**T.R.**
**GEBZE TECHNICAL UNIVERSITY**
**GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**

# MONITORING HAND HYGIENE VIA TEMPORAL VIDEO PROCESSING

**FURKAN KASIM**
**A THESIS SUBMITTED FOR THE DEGREE OF**
**MASTER OF SCIENCE**
**DEPARTMENT OF INDUSTRIAL ENGINEERING**

**GEBZE**
**2022**

**T.R.**

**GEBZE TECHNICAL UNIVERSITY**

**GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**

# MONITORING HAND HYGIENE VIA TEMPORAL VIDEO PROCESSING

**FURKAN KASIM**

**A THESIS SUBMITTED FOR THE DEGREE OF**

**MASTER OF SCIENCE**

**DEPARTMENT OF INDUSTRIAL ENGINEERING**

THESIS SUPERVISOR

ASSOC. PROF. DR. AYŞENUR BUDAK

II. THESIS SUPERVISOR

ASSIST. PROF. DR. YAKUP GENÇ

**GEBZE**

**2022**

**T.C.**
**GEBZE TEKNİK ÜNİVERSİTESİ**
**FEN BİLİMLERİ ENSTİTÜSÜ**

# ZAMANSAL VİDEO İŞLEME İLE EL HİJYEN TAKİBİ

**FURKAN KASIM**
**YÜKSEK LİSANS TEZİ**
**ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI**

DANIŞMANI
DOÇ. DR. AYŞENUR BUDAK
II. DANIŞMANI
DR. ÖĞR. ÜYESİ YAKUP GENÇ

**GEBZE**
**2022**

| | |
|---|---|
| GEBZE TEKNİK ÜNİVERSİTESİ | **YÜKSEK LİSANS JÜRİ ONAY FORMU** |

GTÜ Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 23/06/2022 tarih ve 2022/31 sayılı kararıyla oluşturulan jüri tarafından 18/07/2022 tarihinde tez savunma sınavı yapılan Furkan Kasım'ın tez çalışması Endüstri Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

**JÜRİ**

| | |
|---|---|
| ÜYE (TEZ DANIŞMANI) | : Doç. Dr. Ayşenur BUDAK |
| ÜYE | : Dr. Öğr. Üyesi Yakup GENÇ |
| ÜYE | : Prof. Dr. Erchan APTOULA |
| ÜYE | : Dr. Öğr. Üyesi Ahmet Burak PAÇ |
| ÜYE | : Doç. Dr. Ayşe Betül OKTAY |

**ONAY**

Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun

...../...../…… tarih ve ……./…..... sayılı kararı.

# SUMMARY

The rapid spread of infectious diseases is caused by excessive hand contact with people or surfaces. These increased infection rates can result in patient problems, lengthy hospital stays, financial difficulties and even deaths. Therefore, hand hygiene has been of more vital importance than ever, especially due to the Coronavirus disease pandemic. Successful hand sanitization using the World Health Organization's (WHO) suggested hand movement sequences is a crucial procedure in preserving hygiene and reducing contamination. The purpose of this thesis is to offer a hand sanitization monitoring framework that can be used to ensure that the proper handwashing procedure is followed. Approaches in the world of computer vision and deep learning are utilized in the action recognition of hand movements in the handwashing process. The optical flow method, which addresses the movement of pixels in the image plane, is used to enrich single and multiple images taken from RGB handwashing videos. For classifying hand movements, diverse deep learning approaches based on convolutional neural network (CNN), long short-term memory (LSTM), and vision transformer (ViT) are fed with these enriched inputs. Our experiments clearly show that optical flow advances the performance of models when used along with original images. Besides, the best results are obtained by the end-to-end CNN-LSTM model including the angles of motion, which tends to better capture the temporal dependence of hands (93.84%).

**Key Words: Deep Learning, Computer Vision, Optical Flow, Convolution Neural Network (CNN), Long Short-Term Memory (LSTM), Vision Transformer (ViT).**

# ÖZET

Elin insanlarla ve yüzeylerle olan aşırı teması, bulaşıcı hastalıkların hızla yayılmasına sebep olmaktadır. Artan bu infeksiyon hızı da hasta sorunlarına, hastanede uzun kalış sürelerine, finansal problemlere ve hatta ölümlere neden olabilmektedir. Bundan dolayı, özellikle de Coronavirüs hastalığı pandemisi nedeniyle, el hijyeni her zamankinden daha hayati bir önem kazanmıştır. Dünya Sağlık Örgütü'nün önermiş olduğu belirli el hareketleriyle gerçekleştiren başarı el yıkama, hijyeni korumak ve kontaminasyonu önlemek için gerekli bir prosedürdür. Bu tezin amacı, başarılı el yıkama sürecini takip etmek için kullanılabilecek bir el yıkama izleme sistemi sunmaktır. El yıkamada sürecinde hareketlerin tanınmasında bilgisayarlı görme ve derin öğrenme dünyasındaki yaklaşımlardan yararlanılmaktadır. Görüntü düzlemindeki piksellerin hareketini ele alan optik akış yöntemi, RGB el yıkama videolarından alınan tekli ve çoklu görüntüleri zenginleştirmek için kullanılır. El hareketlerini sınıflandırmak için, evrişimsel sinir ağı (CNN), uzun kısa süreli bellek (LSTM) ve görüntü dönüştürücü (ViT) tabanlı çeşitli derin öğrenme yaklaşımları bu zenginleştirilmiş girdilerle beslenir. Deneylerimiz, optik akışın orijinal görüntülerle birlikte kullanıldığında modellerin performansını iyileştirdiğini açıkça göstermektedir. Ayrıca, en iyi sonuçlar, ellerin zamansal ilişkisini daha iyi yakalama eğiliminde olan ve optik akış açılarını içeren uçtan uca CNN-LSTM modeli ile elde edilmiştir (%93,84).

**Anahtar Kelimeler: Derin Öğrenme, Bilgisayarla Görme, Optik Akış, Evrişimsel Sinir Ağları (ESA), Uzun Kısa Süreli Bellek Ağları (UKSBA), Görüntü Dönüştürücü (GD).**

# ACKNOWLEDGEMENTS

First of all, I would like to thank the Department of Computer and Industrial Engineering at Gebze Technical University and my valuable advisor Assoc. Prof. Dr. Ayşenur Budak for all the opportunities they have provided.

I am grateful to my esteemed thesis advisor, Assist. Prof. Yakup Genç, who guided me in the academia and science world, for everything he taught. I would like to thank him very much for instilling a researcher spirit in me, for patiently solving all my problems during my thesis process, and for being an inspiration to me.

My sincere thanks also go to my family and friends for making me feel that I am not alone in this rocky road.

Finally, I would like to say "so glad I have you" to my fiancée Sinem for her endless love, support, and attention.

# TABLE of CONTENTS

# LIST OF ABBREVIATIONS AND ACRONYMS

| Abbreviations and Acronyms | | Explanations |
|---|---|---|
| 2D | : | 2 Dimensional |
| 3D | : | 3 Dimensional |
| Bi-LSTM | : | Bi-directional Long Short-Term Memory |
| CNN | : | Convolutional Neural Network |
| CT | : | Computed Tomography |
| CV | : | Computer Vision |
| DL | : | Deep Learning |
| EMG | : | Electromyography |
| FCL | : | Fully-Connected Layer |
| HCI | : | Human Computer Interaction |
| HGR | : | Hand Gesture Recognition |
| HSV | : | Hue-Saturation-Value |
| LSTM | : | Long Short-Term Memory |
| NLP | : | Natural Language Processing |
| ReLU | : | Rectifier Linear Unit |
| RGB | : | Red-Green-Blue |
| RNN | : | Recurrent Neural Network |
| ViT | : | Vision Transformer |
| WHO | : | World Health Organization |

# LIST OF FIGURES

# LIST OF TABLES

# 1. INTRODUCTION

In recent years, technological advancements have had spectacular effects in our world and routine life. All of these advancements have made life easier, faster, and better for humans. Besides, in these days of the coronavirus epidemic, it is apparent that technology has brought about enormous and pleasant changes in the health industry as well. With several innovations such as mask detection and disease diagnosis from chest computed tomography (CT), it has played a key role in human health. Due to the increasing number of infectious diseases today, the numerous contacts of people with diverse surfaces or with different people during the day increases the transmission of the disease. As a result, decreasing the risk of disease transmission and maintaining hygiene are critical for the proper management of this process. In this context, hand hygiene must be fully ensured in order to prevent the spread of diseases such as coronavirus and flu, which can be easily transmitted through touching [1]. The main purpose of such research is to develop a framework that can monitor hand hygiene and be able to check the movements of handwashing. Different methodologies that evolved in the world of deep learning (DL) have also achieved impressive progress on computer vision challenges. Therefore, it is aimed to overcome this challenge by making use of the solutions offered by computer vision and deep learning methods.

In this thesis, we determine the handwashing movements that are specified by WHO. Subsequently, we offer a system to recognize predefined hand movements from RGB handwashing videos. Besides, we provide an extensive solution for hand gesture analysis by applying novel methods to basic deep learning models. To the best of our knowledge, this is the first time that the proposed approaches for this problem have been addressed in this thesis.

Gesture recognition is a popular topic in computer vision, and it's crucial in a human-computer interface thanks to the path-breaking developments in deep learning. Its applications have a huge impact on lots of fields, which attempts to corporate the gestural channel into Human-Computer Interaction (HCI). In traditional hand gesture recognition (HGR) problem, certain hand movements determined according to selected tasks like sign language translation and robot remote control are recorded as images/videos with the help of the camera. Because of the lack of automated systems

for monitoring handwashing, this project helps to analyze hand gestures at the time of washing. In addition, the solutions proposed for this problem may be beneficial for a variety of other gesture recognition situations. Furthermore, this research may allow for the monitoring of hand hygiene in schools, hospitals, collective events, and the food/production industry.

Our proposals research computer vision approaches together with deep learning models for video analysis. We first examine hand movements as single images. Afterward, we analyze multiple images to figure out the temporal relationship between consecutive actions. We utilize the optical flow method to capture the pattern between hand movements on both single and multiple images. The optical flow helps us explore the movements of pixels in the image plane. As a consequence, we bring a new perspective on this problem by employing the outputs of this approach in deep learning models, which is one of the contributions of this thesis.

As a first step in problem solving, we obtain the motion information of pixels with optical flow in order to understand hand movements deeply. Subsequently, the model inputs are enriched in terms of the information it contains by combining this motion information with the original images before training the models. As the second stage, convolutional neural network (CNN)-based, long short-term memory (LSTM)-based and vision transformer (ViT)- based models are fed with these aggregated inputs including optic information. Thus, the effect of the optical flow approach and several deep learning architectures on the problem is assessed in this manner.

Initially, we train CNN-based models, which are typically used to analyze visual imagery, as baseline models only with original images in RGB videos. We do this in two different types of CNN model, single-frame and multi-frame. Following, using CNN-based models, we assess the effect of optical flow by including both spatial and motion information of pixels in the image plane. Secondly, it is necessary to capture the temporal relationship between previous and subsequent movements to accurately recognize actions during washing. Therefore, we exploit the LSTM architecture fed with original images and inputs including motion information, one of the recurrent neural networks (RNN), to catch this relationship. An end-to-end deep learning model is created by aggregating CNN and LSTM structures to benefit from these spatial, temporal and motional information. In addition, the LSTM architecture is performed uni-directional as well as bi-directional. Finally, transformers, which have left their mark on the world of natural language processing (NLP), have also recently achieved

successful results in the world of image classification and object detection. Thus, we use the architecture of vision transformer (ViT) by using self-attention mechanisms, which is the most important module for this model. Images are divided into patches of a certain size, and gesture recognition is performed by revealing the relationship between these patches with the help of that mechanism.

In this research, the handwashing gesture recognition problem is performed by exploiting RGB videos while someone is washing hands. During washing, the interaction of the two hands with each other is quite high. Therefore, gestures are also difficult to recognize due to the close interactions between the hands and fingers. In contrast to prior research, a variety of deep learning models capable of dealing with this close interaction are proposed. The small amount of labeled data, which has a considerable number of incorrect labels, is another challenge of this study. Experiment results show that the CNN-LSTM architecture fed by inputs containing optical flow information provides the best solution for this problem.

This thesis is organized as follows. Section 2 provides information on prior works on hand gesture recognition and handwashing monitoring in the literature. Section 3 addresses the experiments and established DL models for this problem in detail. Likewise, Section 4 presents the results of the experiments in a comparative form. Finally, we evaluate the findings and offer suggestions for further research in Section 5.

# 2. BACKGROUND

The aim of the study is to analyze the movements in the hand washing process. In this chapter, we look at existing research on hand gesture recognition and monitoring handwashing process from the extensive perspective to the specific. Thereafter, we address proposed some techniques and DL approaches for this problem.

