Heraklion, Crete, Greece: Springer, 2010, pp. 234–243.

[40] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Transactions on Graphics (ToG)*, vol. 33, no. 5, p. 169, 2014.

[41] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, Hawaii, USA: IEEE, 2017, pp. 5168–5177.

[42] V. Badrinarayanan, A. Handa, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," *arXiv preprint arXiv:1505.07293*, 2015.

[43] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[44] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *2015 IEEE International Conference on Computer Vision (ICCV)*. Santiago, Chile: IEEE, 2015, pp. 1520–1528.

[45] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, "Conditional random fields as recurrent neural networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*. Santiago, Chile: IEEE, 2015, pp. 1529–1537.

[46] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *2015 IEEE International Conference on Computer Vision (ICCV)*. Santiago, Chile: IEEE, 2015, pp. 2650–2658.

[47] S. Hong, H. Noh, and B. Han, "Decoupled deep neural network for semi-supervised semantic segmentation," in *Proceedings of the 28th International Con-*

ference on Neural Information Processing Systems-Volume 1. MONTREAL, CANADA: MIT Press, 2015, pp. 1495–1503.

[48] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention.* Munich, Germany: Springer, 2015, pp. 234–241.

[49] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Transactions on Graphics (ToG)*, vol. 33, no. 5, pp. 1–10, August 2014.

[50] D. Tang, H. Jin Chang, A. Tejani, and T.-K. Kim, "Latent regression forest: Structured estimation of 3d articulated hand posture," in *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* Columbus, OH, USA: IEEE, 2014, pp. 3786–3793.

[51] S. Yuan, Q. Ye, B. Stenger, S. Jain, and T.-K. Kim, "Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* Honolulu, HI, USA: IEEE, 2017, pp. 2605–2613.

[52] M. Mahmoud, R. El-Kaliouby, and A. Goneid, "Towards communicative face occlusions: machine detection of hand-over-face gestures," in *International Conference Image Analysis and Recognition.* Halifax, NS, Canada: Springer, 2009, pp. 481–490.

[53] B. Nojavanasghari, C. E. Hughes, T. Baltrušaitis, and L.-P. Morency, "Hand2face: Automatic synthesis and recognition of hand over face occlusions," in *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII).* San Antonio, TX, USA: IEEE, 2017, pp. 209–215.

[54] A. Kurakin, Z. Zhang, and Z. Liu, "A real time system for dynamic hand gesture recognition with a depth sensor." in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*. Bucharest, Romania: IEEE, 2012, pp. 1975–1979.

[55] M. Mahmoud, T. Baltrušaitis, P. Robinson, and L. D. Riek, "3d corpus of spontaneous complex mental states," in *International Conference on Affective Computing and Intelligent Interaction*. Memphis, TN, USA: Springer, 2011, pp. 205–214.

[56] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 12, pp. 2930–2940, 2013.

[57] J. F. Kooij, "Sensecap: synchronized data collection with microsoft kinect2 and leapmotion," in *Proceedings of the 2016 ACM on Multimedia Conference*. New York, NY, USA: ACM, 2016, pp. 1218–1221.

[58] D. K. Iakovidis, T. Goudas, C. Smailis, and I. Maglogiannis, "Ratsnake: a versatile image annotation tool with application to computer-aided diagnosis," *The Scientific World Journal*, vol. 2014, 2014.

[59] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, USA: IEEE, 2017, pp. 6230–6239.

[60] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[61] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *European conference on computer vision*. Florence, Italy: Springer, 2012, pp. 746–760.

[62] A. A. Argyros and M. I. Lourakis, "Real-time tracking of multiple skin-colored objects with a possibly moving camera," in *European Conference on Computer Vision*. Berlin, Heidelberg: Springer, 2004, pp. 368–379.

[63] M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection," *International Journal of Computer Vision*, vol. 46, no. 1, pp. 81–96, 2002.

[64] P. Kakumanu, S. Makrogiannis, and N. Bourbakis, "A survey of skin-color modeling and detection methods," *Pattern recognition*, vol. 40, no. 3, pp. 1106–1122, 2007.

[65] J. M. Rehg and T. Kanade, "Visual tracking of high dof articulated structures: an application to human hand tracking," in *European conference on computer vision*. Stockholm, Sweden: Springer, 1994, pp. 35–46.

[66] B. Stenger, P. R. Mendonça, and R. Cipolla, "Model-based 3d tracking of an articulated hand," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 2. Kauai, HI, USA: IEEE, 2001, pp. II–II.

[67] E. B. Sudderth, M. I. Mandel, W. T. Freeman, and A. S. Willsky, "Visual hand tracking using nonparametric belief propagation," in *2004 Conference on Computer Vision and Pattern Recognition Workshop*. Washington, DC, USA: IEEE, 2004, pp. 189–189.

[68] E. Hayman and J.-O. Eklundh, "Statistical background subtraction for a mobile observer," in *Proceedings of the Ninth IEEE International Conference on Computer Vision-Volume 2*. Nice, France: IEEE, 2003, pp. 67–74.