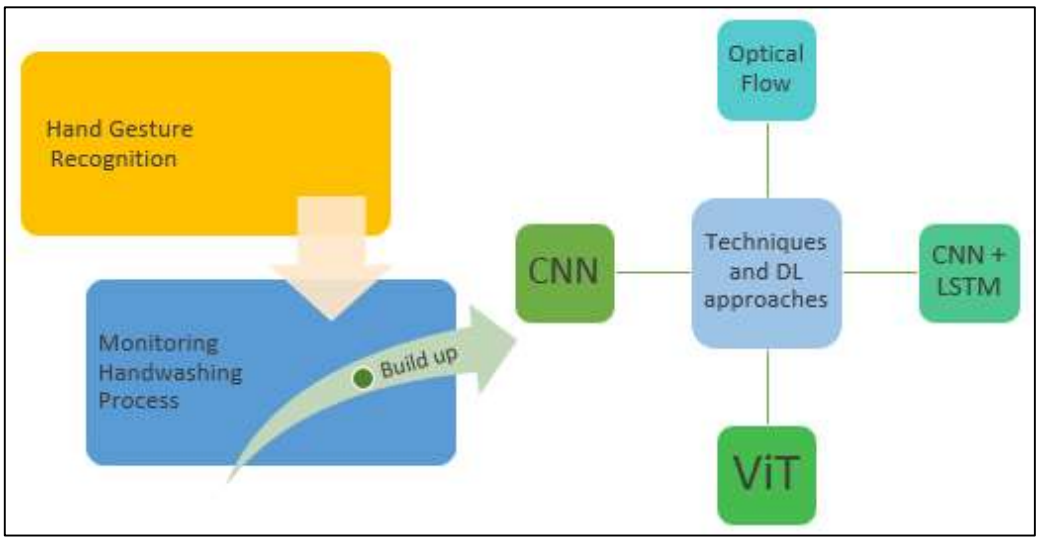


Figure 2.1: The background schema on hand gesture recognition.

## 2.1. Hand Gesture Recognition

Hand gesture is an essential part of performing some tasks as well as a non-verbal language for humans to express themselves more clearly. Hand gesture recognition systems capture and interpret hand movements before performing the required procedure, which is a great role in the human-computer interaction (HCI) system. The way people interact with machines is quickly evolving, and HCI solutions are adjusting to the ever-increasing needs of modern domains. Besides, HGR systems facilitating the life of human beings have been capitalized on many other different areas, such as surgical systems, alternatives to touch-based devices, robotic control, 3D virtual environment, sign language translation, and augmented reality. There are two sorts of hand gestures as static and dynamic. While static hand gestures contain a

well-defined hand position at any given time, dynamic hand gestures are a series of movements that occur over a period of time.

The electrical activity of skeletal muscles is another kind of input for this problem before vision-based hand gesture recognition. The electromyography (EMG) wearable device records these electrical impulses, which has information on the purpose of the movement performed by the human brain. In a study to control the home appliances with ten different hand gestures combine k-nearest neighbor and decision tree algorithms as a classifier for accomplishing gesture recognition by using 3-channel EMG sensors [2]. In [3], [4], support vector machines (SVM) are trained by EMG data to recognize sign language for deaf people. Another study [5] based on superficial EMG signals uses feedforward neural networks after extracting features to distinguish six hand gestures, which respond in about 29 milliseconds in real-time. Similar research compares the results of feedforward neural network (FNN), LSTM, and gated recurrent unit (GRU) on different data formed by forearm muscle signals [6]. CNN, unlike feature-based classifiers SVM, does not require a specific set of features. It automatically decides relevant features from data. Therefore, some studies [7], [8] exploit CNN outperforming compared with other classifiers. Another similar study using CNN fine-tunes the model with transfer learning since it employs the data of many users, which makes it possible to achieve high accuracy with fewer data [9]. To summarize, in hand gesture recognition with EMG data, former studies use algorithms like SVM, decision trees, and k-nearest neighbor. However, end-to-end FNN, RNN, and CNN architectures have been recently preferred more with the development of deep learning. In addition to the data obtained with EMG devices (which can face noise because of the environment, the variation of signals in electrodes due to sweating, and electrode donning-doffing), hand gesture recognition can also be performed on infrared data. A study [10] on the data collected in this way employs LSTM and CNN architectures that are trained to process sequences of 3D spatial information and velocities of fingers. LSTM outperforms CNN in this study.

On the other hand, computer vision-based techniques utilize only bare hands without being colored gloves or sensors. Vision-based solutions are considered to have higher mobility and normality for users than sensor-based alternatives, as well as being more cost-effective due to the use of a single camera. In vision-based hand gesture recognition, whereas early studies exploit RGB images, depth and RGB images are begun to use in HGR with the release of the Kinect camera, which can provide high-

quality deep images. Complex backgrounds and lighting variations are dealt with using depth pictures. The pipeline of these studies is that some features (hand location, hand shape, direction) are first obtained by feature extraction on depth or RGB images (edge detection, trajectory generation), and the hand gesture is then recognized by classifiers fed with these features. Generally, hidden Markov model, SVM, ensemble algorithms, and artificial neural networks are used as a classifier [11].

A study that classified seven different static hand gestures with different backgrounds and lightings performs the recognition using CNN [12]. In [13], Hung *et al*. start by removing unnecessary background information from the image using skin color detection and morphology, and then feed modified AlexNet [14] and VGGNet [15] (CNNs) models. Another similar work [16] has a deep CNN structure proposed on data with two different backgrounds, simple and complex. The model with simple background has better result on test data. CNN models recognize multiple hand gestures after depth images provided by Kinect are first processed with skin color [17][18]. Another study [19] categorizes video sequences of hand motions using a long-term recurrent convolution network. Unlike a standard long-term convolution network, tiled image patterns, which represent the entire gesture-based video sequence within a single frame, are utilized instead of multiple frames as they increase the computational complexity. In [20], the hand edge images are obtained after denoising and edge detection. Subsequently, two-stream CNNs, which have the same number of layers and parameters but different weights, are fed with hand gestures and edge images. These two-stream CNNs are combined with a sum fusion way, hand gestures are classified after a fully connected layer. Sign language recognition, which is an important research field in HGR, helps us to communicate with hard of hearing, and deaf people. Word-level dynamic sign language recognition with 17 classes on deep images collected by phone camera first removes background on deep images. It uses pre-trained MobileNet v2 [21] (CNN) for feature extraction on frames first and then LSTM network for temporal extraction [22].

Generally, two types of data acquisition are performed in HGR: sensor-based or vision-based. While muscle activities are recorded with wearable EMG devices for a sensor-based system, either colored gloves or RGB/RGB-D cameras are used for vision-based data to collect data. EMG devices are regarded to be inappropriate for hand hygiene monitoring because of the difficulties in clothing and taking off in each wash, time constraint, contact with water and disinfectant, and calibration. In other

words, sensor-based systems are very unnatural and more restrictive besides being costly. Furthermore, as the incident involves hand washing, gloves are not required. Therefore, it is envisaged to establish a vision-based system on the images obtained with the camera.

### 2.1.1. Monitoring Handwashing Process

Recognizing and monitoring the necessary hand movements in the handwashing process by an automatic system is an interesting problem for the computer vision world. However, it is very difficult to distinguish the actions even when viewed with the human eye because of the close interaction of the hands. For this reason, deep learning methods, which can provide high success in difficult problems in the world of computer vision, have been used in the proposed models and previous studies. In the literature, most studies have focused only on checking whether a person disinfects their hands, not on how they should disinfect or the way they disinfect. Hand hygiene monitoring systems that are electronic or electronically aided are being developed in large numbers. These studies include automated counting systems (ex., counters in pump bottles), video monitoring, and a completely automated monitoring system. They generally utilize a wearable/mobile component/camera as a way to record all hand hygiene opportunities and provide a feedback or reminder system [23]. Another similar research makes use of a camera that is placed above the sanitizer, and people who do not use alcohol-based sanitizer are detected by face detection with the help of the camera [24]. The purpose of another study using convolutional neural networks is to detect the usage of hand-hygiene dispensers. They exploit depth images with human annotations. A body part heatmap and foreground mask are obtained by giving the single depth image to the pose network (CNN) and segmentation module. They constitute augmented input by stacking of depth image, body part heatmap, and foreground. The ResNet-152 [25] receiving the augmented input outperforms compared with other CNN architectures [26,27].

In [28], 3D CNN is used to monitor anesthesiologists' hand hygiene in an operating room. They begin by detecting and cropping the region of interest (ROI) of anesthesiologists' upper body. The ROIs are smoothed out with a temporal smoothing filter. The ROIs are then fed into a 3D CNN, which is classified into two categories: rubbing hands and other movements. The other study [29] uses the 2D ResNet-50,

LSTM, and temporal relational network (TRN) [30] in order to classify four different hand hygiene actions by using RGB images taken at three different angles. The ResNet-50 + TRN model outperforms compared with other models. [31], where 2D CNN and LSTM architecture are used together, has worked on a more comprehensive dataset with 11 different classes in the handwashing process. After CNN performs feature extraction offline on single frames, the obtained features are trained to extract temporal information between frames in the LSTM architecture.

This planned thesis, it is aimed to enrich the frames by obtaining the motion information of the hands with the optical flow method, and it is thought that better results would be obtained in solving the problem with this approach. When we look at the studies in the literature, neither the information of the optical flow is employed with the original images, nor the transformer-based deep learning model has been found in the field of handwashing gesture recognition.

## 2.2. Techniques and DL Approaches

In comparison to standard computer vision (CV) techniques, DL approaches necessitate less professional analysis and fine-tuning, and could make use of today's systems' huge amounts of image/video data. Besides, as DL algorithms can be retrained using a specific dataset for any case, they offer greater flexibility. Another issue with the conventional method is that it involves determining which features in each image are essential. Because recognition or detection operation is achieved thanks to features such as edge, corner, and color. Feature extraction becomes more challenging as the number of classes to be categorized grows. A long trial and error process results since personal judgments are used in traditional methods to find out which features best describe distinct classes of objects, which require the management of a large number of parameters that need to be fine-tuned. In contrast, end-to-end DL approaches find key patterns in image classes and automatically resolve each object's most revealing and salient features associated with its particular object class. Increases in computer power and data available for training neural networks have resulted in a considerable improvement in the ability to recognize objects. As a result, we propose end-to-end deep learning models for the problem in this study. The technique and deep learning solutions to be leveraged in the proposed models are examined below.

### 2.2.1. Optical Flow

Understanding video structures is one of the most important aims of computer vision. Videos have temporal information as well as spatial information as in images, which separate them from images. In other words, video is the collection of spatial information in a certain order. This extra temporal information is what makes recognizing videos difficult. Therefore, the optical flow method can be useful in making sense of the temporal flow in videos.

Optical flow is an approach that predicts how each pixel in the image plane moves from one moment to the next. In simple terms, optical flow describes the displacement vector of pixels between two consecutive frames. The main idea behind it is that it attempts to project how pixels move across the screen over time by assuming a direct relationship between object movement and changes in intensity in a sequence of images (a brightness constancy). One of the biggest advantages of optical flow is that it does not require prior knowledge of object appearance because of being pixel-based method. Its theoretical basis is that a function describes the density of pixels in a frame with the location of pixel and the order of frame. Subsequently, this method tries to find the motion vector by assuming that the intensity of the pixels of a certain object does not change in a time interval. In Figure 2.2, the theoretical basis of the optical flow method is shown below.

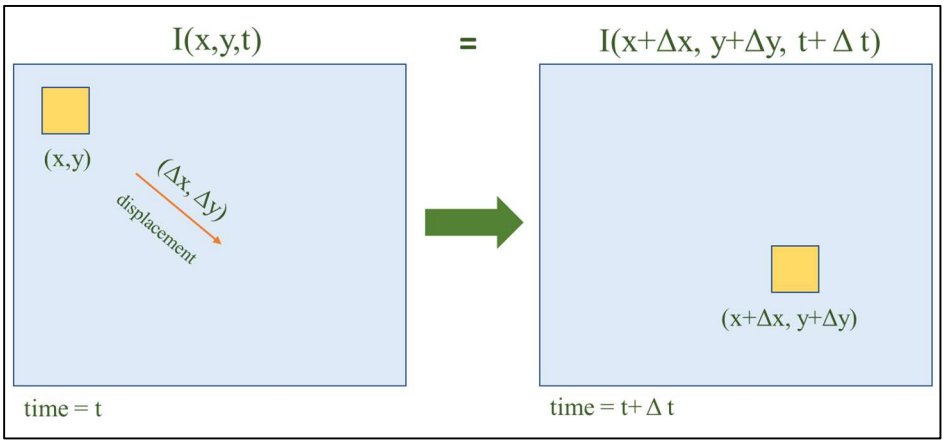


Figure 2.2: The theoretical basis of the optical flow

Using motion information between consecutive frames in classifying videos yields good results. Simonyan *et. al.* [32] propose a two-stream CNN structure for action recognition in videos. First stream of that model catches the spatial relationships in single frames. Conversely, the second stream is exploited for temporal recognition by utilizing multi-frame optical flow displacements between several frames, which makes the recognition easier. Finally, the actions are classified with the fusion of the class scores from two different CNN streams. Similarly, the two-stream networks in [33] are designed with appearance and motion as separate inputs to improve sign language recognition system. In contrast to [32], sum fusion approach is adopted at the fifth convolutional layer for fusing the two stream networks. This approach calculates the element-wise sum of two feature maps by putting the appearance and motion information in the same spatial location, without adding extra filters to put the feature maps together. Next, the feature map obtained by sum fusion feeds bi-directional recurrent neural network to learn the temporal dependencies after some 1D convolutional and pooling layers.

The fact that motion information feeds the models as extra information in deep learning networks has given high results in prediction. In addition, motion information is usually utilized separately from the original images when looking at previous studies. Therefore, the original images and motion information are used in the same input in this study.

### 2.2.2. CNN and LSTM

The de-facto method for visual data is CNN, which can recognize, classify and detect picture features. The architecture of CNN is intended to replicate the connections between neurons in the human brain and the visual cortex's functions. Unlike conventional methods, CNNs do not require any feature extraction. The system automatically performs feature extraction using the convolution of images and filters and transfers these invariant features to the next layer. CNNs consist of one or more convolutional layers, pooling layers, and one or more fully connected (FC) layers like a standard multilayer neural network. One of the most significant advantages of CNNs over FC networks is that they require less training and parameters. CNNs capture spatial information of pixels by using a large number of filters and layers, which allows

it to solve recognition/detection problems. However, using only single frame for inputs being sequential information such as videos affects the prediction success. Because in this case, the connection between the previous frames and the next frames also needs to be resolved. From a perspective, a certain number of consecutive frames can be utilized as input instead of using only a single frame. It can increase the prediction power of the model because the input contains much information about the same class. But it does this again by dissolving spatial connections, not temporal relations. Therefore, the inability to resolve the temporal relationship in solving sequential data in computer vision greatly limits the success of the model. This is due to the complexities of including time in the equation. Leveraging different viewpoints or model architectures can boost model flexibility and power. Another possible solution can be 3D CNNs. Unlike 2D CNNs, they are used to find patterns across 3 spatial dimensions: depth, height, and width. 3D CNNs are more appropriate for extracting temporal-spatial features as they can relate this depth pattern to the temporal context in videos. On the other hand, as 3-D CNNs fail to capture long-term spatiotemporal dependencies, they are only employed as low-level feature extractors during many short intervals.

In temporal data, giving the meaning to the next situation can only be done by remembering the previous situations. Traditional neural networks cannot do such a thing, and this is a major shortcoming. A recurrent neural network (RNN), on the other hand, is developed to address such problems and offer solutions to them. The basic idea behind a recurrent neural network is to use sequential information. In a typical neural network, all inputs and outputs are supposed to be independent of one another. In contrast, RNN does the same task for each state in a timeline based on previous outputs. That is, it carries "memory" that collects information about what has been calculated so far. In other words, it makes decisions using the current input and the output learned from the previous input. Gradients are backpropagated through layers and also through time. All the prior contributions are had to be summed up until the current one in each time step. Thus, RNNs have a vanishing gradient problem due to the long series of multiplications of small numbers and this issue causes the learning process to degenerate. In other words, it makes difficult to learn some long period dependencies. In addition, if the gradient multiplications grow rapidly to infinity, the change value in training will not be number (NaN) due to the unstable process, which

is called exploding gradient problem. For these reasons, training RNNs is quite difficult. It is unable to process lengthy sequences.

LSTM being a special type of RNN is designed to better store information, eliminating the short-term memory problem of standard RNN. There are three gates in LSTM architecture: input, output and forget. Forget gates decide what information should be left over the network. Therefore, LSTM allow to have short-term and long-term memory. Input gate updates the Cell State. It is decided whether to update the previous and current information according to the result of the sigmoid process. The output gate determines the next cell's input and is also used to make predictions. LSTM also solves the vanishing gradient problem of RNN by using a unique additional gradient structure that includes direct access to the forget gate activations. LSTM is a very convenient algorithm for predicting and categorizing time series. LSTM makes a prediction for the last situation by remembering the information from the previous time positions in the specified time step. Since videos have a temporal structure, LSTMs can be used for them. However, videos also have spatial information as they are composed of images. Therefore, LSTM architecture can be combined with CNNs that can perform spatial inference very well. Each frame is represented as a one-dimensional vector after feature extraction performed by CNNs. Then, the temporal relationship between these obtained vectors is tried to figure out with LSTM at a certain time step. Prior studies related to these described architectures are explained in Section 2.1.

### 2.2.3. Vision Transformer (ViT)

Transformers, consisting entirely of attention mechanisms, have surpassed other natural language processing (NLP) models after their release. Furthermore, thanks to its straightforward design, transformers enable to process multiple modalities like images, videos, text, and speech. Transformers have begun to be used in computer vision tasks as it has tremendous success for NLP tasks. Transformers make use of self-attention mechanisms to effectively deal with the limits posed by inductive convolutional biases. Transformers, unlike convolutional networks, require low inductive biases in their design. Nevertheless, previous attempts to use fully-attention networks in large-scale computer vision failed poorly. The previous proposed self-

attention methods, which is the most significant module in transformers, were not appropriate for large images as the complexity depended on the number of pixels. Later, a new perspective is brought to transformers and vision transformer (ViT) [34] architecture is developed as a novel approach. ViT first divides images into multiple fixed-size patches, which can be considered just like words. As seen in Figure 2.3, each patch can be thought of as a word and computer vision tasks can be performed by investigating the links among these patches. Self-attention module in ViT enables the global integration of information throughout the entire image. Positional embedding is added as an input after patches are first embedded linearly. In other words, adding the learnable position embeddings to each patch provide the model to learn about the structure of the image by investigating relationships and interdependence between patches. Finally, transformer structure is fed after obtained embeddings of patches. While ViT achieves competitive results on medium-sized datasets such as ImageNet, it has been shown to either beat CNNs or achieve similar results when applied to larger datasets. In addition, ViT outperforms the state of art CNNs in terms of computational efficiency.

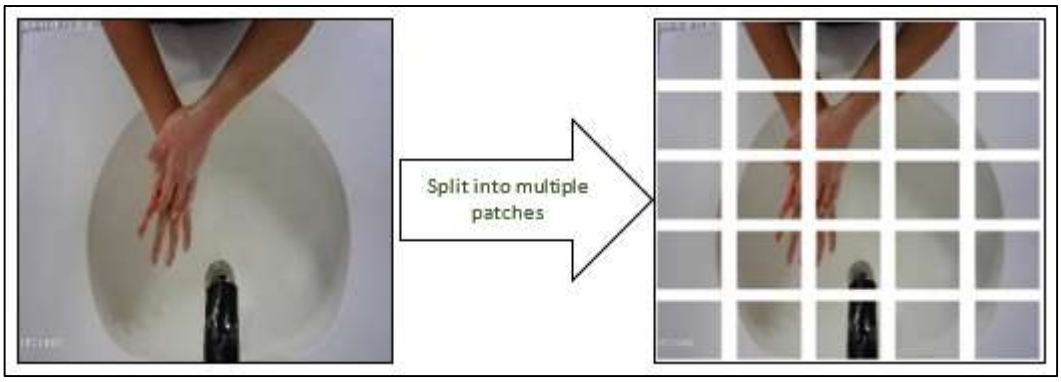


Figure 2.3: Converting the original image to fixed-size patches.

Although ViT's applicability to hand gesture recognition, in particular, has yet to be thoroughly researched, there are few studies utilizing this method. In [35], transformer-based architecture is proposed for the dynamic hand gesture recognition task on depth images and surface normals. After frame-level feature extraction with ResNet, temporal aggregation is calculated by a six-layer transformer encoder, which achieves state-of-the-art results. Another similar study [36] uses a multi-head fused transformer for facial action unit detection. The suggested structure is made up of two

pipelines, each with two input modalities: RGB and depth images. There are two components in the framework as fusion transformer and transformer encoder. They are embedded after these input modalities are first extracted by ResNet as a backbone network. These embeddings of modalities are fused through a fusion transformer module to obtain the fusion futures. Then following four transformer encoders in each pipeline, features are encoded. Lastly, the final loss is calculated by summing the losses of two pipelines. Montazerin *et. al.* [37] recognize hand gestures with vision transformer-based model from high density EMG signal. Firstly, they convert a cropped portion of the original signal (like signal graph) to small patches. Secondly, these patches proceed a patch + position embedding layer and a class token is prepended to them after linear projection. After that, they're fed into the transformer encoder. Even though there are a few vision transformer-based studies in the field of hand gesture recognition, no study has been found in the field of monitoring handwashing.

# 3. HANDWASHING MONITORING

Monitoring hand washing movements in hand gesture recognition has gained a special importance in this pandemic period. Successful hand washing is accomplished with hand gestures determined by the WHO. Even when observed with the naked eye, it is difficult to distinguish between the activities due to the close contact of the hands and fingers in handwashing monitoring. For this reason, in the proposed models, end-to-end deep learning methods that can provide high success in difficult problems in the world of computer vision are used. The main goal of the developed system is to correctly recognize hand movements having temporal relation. In Figure 3.1, the problem is briefly schematized.

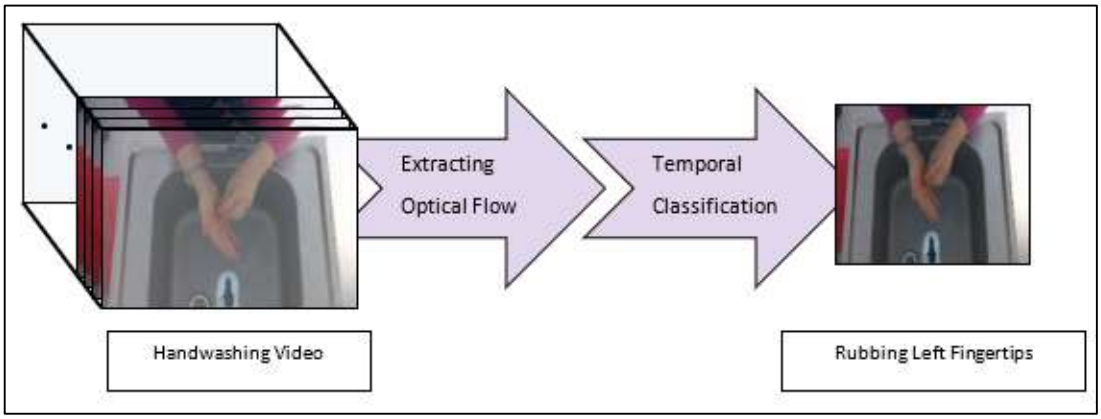


Figure 3.1: The definition of problem for handwashing recognition.

As a result of this scheme, each frame in videos affects what the next action is. As the baseline model, the CNN-based model, which recognizes only by looking at the single frame, is exploited. In contrast, deciding what the action is by just looking at a single frame, weakens the predictive power because of the dependence between consecutive frames. Two different ways are employed to capture or reveal this dependence. The first one is to enrich the model input in terms of information by subtracting the displacement of the hands during the movements. The displacement of the pixels on the hands is obtained by the optical flow approach. The second one is to run an end-to-end model with a deep learning approach that can capture this temporal dependency between actions. This end-to-end model is built using successful CNN and LSTM architectures together. In addition, transformer-based models are also state-

of-art model in the computer vision after making a splash in the NLP world in recent years. Therefore, the success of transformer-based models on the problem is also measured compared with CNN's.

## 3.1. Dataset

The dataset, that is, all the videos, was prepared by a private company. Although there is some data set in the literature, there is not such a comprehensive data set on this subject. The hand washing process was recorded with IP cameras on two different sinks while someone was washing their hands with water and soap. In addition, handwashing videos of different people were recorded under different lighting conditions and different backgrounds. Besides, the movements in the handwashing process (determined by the WHO) were made by individuals in a completely random order. The videos are usually recorded at about 12 frames per second. In addition to the diversity of the movement sequence, the videos have different durations due to the fact that there is a person-dependent movement. Frames of different classes taken from the videos are shown in Figure 3.2.

Figure 3.2: Some frames belonging different classes in videos.

All frames in all videos have been extracted and labeled frame by frame by dividing the videos into parts according to movements. There is a total of 11 different classes determined according to which part of the hand is cleaned, and some classes have a right-handed and left-handed distinction. In the proposed models, the frames are classified either individually or together with neighboring framers. There are also a lot of incorrect labels in the data as an outlier. In order to avoid an unbalanced data set, the data is created by selecting equal number of samples from each class. For

training and test, 4000 samples from each class are chosen at random and split into 85% and 15%. These frames feed the deep learning models individually or as a time series. They are also stacked with extra information to explain the movement better.

## 3.2. Pre-processing

Pre-processing is the first step in this problem. 1920x1080 sized RGB frames are first extracted from the videos. These frames are resized to 224x224 to feed the models. On the one hand, in models where optical flow information is not used, only 224x224 RGB or grayscale frames are exploited for inputs. On the other hand, since the information on displacement of the hands is thought to help predict movement correctly, it is intended to be used in model inputs as extra information. Therefore, after some preprocessing, this information is acquired and implemented into the input. The following describes how the displacement of hand pixels is obtained and how it is used in input is explained.

### 3.2.1. Displacement of Hand Pixels

Each frame in the handwashing process does not constitute a movement by itself. Handwashing movements occur as a result of sequential actions. That is, hands constantly repeat the handwashing gesture of each different class in certain patterns. Thus, in order to accurately predict the handwashing gesture, it is necessary to catch the link between the frames and before/after the frames themselves. The displacement vectors of the hands between consecutive images can be used as a way to extract the pattern between these frames. These displacement vectors are got by the optical flow approach in the computer vision world.

Optical flow gives information about the movement of all pixels in the image plane when they pass to the next frame. In other words, each pixel has actually a displacement vector including angle and magnitude information. The theoretical basis of it, that is, how this displacement is calculated, is explained in section 2.2.1. There are two types of optical flow, Sparse and Dense. On the one hand, sparse optical flow calculates the motion vector for the particular set of objects on the image, such as detected vertices. Therefore, it requires some preprocessing for feature extraction.