[69] Y. Sheikh, O. Javed, and T. Kanade, "Background subtraction for freely moving cameras," in *2009 IEEE 12th International Conference on Computer Vision*. Kyoto, Japan: IEEE, 2009, pp. 1219–1225.

[70] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun, "Realtime and robust hand tracking from depth," in *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Columbus, OH, USA: IEEE, 2014, pp. 1106–1113.

[71] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*. Providence, RI, USA: IEEE, 2011, pp. 1297–1304.

[72] B. Kang, K.-H. Tan, N. Jiang, H.-S. Tai, D. Tretter, and T. Nguyen, "Hand segmentation for hand-object interaction from depth map," in *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. Montreal, QC, Canada: IEEE, 2017, pp. 259–263.

[73] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, Massachusetts, USA: IEEE, 2015, pp. 3431–3440.

[74] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[75] A. Valada, J. Vertens, A. Dhall, and W. Burgard, "Adapnet: Adaptive semantic segmentation in adverse environmental conditions," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. Singapore, Singapore: IEEE, 2017, pp. 4644–4651.

[76] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *European Conference on Computer Vision*. Munich, Germany: Springer, 2018, pp. 334–349.

[77] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017.

[78] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *European Conference on Computer Vision*. Munich, Germany: Springer, 2018, pp. 833–851.

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

[80] S. Gupta, P. Arbelaez, and J. Malik, "Perceptual organization and recognition of indoor scenes from rgb-d images," in *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Portland, OR, USA: IEEE, 2013, pp. 564–571.

[81] Y. He, W.-C. Chiu, M. Keuper, and M. Fritz, "Std2p: Rgbd semantic segmentation using spatio-temporal data-driven pooling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4837–4846.

[82] M. Derome, A. Plyer, M. Sanfourche, and G. L. Besnerais, "Moving object detection in real-time using stereo from a mobile platform," *Unmanned Systems*, vol. 3, no. 04, pp. 253–266, 2015.

[83] J. Klappstein, T. Vaudrey, C. Rabe, A. Wedel, and R. Klette, "Moving object segmentation using optical flow and depth information," in *Pacific-Rim Symposium on Image and Video Technology*. Springer, 2009, pp. 611–623.

[84] A. Cavallaro, O. Steiger, and T. Ebrahimi, "Tracking video objects in cluttered background," *IEEE transactions on circuits and systems for video technology*, vol. 15, no. 4, pp. 575–584, 2005.

[85] R. Li, S. Yu, and X. Yang, "Efficient spatio-temporal segmentation for extracting moving objects in video sequences," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 3, pp. 1161–1167, 2007.

[86] C. Rasmussen and G. D. Hager, "Probabilistic data association methods for tracking complex visual objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 560–576, 2001.

[87] D. Cremers and S. Soatto, "Motion competition: A variational approach to piecewise parametric motion segmentation," *International Journal of Computer Vision*, vol. 62, no. 3, pp. 249–265, 2005.

[88] H. Shen, L. Zhang, B. Huang, and P. Li, "A map approach for joint motion estimation, segmentation, and super resolution," *IEEE Transactions on Image processing*, vol. 16, no. 2, pp. 479–490, 2007.

[89] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi, "Tracking deforming objects using particle filtering for geometric active contours," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 8, pp. 1470–1475, 2007.

[90] R. Stolkin, A. Greig, M. Hodgetts, and J. Gilby, "An em/e-mrf algorithm for adaptive model based tracking in extremely poor visibility," *Image and Vision Computing*, vol. 26, no. 4, pp. 480–495, 2008.

[91] B. K. Horn and B. G. Schunck, "Determining optical flow," in *Techniques and Applications of Image Understanding*, vol. 281. International Society for Optics and Photonics, 1981, pp. 319–331.

[92] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *International journal of computer vision*, vol. 9, no. 2, pp. 137–154, 1992.

[93] P. F. Alcantarilla, J. J. Yebes, J. Almazán, and L. M. Bergasa, "On combining visual slam and dense scene flow to increase the robustness of localization and mapping in dynamic environments," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 1290–1297.

[94] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2758–2766.

[95] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2462–2470.

[96] E. Ilg, T. Saikia, M. Keuper, and T. Brox, "Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 614–630.

[97] S. Guo and Z. Yang, "Multi-channel-resnet: An integration framework towards skin lesion analysis," *Informatics in Medicine Unlocked*, vol. 12, pp. 67–74, 2018.

[98] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal deep learning for robust rgb-d object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 681–687.

[99] K. Liu, Z. Zhang, and W. Zhou, "Large-scale protein atlas compartmentalization analysis," *stanford*, 2018.

BIOGRAPHICAL STATEMENT

Sakher Ghanem was born in Jeddah, Saudi Arabia. He received his Bachelor and Masters' degree in Computer Science from the King Abdulaziz University, Saudi Arabia, in 2002 and 2008 respectively. In 2014, he started his studies to pursue his Ph.D at the University of Texas at Arlington. His current research interests include Computer vision, and Machine learning.