However, using only the sparse optical flow method means that motion information cannot be obtained about pixels that are not contained in a particular object. On the other hand, the displacement vector is computed for each pixel in the image using the Dense optical flow method to avoid this constraint. Motion vectors are acquired with the Farneback algorithm [38] from dense optical flow methods. Since the Lucas-Canade [39] method has only a first-order Taylor expansion, the linear approximation is used for neighboring pixels. But the Farneback method improves the accuracy of approximation with second order values. In other words, the idea behind the Farneback method is to predict some neighbors of each pixel using the polynomial approximation. In addition, A pair of RGB consecutive photos are converted to grayscale because the Farneback method requires 1D input.

The output of this algorithm, the displacement coordinates obtained ($\Delta x$, $\Delta y$), is converted into polar coordinates with the help of a function as angle and magnitude. In Figure 3.3, the displacement vectors of the pixels for the next move are illustrated.

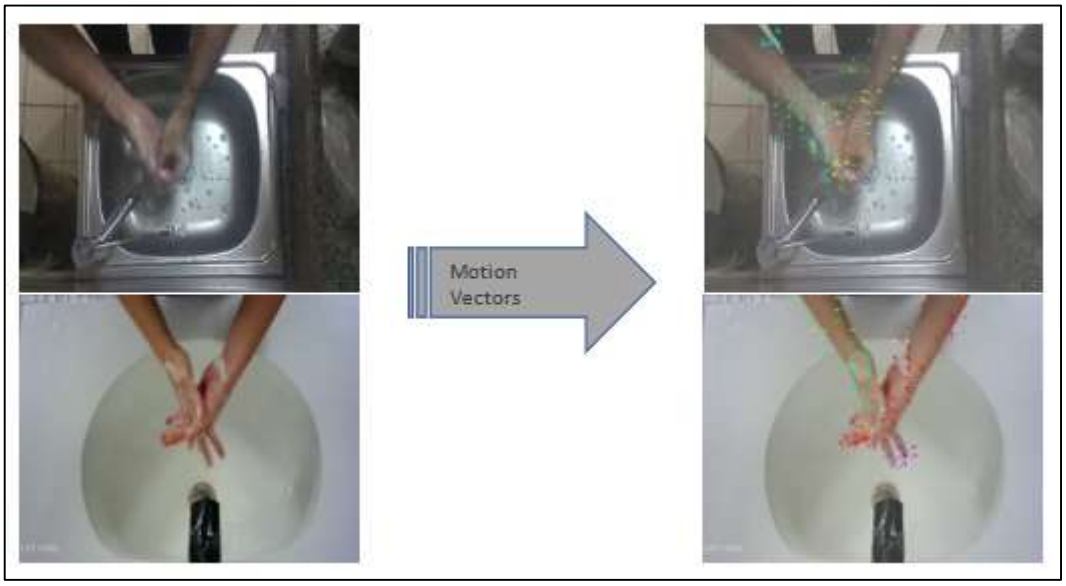


Figure 3.3: The motion vectors of pixels.

The optical flow output can also be encoded in the HSV color space. Here, while the saturation value is kept constant, the hue value is encoded with the angle of the pixels, and the value is encoded with the magnitudes of the pixels. Next, the HSV format is converted to RGB color space for the proper visualization. In Figure 3.4, the optical flow of a pair of sequential images is shown in RGB format.

Considering the angle and distance, which are the optical flow outputs, the angle values of the pixels are more effective in making sense of the sequential movements. Because the angle actually gives us the information in which direction the hand regions move. However, the only magnitude does not make any sense, it has a lot of variability even in movements belonging to the same class. Therefore, the angles of the pixels are visualized and analyzed. When part (c) of Figure 3.4 is examined, a rather noisy structure is seen in the area outside the hand region because of the change in lighting, reflections, shadows, motion blur. Therefore, these angle values are stacked with the inputs without eliminating the background noise first. Subsequently, these angle values are masked using a hand mask created specifically for the hand region. Finally, only the angle values of the pixels in the hand region are put into the input. Two different methods are used to remove the background by creating a hand mask. These methods are described in the section just below.

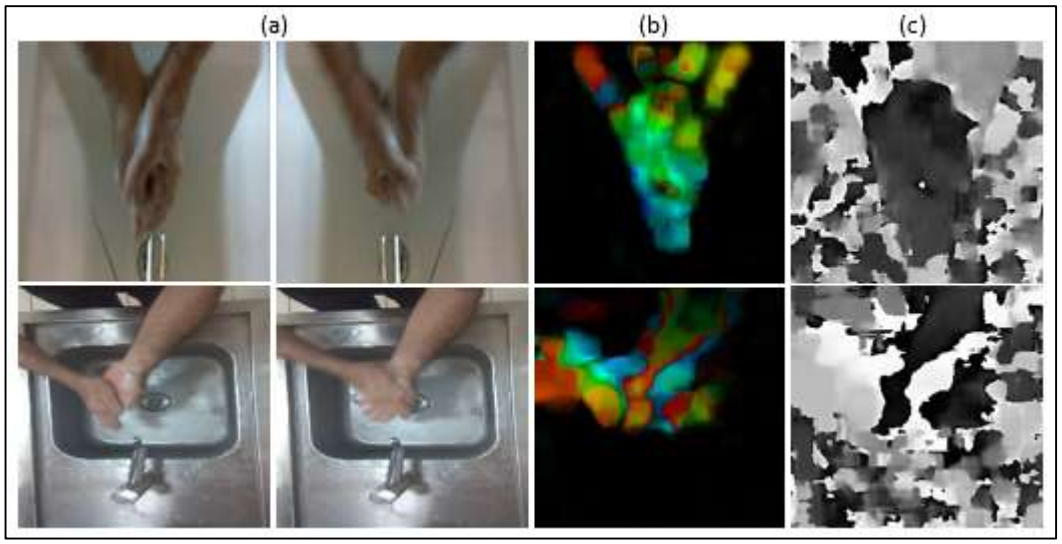


Figure 3.4: a) A pair of consecutive images. b) Visualizing the optical flow output with RGB color space. c) Normalized pixel angles between 0 and 1.

### 3.2.2. Removing Background

The background noise should be minimized so as to use the angle values of the hands during the movement. Therefore, a mask is created that can detect only the hand area. This mask creation process is attempted in two different ways. To begin, the hand field is determined using color-based skin detection. In this method, first the RGB

frame is converted to YCbCr color space, which can better identify the skin pixels. Each pixel in image is represented by **px** = [*Y, Cb, Cr*]. The classification of skin colors is achieved at specified threshold values in this color space in order to capture skin colors. These threshold values are computed empirically to get rid of the non-skin pixels immediately and are [85,135] and [133,180] for *Cb* and *Cr* values, respectively. As can be seen in the (a) part of Figure 3.5, while some non-skin pixels are selected due to light, shadow, and reflection reasons, several skin colors may not be chosen because of the soap, shadow, and motion blur. Thus, morphological image processing techniques are utilized to struggle with these problems. Erosion, which can eat away non-skin colors, first minimizes the selection of pixels outside the hand. Next, dilation operation, which can grow the skin pixels increases the choosing of pixels both in the interior and on the edges such as fingertips. The results of morphological operations are indicated in part (b) of Figure 3.5. Finally, noise is removed with median filter protecting the edges to achieve a smoother mask.

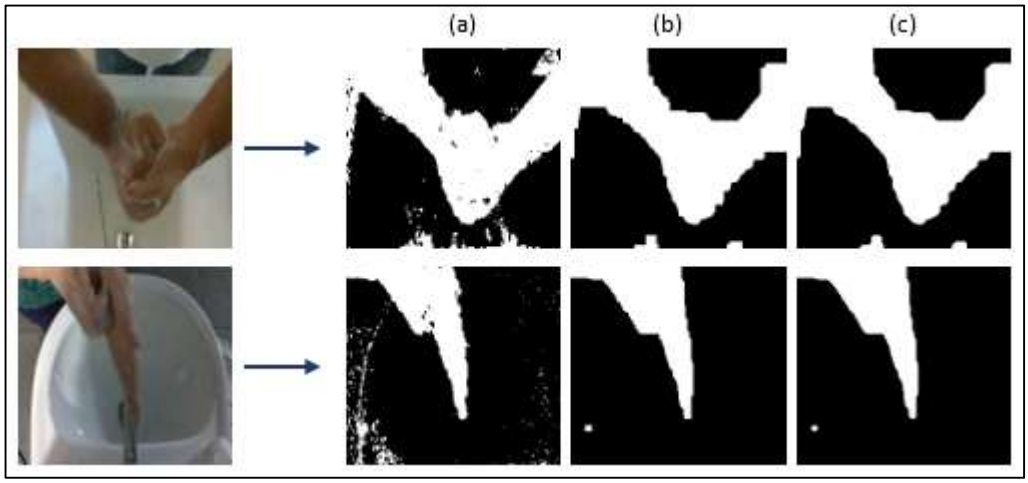


Figure 3.5: The stages of color-based skin detection. (a) Classifying pixels on YCbCr color space. (b) Morphological operations. (c) The application of median filter.

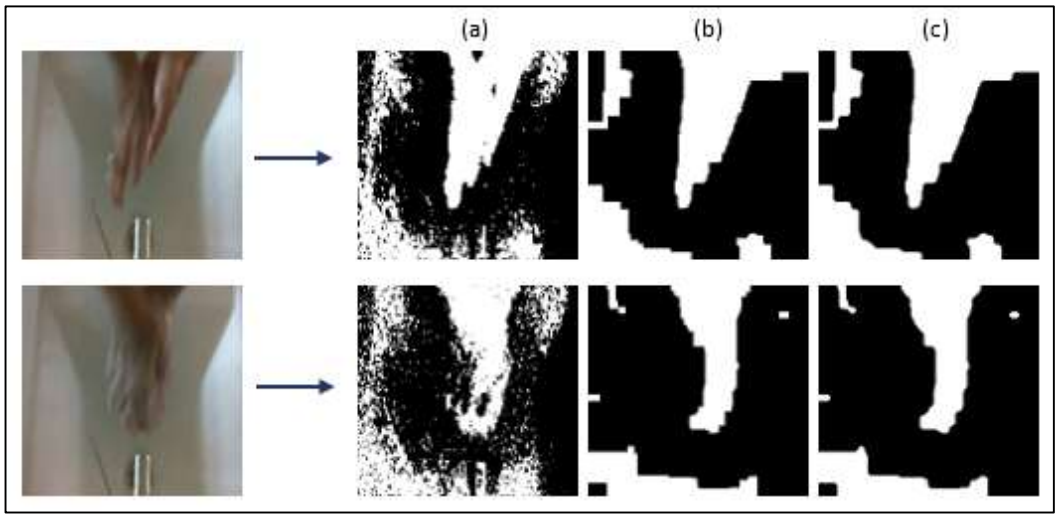


Figure 3.6: Some poor results of color-based skin detection. (a) Classifying pixels on YCbCr color space. (b) Morphological operations. (c) The application of median filter.

This strategy produces good results in some cases whereas it produces poor results in others where environmental elements (light, etc.) are not suitable. This situation is clearly illustrated in Figure 3.6 with some examples. Hence, a second approach is recommended for the detection of the hand region. The second strategy makes use of optical flow outputs when creating a hand mask.

The magnitude values in the optical flow are examined on the image plane since the image movement is carried out by hands. After examination, it is clearly observed that pixels being significant magnitude values are generally in hand regions. Therefore, to classify these pixels, 1 is chosen as the threshold value for the magnitude. As in the first method, morphological operations are applied again to reduce the noise. After this stage, there might still be a set of non-skin pixels selected. Here, the contour selection having maximum area is performed to do away with these kinds of pixels completely. Finally, this biggest contour structure is converted from concave to convex for a smoother state. This second strategy is outlined in Figure 3.7.

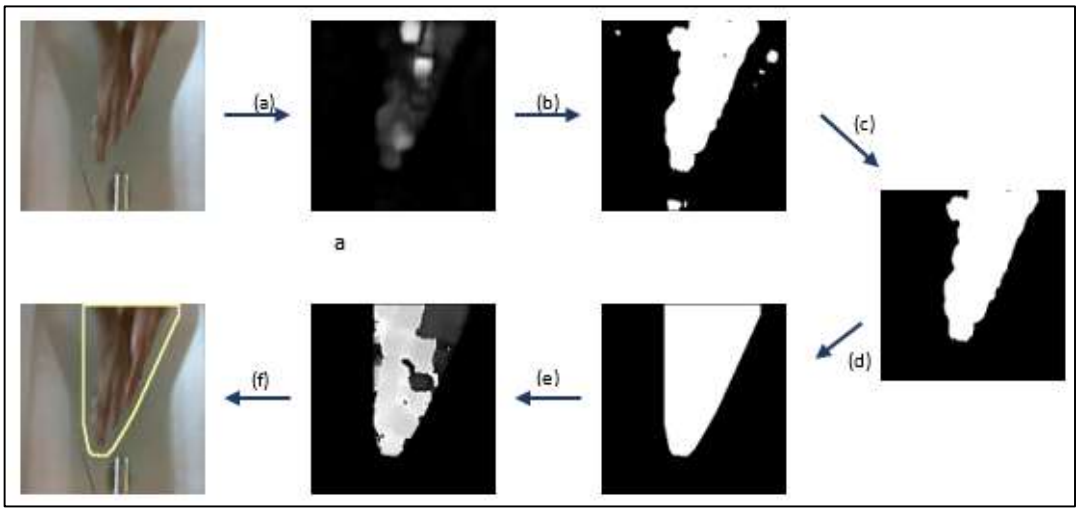


Figure 3.7: The stages of magnitude-based second strategy. (a) Magnitude values normalized between 0 and 255. (b) Filtering pixels higher than threshold. (c) Finding the biggest contour. (d) Concave to convex structure. (e) Applying hand mask to angle. (f) Visualization of hand mask on original image.

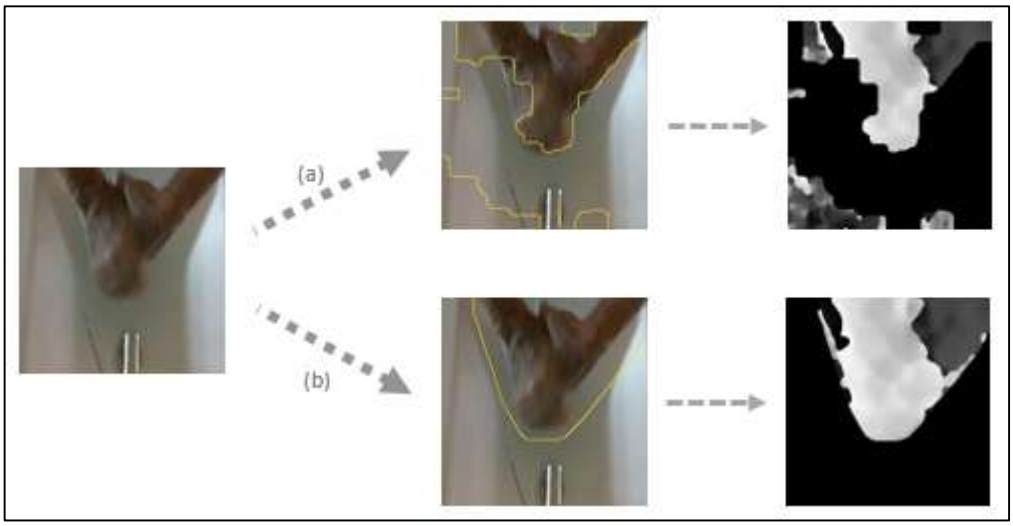


Figure 3.8: (a) The hand mask obtained by first approach. (b) The hand mask obtained by second approach.

By applying the hand masks created by both methods to the angle plane of the image, it is aimed to exploit only the angles of the pixels belonging to the hand region. Angle values masked by the two methods are also included in different model inputs. However, when Figure 3.8 is analyzed, it is understood that the second approach gives better results even under different environmental factors. The inputs are created by stacking the masked angle information together with the original frames to run the deep learning models.

## 3.3. Deep Learning Models

The next step after preprocessing is to build up end-to-end deep learning architectures. Three distinct deep learning methods are used to recognize hand gestures from handwashing videos: CNN-based, LSTM-based and ViT-based. These models are fed both pure original images and inputs that include motion information after preprocessing. In addition to the impact of the optical flow used as extra data, the effect and success of various models on the problem are assessed.

### 3.3.1. CNN

The first model that comes to mind to classify handwashing movements in videos is CNN. Two types of CNN models are employed as single-frame and multi-frame. Single-frame CNN handles each frame individually, regardless of the temporal relationship of the video. We propose the reduced-depth version of the DenseNet [40] model, one of the important CNN architectures. This structure helps to alleviate the vanishing gradient problem, increases the reuse of features by using channel-wise concatenation between blocks and also reduces the number of parameters considerably. This network consists of four dense blocks including only convolution layers and three transition blocks containing convolutional and pooling layers. The transition layer uses 1D filters to reduce the depth of feature map from the previous dense block by 50% on a per-channel basis. Thus, both the computational cost is reduced and the aggregated information from the preceding layers is kept under control. In addition, there are two hyperparameters that must be decided to tune architecture when building the model: the number of convolutional layers in dense blocks and growth rate. These parameters are tuned according to the dataset and problem complexity for obtaining the optimum model which can produce the best results. In Figure 3.9, the architecture of the proposed shallow DenseNet model is illustrated.

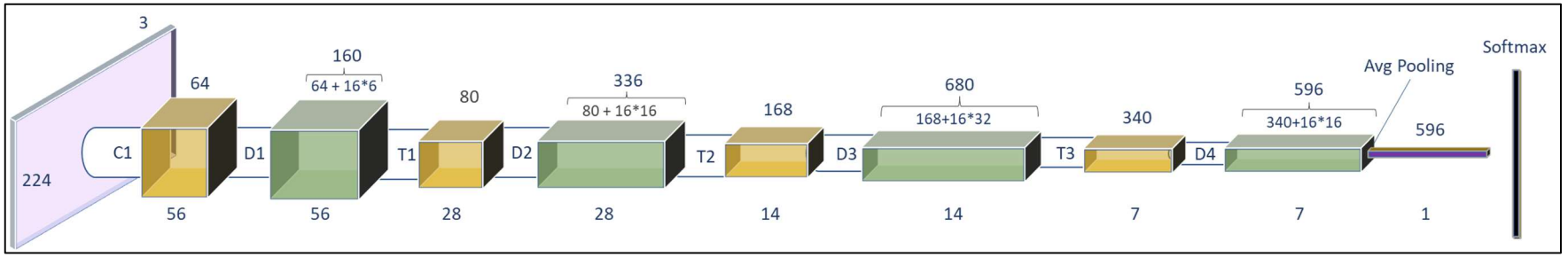


Figure 3.9: The architecture of a shallower DenseNet.

The number below each block indicates the width and height of the tensor, while above it indicates its depth. In fact, in addition to concatenating the information from the prior block, this depth is specified by multiplying the growth rate by the number of layers in that dense block. The baseline model's growth rate is 16, and the number of layers in dense blocks is 6,16,32, and 16 in that sequence. This model is fed with 224x224 RGB frames and has about 2.55 million parameters.

This architecture is fed only with original single RGB images as a baseline model. Then, different input structures are created with pre-processing operations and the contribution of the information included in the input on the model is tested. Firstly, normalized angle information of the previous and next movement of all pixels in the frame is agglomerated with the RGB frame to create a complex input. The motion information of all pixels obtained by taking the weighted average of the angle information two before and two after the centre frame is stacked with that image. Here, the optical angle value that is closer to the target frame is multiplied by 0.6 weight, while the far one is multiplied by 0.4. Subsequently, the angles are masked with the hand area obtained by two different methods to eliminate the background noise at them.

Table 3.1: Information of input and model variables belong to single-frame CNN

| Single Frame Models | Color | Motion Information | | Masking Method | | Channel |
|---|---|---|---|---|---|---|
| | | Angle | Norm | 1st | 2nd | |
| SF_RGB | RGB | | | | | 3 |
| SF_RGB+Ang | RGB | ✓ | | | | 5 |
| SF_RGB+ScaledAng | RGB | ✓ | | | | 5 |
| SF_RGB+MaskedAng-1 | RGB | ✓ | | ✓ | | 5 |
| SF_GRAY+MaskedAng | Gray | ✓ | | | ✓ | 3 |
| SF_RGB+MaskedAng-2 | RGB | ✓ | | | ✓ | 5 |
| SF_RGB+Ang&Mag | RGB | ✓ | ✓ | | ✓ | 7 |

Finally, besides the masked angle information, the magnitude information is also stacked with the RGB frame as a distinct input. The number of filters of the

convolution layer before the first density block of the model, in which angle and magnitude information are used together, is gone up and the growth rate is risen from 16 to 17 due to increased input complexity. All other single-frame CNN models have the same variables. The aim of creating diverse input structures is to increase the predictive power of the model on videos by including motion information in the frame. Input structures are explained in detail in Table 3.1. Motion information about the previous movement of the target frame is added as a channel just before it, and movement information about the next movement is added as a channel directly after it, as seen in the Figure 3.10.

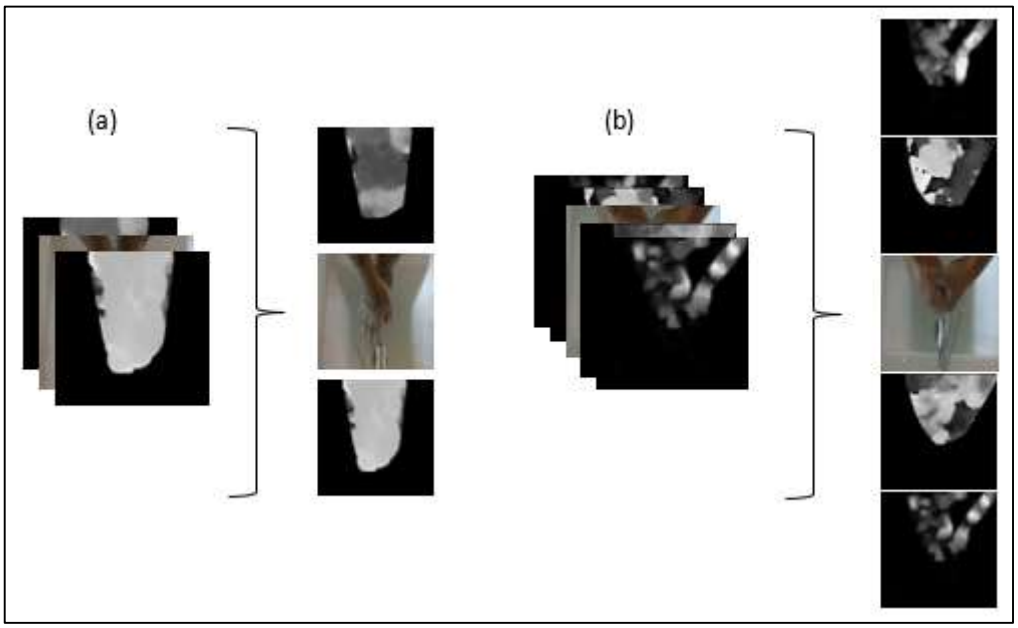


Figure 3.10: Some single-frame CNN input structures. (a) SF_RGB+MaskedAng-2 input being 5-channels. (b) SF_RGB+Ang&Mag input being 7-channels.

On the other hand, considering the single-frame CNN model, making classification by looking at one frame alone may not offer a very generalizable solution because it ignores the temporal relationship caused by the sequential nature of the videos. That's why multi-frame CNN taking consecutive frames to classify the center frame according to previous and next frames is proposed. Besides, not only the models fed by the original consecutive images, but also the inputs to which the motion information is again included feed the different multi-frame models. Multi-frame CNNs are trained with inputs consisting of only original images, generated in different numbers and in different color spaces. Later, these models are also retrained with

complex inputs involving only the angle information masked with the hand region in two different ways. This masked angle information is placed as an extra channel between two corresponding consecutive frames. This case is visualized in Figure 3.11.

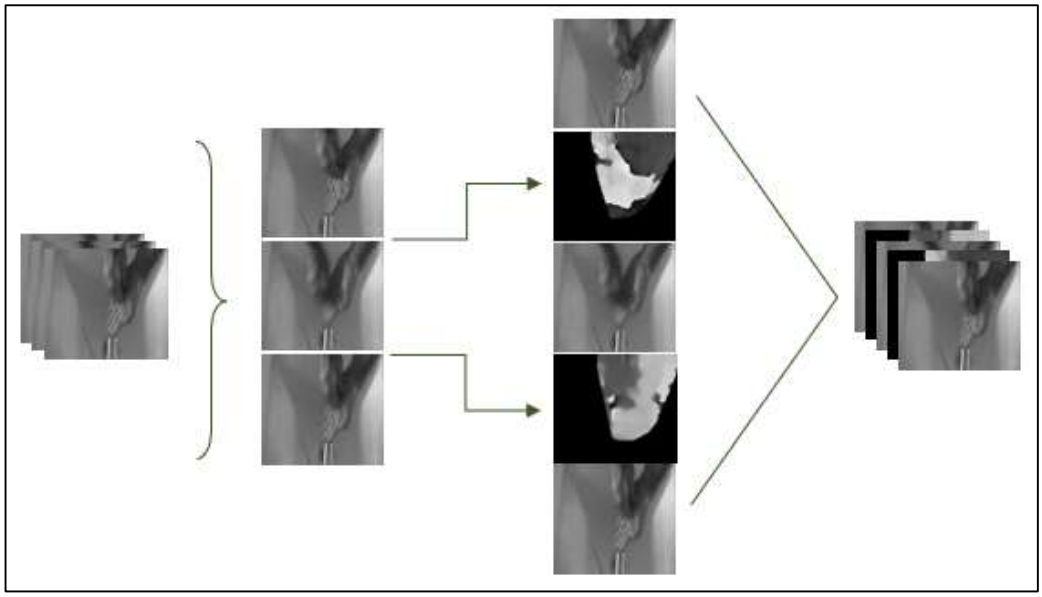

Figure 3.11: Incorporation of masked angle information into between relevant frames, from 3-channel original frames to 5-channel complex input.

Table 3.2: Information of input and model variables belong to multi-frame CNN.

| Multi Frame Models | Color | Input | | Channel | Filters in 1st Conv | Growth Rate | Parameter (Million) |
|---|---|---|---|---|---|---|---|
| | | Frame | Angle | | | | |
| MF_GRAY-1 | Gray | ✓ | | 3 | 64 | 16 | ~2.55 |
| MF_GRAY-2 | Gray | ✓ | | 5 | 96 | 16 | ~2.6 |
| MF_RGB | RGB | ✓ | | 9 | 128 | 17 | ~3.1 |
| MF_GRAY +MaskedAng-1 | Gray | ✓ | ✓ | 5 | 64 | 16 | ~2.55 |
| MF_RGB + MaskedAng | RGB (only middle) | ✓ | ✓ | 7 | 128 | 17 | ~3.1 |
| MF_GRAY + MaskedAng-2 | Gray | ✓ | ✓ | 9 | 128 | 17 | ~3.1 |

As in Figure 3.11, inputs consisting of pure multiple images firstly feed into the model, and the model is then trained again with the inclusion of information about movement direction. Only the angle information is implemented in the inputs since the magnitude information does not have sufficient effect on the single-frame classification. The stride between consecutive frames is chosen as 1 because they are not very similar to each other. Moreover, gray-scale frames are generally preferred to avoid the negative impacts of excessive depth of the input that comes with the increase

in the number of frame and motion information used. However still, the model hyperparameters are modified as the number of channels and input complexity increase, allowing the model to better represent this variation. Furthermore, to avoid going up the number of model parameters, model complexity, and computational cost too much, just the number of filters in the first convolutional layer before the dense blocks and the growth rate are altered. The model variables and model inputs are detailed in Table 3.2.

The number of dense layers in dense blocks in all models are the same as [*6,16,32,16*]. Also, when masking the angles with the hand region to eliminate background noise, two different methods described in Section 3.2.2 are benefited for separate models. As a result, these models identify the frame in the center of the input by using either merely nearby frames or neighboring frames with optic information.

### 3.3.2. The Joint CNN+LSTM

The LSTM-based model, which performs better on time-series data, is recommended because of CNNs' poor capacity to capture temporal relationships. This model is a joint architecture of CNN with a high ability to process spatial information and LSTM with strong temporal link solving capability. Here again, DenseNet architecture is used as the CNN structure, and it feeds the LSTM by extracting features on those inputs. However, CNN does not run offline as a pre-trained model in this architecture, it is fully trained with LSTM as an end-to-end model. If it is employed as only an offline model, there would be no improvement in the feature extraction function as it always represents the same images with the same feature vectors. It only helps LSTM resolve the connection between these ordered vectors by simply representing images with smaller dimensional vectors. In other words, the capability of LSTM is also limited in this case because the better the feature extraction operation performs, the much better it is expressed in temporal correlation. Likewise, the LSTM architecture has no influence on the value of the incoming feature vectors, the output of the feature extraction function, and a full spatio-temporal context is not acquired as a result in this way. For these reasons, it is proposed to train the end-to-end CNN+LSTM structure together. Thus, LSTM may put pressure on the CNN variables due to temporal links among features, causing it to yield better and more

distinguishable feature vectors. The spatio-temporal relation between sequences can be worked out more effectively owing to this approach. Finally, LSTM-based models classify the last frame in the time series by analyzing the previous ones, not the middle frame as in the multi-frame CNN.

In the feature extraction stage, four blocks are utilized as in other CNN models, but the classification layer is discarded, so the last layer becomes the global average pooling. Unlike proposed CNN-based models, dense blocks have 6,12,24, and 12 layers, respectively, with a growth rate of 12. Because LSTM is included in the model, the depth, and the number of parameters of the CNN are reduced so as to keep the model complexity at the optimum level. Similar to CNN model inputs, LSTM-based models are trained separately with two different inputs, either original images or frames with optic angles. When the model is trained with complex input containing motion information, the number of layers in the first block is increased from 6 to 12 in order to make feature extraction more powerful. In this method, there is grayscale conversion as a minor pre-processing to avoid making the input volume too large, like multi-frame classification. Again, only the masked angle output is stacked just after the grayscale frame to inform the next movement of it. In all LSTM-based models, the time window is set to 5 and each input at *t* is represented by a 350-dimensional vector after the feature extraction step. In Figure 3.12, this hybrid architecture trained by two different inputs is schematized. The details of the LSTM-based models are expressed in Table 3.3.

Table 3.3: The variables of LSTM-based models.

| CNN+LSTM Models | Time Step | Motion (Angle) | Feature Extractor | | LSTM Units | Dense Nodes | Parameter (Million) |
|---|---|---|---|---|---|---|---|
| | | | Dense Layers | Growth | | | |
| CNN+LSTM | 5 | x | [6,12,24,12] | 12 | 512 | 128 | ~2.71 |
| CNN+LSTM+Ang | 5 | ✓ | [12,12,24,12] | 12 | 512 | 128 | ~2.93 |
| CNN+Bi-LSTM+Ang | 5 | ✓ | [12,12,24,12] | 12 | (2x256)=512 | 128 | ~2.5 |

This proposed LSTM-based architecture is unidirectional so the information flow in LSTM units is only from the beginning to the end point of the time series. That is, there is only one LSTM layer which flows forward direction. However, there are also bi-directional LSTM (Bi-LSTM) as powerful tools that can model dependencies between sequential data in both directions. Contrary to traditional LSTM, the bi-directional model carries out sequential data in both directions and uses the knowledge

of both flows for output. It does this by reversing the flow of information by adding another LSTM layer after forward-directional layer.

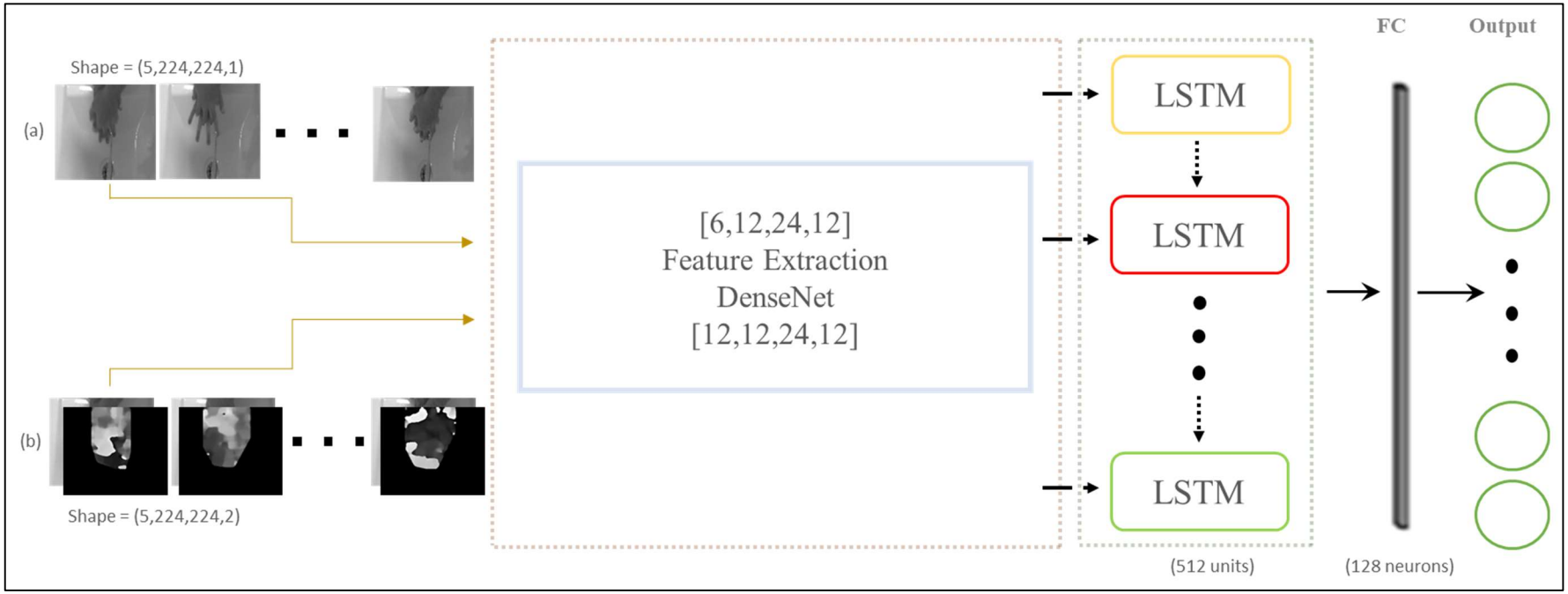


Figure 3.12: The architecture of CNN+LSTM with two different inputs. (a) Only original grayscale images. (b) Grayscale images with just next masked angle information. The architecture is trained as separate models with these inputs.

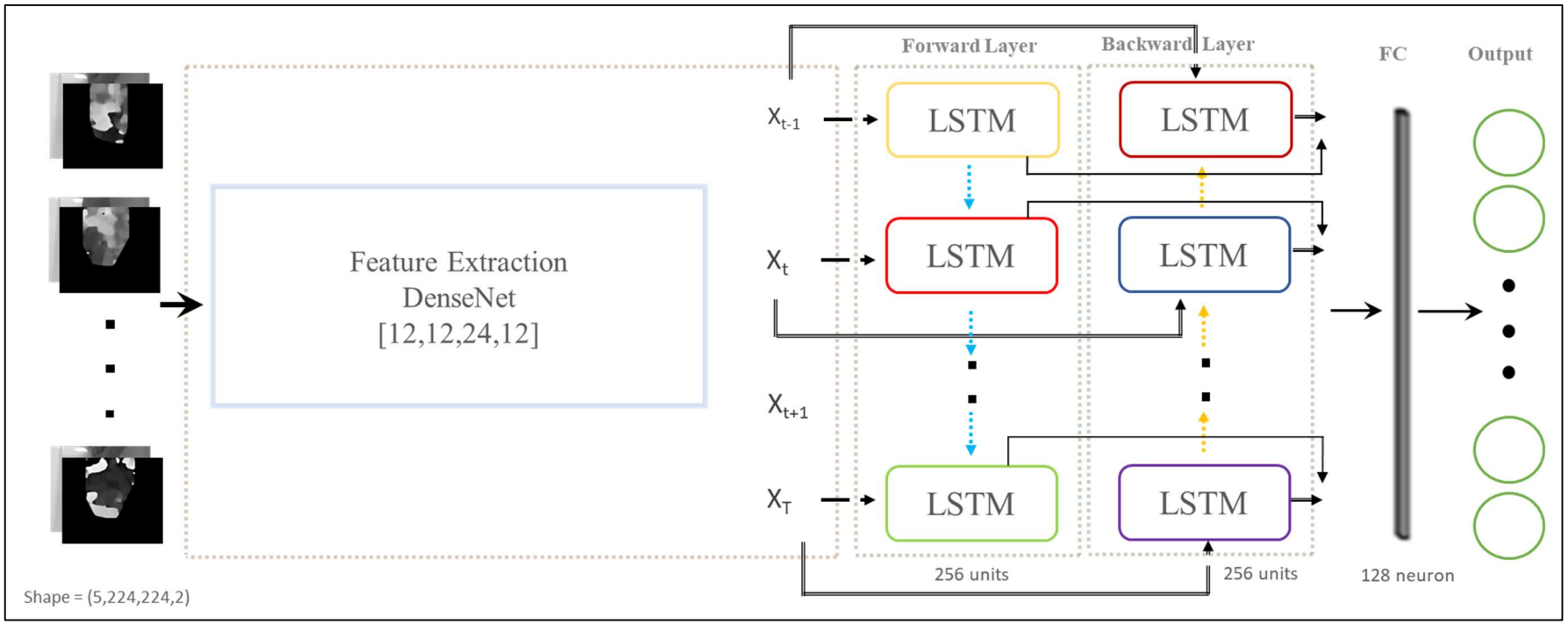


Figure 3.13: The architecture of CNN+Bi-LSTM model fed by inputs including next masked angle.

Therefore, it tries to establish a reverse temporal dependency with a backward flow from the last moment to the starting point in the time series. The variables of this Bi-LSTM model have been modified due to the addition of another LSTM layer flowing in the opposite direction. While there are 512 units in the unidirectional LSTM model, this unit is reduced to 256 for the Bi-LSTM model. While there are 512 units in the unidirectional LSTM model, the number of units is reduced to 256 for the Bi-LSTM model to prevent overfitting and increased model complexity. Thus, there are two LSTM layers with 256 units each processing information in two separate directions to learn the temporal dependence of the feature vectors. In both models, the outputs of the LSTM layers are linked to the fully connected layer having 128 neurons, and classification is performed with the softmax [41] function. Furthermore, this Bi-LSTM has the same DenseNet variables as the previous LSTM model as the feature extractor and is trained by only input involving next motion information. In Figure 3.13, the detailed structure of Bi-LSTM based model is demonstrated.

### 3.3.3. Vision Transformer

Transformer-based models have been implemented in the computer vision world after their conspicuous success in the NLP world. These architectures are called vision transformer in the image world. There is a trend toward vision transformer-based models from CNNs since it has state-of-art results for problems such as image classification and object detection. In the NLP, this idea works on words, so some preprocessing is required to adopt this architecture to the images. This preprocessing takes place as follows: an image is split into multiple patches of a certain size. In other words, if an image is thought of as a sentence, it is divided into patterns that compose it, just like the words that make up sentences. The main idea is that transformers do the task of CV by tracking the relationship between the patches that make up this image.

To begin with, single-frame classification is carried out by ViT, with original images and frames containing motion direction. In order to export the image to the transformer structure, the first step is that the *224x224* frame is segmented into *16x16* patches with a stride of 16. This splitting process is performed by a 2D convolutional layer with a dimension of 256. To sum up, a total of 196 vectors with a size of 256 are

obtained, each representing a separate *16x16* image area. Subsequently, position embeddings are inserted into each patch embedding so that it can preserve the location information and learn the structure of image from that. In addition to 196 patch embeddings, the extra class token is added to the beginning of the queue and a total of 197 vectors with 256-dimensional are ready to feed the transformer encoder. That is, these sequential vectors of patch embeddings form the inputs of the encoder. The basis of the encoder, which gives the same dimensional output vector as the input size by encoding incoming features, is the multi-head attention mechanism. This mechanism allows learning local and global dependencies in the frame. It tries to find similarities and relations between patches in the image with multiple attention matrixes. In addition to multi-head attention, normalization layer, multi-layer perceptron and residual connections are also included in the encoder. While the normalization layer help improve generalization capability, residual connections enable gradients to flow directly over the network without going through non-linear functions. The 0th element of the encoder feeds the linear layer, thereby classifying the handwashing gesture. These transformer encoders can work in parallel with more than one. As ViT model variables, 8 transformer encoders are used with 8 heads in single frame classification. Whereas the hidden size of the model is 256, the size of MLP is 512. This single frame classifying ViT model is individually trained on inputs with both the original image and optical information added, as in previous deep learning models. This model structure is designed to be step by step in Figure 3.14.

We take the ViT-based model further to better articulate the relationships on the image and their commitment. Thus, we propose a variant that is a combination of them to take advantage of the positive aspects of both CNN and transformer. Because CNN, which can be called a great feature extractor, creates small feature maps from the image and represents it with these maps. That is, the transformer is not run directly by patches from the image, but by feature maps generated by pre-trained CNN. In the previous model based only on the transformer, the one-layer convolution is actually used as a patch extractor ensuring just a low-level feature. However, this hybrid architecture exploits a pre-trained CNN that can extract high-level features. The shallower DenseNet architecture trained on this data from the previous CNN section is the backbone of this hybrid structure. The pre-trained DenseNet model is selected according to the model input.

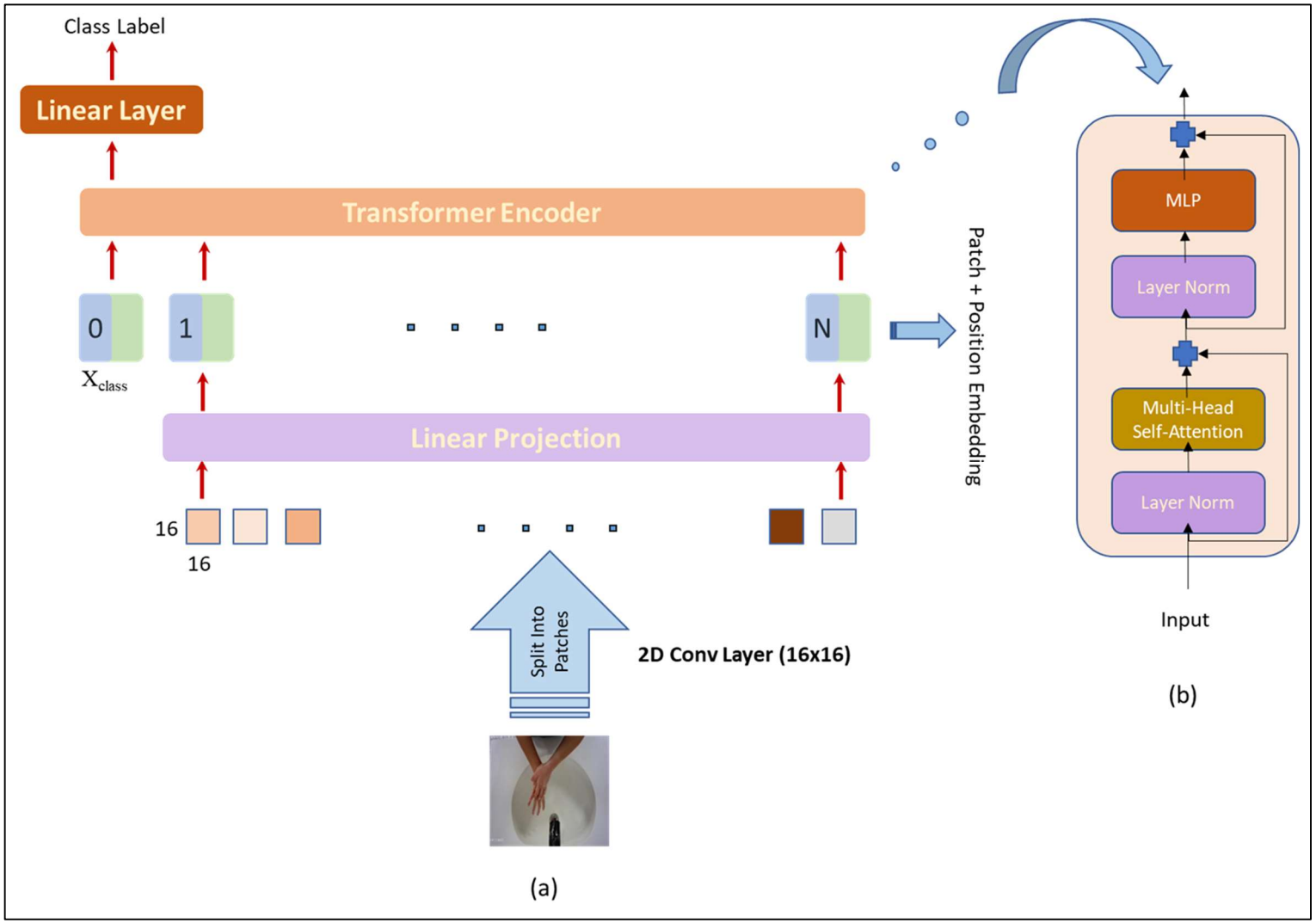


Figure 3.14: The architecture of vision transformer. (a) The general steps of ViT. (b) The structure of transformer encoder.

For instance, while the SF_RGB model is preferred as the backbone for classification only on the original images, the SF_RGB+MaskedAng-2 model is chosen for the input including the optical angle. This pre-trained backbone models are fine-tuned with transformers on data. The spatial size of the feature maps obtained after CNN, namely patches, is 1x1. Then, gesture recognition is performed by executing the same embedding operation and the encoding process operated by multi-head attention. The variables of all transformer-based models are specified in Table 3.4.

Table 3.4: The variables of ViT and hybrid models.

| ViT Models | Backbone Model | Layers | Hidden Size | MLP size | Heads | MLP Head | Parameter |
|---|---|---|---|---|---|---|---|
| ViT_RGB | x | 8 | 256 | 512 | 8 | 256 | ~4.5 |
| ViT_MaskedAng | x | 8 | 256 | 512 | 16 | 256 | ~4.7 |
| SF_CNN+ViT_RGB | SF_RGB | 4 | 256 | 512 | 8 | 256 | ~4.9 |
| SF_CNN+ViT_MaskedAng | SF_RGB+MaskedAng-2 | 4 | 256 | 512 | 8 | 128 | ~4.9 |
| MF_CNN+ViT | MF_GRAY-2 | 4 | 256 | 512 | 4 | 128 | ~4.8 |
| MF_CNN+ViT_MaskedAng | MF_GRAY + MaskedAng-2 | 2 | 256 | 512 | 4 | 128 | ~4.4 |

# 4. RESULTS

Many experiments are carried out in order to accurately predict handwashing movements by establishing different architectural-based models. These models are trained both with single frames removed from videos and with neighboring frames of those images. In addition, the direction of motion of the frames is added to the input as extra information, which can be useful in describing gesture. This motion information between adjacent frames is calculated by the optical flow approach and then masked by the hand region extracted by two distinct methods. The success of both separate model architectures and different input structures proposed for the handwashing problem is evaluated with lots of experiments. While all other models perform the classification based on the label of the middle frame of the input, LSTM-based models predict the class depending on the last image in the timeline. In all proposed models, Adam is employed as the optimization function and category cross-entropy is used to compute the model loss.

Initially, the results of the CNN-based model for single-frame classification are given in Table 4.1. This architecture, the baseline model, fed by only pure RGB images has an accuracy of 77%. Without any masking operations, using the angle information of all pixels before and after the image improves performance by around 3%. Moreover, taking the weighted average of the two prior and next motion angles of the target image increases the accuracy from 77% to 82%. It is an expected result that models fed by masking only the hand area on the angle channel gives the best performance. Because the noise of the background has been tried to be purified thanks to masking methos. However, when using the second strategy, this process has an obvious advantage over the first one because it allows for superior hand selection. Therefore, there is about 8% increase in accuracy over the baseline model (77% to 85%). Another observation is that using these motion directions with grayscale images is not as successful as SF_RGB+MaskedAng-2 since this model lacks the RGB color space's additional information. Finally, a remarkable result is that using the norm of motion alongside the angle does not improve the model performance compared with

the SF_RGB+MaskedAng-2 model, containing only the masked angle. Thus, the direction of motion is more significant than the magnitude of displacement to recognize handwashing. As a result, the model of SF_RGB+MaskedAng-2 has the best result in single frame classification with 85% accuracy.

In Table 4.2, the results of CNN experiments operating multi-frame classification are indicated. In these models, only masked angle is utilized as motion information due to its success in single-frame classification. When the models that work on the original images are examined first, it is seen that the RGB format does not make much difference on the gray scale. In other words, these two models produce remarkably comparable outcomes when employing the same amount of images but distinct color spaces. As the number of frames rises in input, the enhancement of the model on the problem is clearly seen with the 88% accuracy of the MF_GRAY-2, which is an 11% increase compared with baseline. Similarly, as with single-frame categorization, trials using the second masking strategy yield better performance. Furthermore, the angle information enables performance advancement in all these models and contributes to the 91% success of the MF_GRAY+MaskedAng-4.

Table 4.3 contains the results of LSTM-based architectures capable of extracting the temporal expression. In these models, only the masked angle values of the next action, which is obtained by 2nd strategy, are made use of besides the original image as it has a considerable effect on the problem when looking at the consequences of CNN. The CNN+LSTM model, which does not involve motion information, has close accuracy of 91% compared to the best CNN-based model. Due to its ability to capture the temporal connection, it can work with high accuracy without the need for any additional information. With the inclusion of extra directional information, unidirectional LSTM beats all CNN structures with the major development from 77% to around 94%. In addition, the Bi-LSTM model, which runs bi-directionally as opposed to uni- LSTM, has slightly poor consequences than uni-LSTM. Adding another counter-flowing LSTM layer increases the model complexity. Thus, the model becomes more open to overfitting from its flexible structure and loses its generalization power, which is the main goal. Because there is already a trend going forward, and the reverse is the opposite trend of the forward, so the model tends to overfit. As a result, the best performance belongs to the CNN+LSTM+Ang model.

Table 4.1: The experimental results of single-frame CNN-based model.

| | Models | Motion Info | | Mask Way | Data | Loss | Accuracy (%) | F1 (%) | Precision (%) | Recall (%) |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Angle | Norm | | | | | | | |
| Single Frame DenseNet | SF_RGB | x | x | x | Train | 0.05 | 98.17 | 98.16 | 98.26 | 98.07 |
| | | | | | Test | 0.88 | 77.05 | 77.11 | 79.56 | 74.85 |
| | SF_RGB+Ang | ✓ | x | x | Train | 0.04 | 98.63 | 98.63 | 98.72 | 98.55 |
| | | | | | Test | 1.16 | 80.5 | 80.5 | 81.22 | 79.92 |
| | SF_RGB+ScaledAng | ✓ | x | x | Train | 0.04 | 98.74 | 98.73 | 98.80 | 98.67 |
| | | | | | Test | 1.04 | 82.77 | 82.85 | 83.41 | 82.32 |
| | SF_RGB+MaskedAng-1 | ✓ | x | 1st | Train | 0.03 | 98.70 | 98.71 | 98.78 | 98.64 |
| | | | | | Test | 0.93 | 83.53 | 83.58 | 84.22 | 82.98 |
| | SF_GRAY+MaskedAng | ✓ | x | 2nd | Train | 0.03 | 98.80 | 98.81 | 98.86 | 98.76 |
| | | | | | Test | 1.06 | 81.50 | 81.64 | 82.35 | 80.97 |
| | SF_RGB+MaskedAng-2 | ✓ | x | 2nd | Train | 0.02 | 99.40 | 99.40 | 99.43 | 99.37 |
| | | | | | Test | 0.94 | **85.36** | 85.45 | 85.90 | 85.02 |
| | SF_RGB+MaskedAng&Mag | ✓ | ✓ | 2nd | Train | 0.02 | 99.32 | 99.31 | 99.33 | 99.28 |
| | | | | | Test | 1.07 | 84.09 | 84.15 | 84.55 | 83.77 |

Table 4.2: The experimental results of multi-frame CNN-based models.

| | Models | Frame Number | Motion (Angle) | Mask Way | Data | Loss | Accuracy (%) | F1 (%) | Precision (%) | Recall (%) |
|---|---|---|---|---|---|---|---|---|---|---|
| Multi Frame DenseNet | MF_GRAY-1 | 3 | x | x | Train | 0.03 | 98.80 | 98.80 | 98.88 | 98.73 |
| | | | | | Test | 0.80 | 85.80 | 85.99 | 86.50 | 85.50 |
| | MF_RGB | 3 | x | x | Train | 0.04 | 98.51 | 98.51 | 98.58 | 98.44 |
| | | | | | Test | 0.76 | 85.95 | 86.08 | 86.55 | 86.64 |
| | MF_GRAY-2 | 5 | x | x | Train | 0.04 | 98.69 | 98.70 | 98.76 | 98.64 |
| | | | | | Test | 0.60 | 88.47 | 88.62 | 89.05 | 88.23 |
| | MF_GRAY+MaskedAng | 3 | ✓ | 1st | Train | 0.03 | 98.91 | 98.92 | 98.99 | 98.87 |
| | | | | | Test | 0.90 | 85.08 | 85.20 | 85.56 | 84.86 |
| | MF_RGB+MaskedAng | 3 | ✓ | 1st | Train | 0.02 | 99.16 | 99.17 | 99.20 | 99.13 |
| | | | | | Test | 0.87 | 85.59 | 85.78 | 86.25 | 85.35 |
| | MF_GRAY+MaskedAng-2 | 5 | ✓ | 1st | Train | 0.02 | 99.20 | 99.21 | 99.26 | 99.15 |
| | | | | | Test | 0.65 | 88.03 | 88.08 | 88.39 | 87.79 |
| | MF_GRAY+MaskedAng-3 | 3 | ✓ | 2nd | Train | 0.03 | 98.76 | 98.77 | 98.82 | 98.71 |
| | | | | | Test | 0.70 | 86.98 | 87.16 | 87.62 | 86.71 |
| | MF_RGB+MaskedAng-2 | 3 | ✓ | 2nd | Train | 0.03 | 98.95 | 98.95 | 99.01 | 98.90 |
| | | | | | Test | 0.70 | 87.39 | 87.47 | 87.85 | 87.11 |
| | MF_GRAY+MaskedAng-4 | 5 | ✓ | 2nd | Train | 0.02 | 99.07 | 99.07 | 99.12 | 99.03 |
| | | | | | Test | 0.48 | **91.25** | 91.38 | 91.64 | 91.14 |

Table 4.3: The experimental results of LSTM-based models

| | Models | Motion (Angle) | Mask Way | Data | Loss | Accuracy (%) | F1 (%) | Precision (%) | Recall (%) |
|---|---|---|---|---|---|---|---|---|---|
| CNN + LSTM | CNN+LSTM | x | x | Train | 0.02 | 99.40 | 99.39 | 99.45 | 99.33 |
| | | | | Test | 0.37 | 91.02 | 91.19 | 91.61 | 90.79 |
| | CNN+LSTM+Ang | ✓ | 2nd | Train | 0.01 | 99.67 | 99.68 | 99.71 | 99.66 |
| | | | | Test | 0.39 | **93.85** | 93.88 | 94.16 | 93.65 |
| | CNN+Bi-LSTM+Ang | ✓ | 2nd | Train | 0.02 | 99.26 | 99.25 | 99.32 | 99.18 |
| | | | | Test | 0.36 | 91.89 | 92.01 | 92.53 | 91.59 |

Table 4.4: The experimental results of ViT-based models.

| | Models | Backbone Model | Data | Loss | Accuracy (%) | F1 (%) | Precision (%) | Recall (%) |
|---|---|---|---|---|---|---|---|---|
| Vision Transformer | ViT_RGB | x | Train | 0.03 | 98.87 | 98.88 | 98.95 | 98.81 |
| | | | Test | 0.84 | 78.61 | 78.62 | 78.71 | 78.54 |
| | ViT_MaskedAng | x | Train | 0.03 | 98.83 | 98.84 | 98.93 | 98.77 |
| | | | Test | 1.19 | 74.67 | 74.68 | 74.84 | 74.53 |
| | SF_CNN+ViT_RGB | SF_RGB | Train | 0.01 | 99.51 | 99.52 | 99.63 | 99.42 |
| | | | Test | 0.93 | 85.27 | 85.27 | 85.30 | 85.26 |
| | SF_CNN+ViT_MaskedAng | SF_RGB+MaskedAng-2 | Train | 0.01 | 99.85 | 99.85 | 99.91 | 99.81 |
| | | | Test | 0.87 | 85.93 | 85.94 | 86.11 | 85.78 |
| | MF_CNN+ViT | MF_GRAY-2 | Train | 0.01 | 99.83 | 99.83 | 99.88 | 99.80 |
| | | | Test | 0.60 | 90 | 90.14 | 90.15 | 89.87 |
| | MF_CNN+ViT_MaskedAng | MF_GRAY + MaskedAng-2 | Train | 0.01 | 99.90 | 99.90 | 99.93 | 99.88 |
| | | | Test | 0.52 | **92.10** | 92.11 | 92.15 | 92.08 |

The outcome of the transformer-based models is shown in Table 3.4. The ViT_RGB model has a competitive achievement with 78%, slightly surpassing the baseline SF_RGB model. In this challenge, it is evident that in the domain of images, transformers compete with CNN structures. However, the angle feature leads the opposite effect on model performance, resulting in poor results. On the other hand, although the SF_RGB+MaskedAng-2 model, which is the best single-frame outcome, has extra information, the unity of CNN and ViT architecture captures that model with 85% success, without needing any motion information on just RGB images. In addition, the hybrid of MF_CNN+ViT_MaskedAng advances the performance to 92%, outpacing all multi-frame CNN models and the CNN+LSTM model run by only original images.

Overall, the inclusion of motion data expressly improves the models' predictive power with up to 8% enhancement. LSTM gives the best outcome on this sequential problem because of its ability to resolve the temporal trend. In addition, the hybrid system of CNN and ViT can be preferred over LSTM models due to less computational cost and time requirement, as well as competitive achievements. When the confusion matrices of the best model and the baseline model are examined in Figure 4.1, it is seen that both models have problems in distinguishing left and right hand on the motion. For example, the recognition success is lower in rubbing the left fingertip with rubbing the right fingertip or rubbing the left thumb with rubbing the right thumb. Sure, the LSTM model has minimized this discrimination error when compared to other models, but there is still a right-left differentiation confusion.

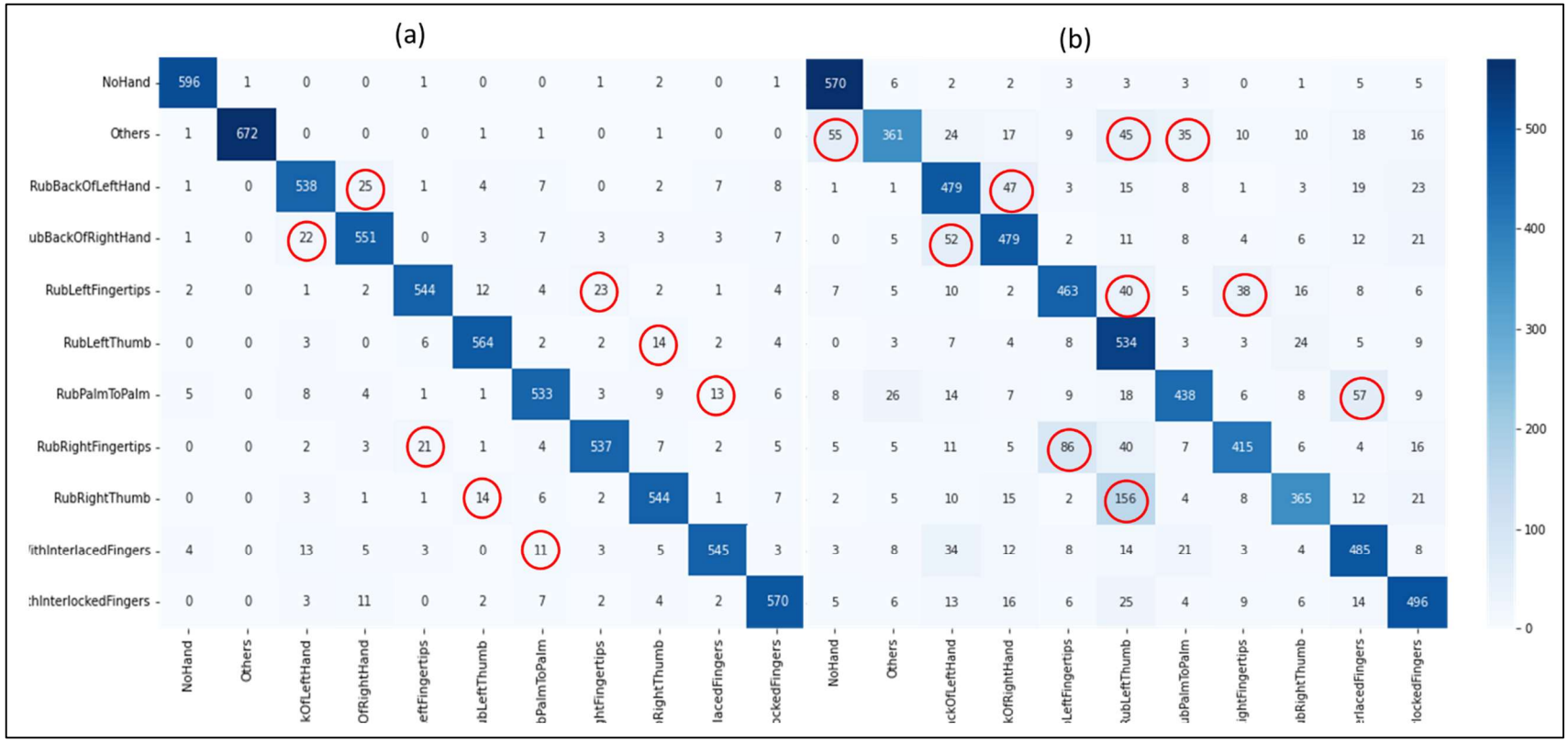


Figure 4.1: (a) The confusion matrix of CNN+LSTM+Ang model. (b) The confusion matrix of SF_RGB model. (The red circles indicate high numbers of incorrect predictions compared to others.)

# 5. CONCLUSION AND FUTURE WORK

The best precaution to take to prevent infection is to sustain hygiene, which is even more critical with the effects of the pandemic periods. The hands, where the simple way of getting the disease, should be disinfected during the day with specific movements to purify the dirtiness. In this thesis, the development of a framework that can correctly track this hand cleaning procedure is addressed. Since no comprehensive dataset exists for this topic, videos prepared by recording the hand washing processes of individuals are studied to recognize hand gestures. In the proposed system for the problem, a variety of end-to-end deep learning methods are employed, and some of these can draw both spatial and temporal inferences as well as utilize hand direction information during the washing process.

In fact, there is hardly any study that covers this handwashing monitoring in such depth. Previous studies have only made a person-based basis on whether the disinfection process is carried out. However, this study takes this process broadly and monitors handwashing actions to check whether the cleaning process is successful. The biggest difficulty in this problem is to follow the movements with high accuracy, since the hands are in close interaction with each other, as well as in touch with soap and water over washing. The proposed approach, on the other hand, eliminates the challenge of distinguishing movements that arise from close contact.

Initially, the videos that contain in random order all the movements specified by the WHO are labeled frame by frame. The purpose is to correctly predict the class of each frame to be able to monitor washing operation. Then, it is recommended to use a second additional information that can help in recognizing the handwashing gesture correctly as there are very similar actions. This extra information is the displacement vectors of hands, which varies throughout the act. As the washing movements constantly repeat a particular trend in different planes, the movement information of the hand advances the discrimination ability of the system. This motion information is gotten by optical flow approach by analyzing neighboring frames. The angle and distance information of all pixels on the image plane for the transition to next move is discovered using this method. When this displacement information between consecutive frames is analyzed, it is noticed that the angle information had a more generalizable structure on movements, unlike the norm of vector. Because every

movement actually has a set of flow patterns, and the hand can have diverse magnitudes while making the same angles. On the other hand, it is discovered that the noise in the angle knowledge is quite high in the area outside the hand region as opposed to the magnitude matrix due to the light variability, motion blur, shadows, and reflections in videos. Optical flow is also greatly affected by these reasons, since displacement is calculated according to the brightness pattern.

Secondly, two different strategies are suggested to isolate the background noise in this angle information. The first is color-based hand segmentation starting with the format conversion from RGB to YCbCr, which can better filter the skin pixels. The segmentation of hand pixels is performed on the frame by determining special threshold values on Cb and Cr channels. However, this method is quite a lot of non-hand pixel selection in some frames, since it is a color-based method. Although this trouble is tried to be solved by morphological image techniques, it is not sufficient to minimize noise. Therefore, the second strategy, which is the choosing of pixels having a value above a certain magnitude figure, is proposed. Large distance changes are intense in that location as movement occurs only in the hand region. If, after filtering according to the norm value, there is still a selection of pixels in the form of clusters in unrelated areas, this problem is eliminated by choosing the contour with the largest field. Finally, it is converted into a convex structure to make the contour softer after morphological operations. Two separate masking methods are tested for hand area detection on the same images, and it is concluded that the second approach eliminates noise more.

Deep learning models with three different fundamental architectures are recommended to categorize handwashing gestures. All models are first fed with only the original images and then run with complex input including this motion information to test the success of that idea on the problem. The first deep learning architectures are CNN-based models that perform single-frame and multi-frame classification. In all these CNN models, the DenseNet structure, which can transfer the features from the previous layers directly to the next layers, is preferred. The model against which we benchmark the success of our proposals is SF_RGB fed by the original single RGB frame, which has an accuracy of 77%. This angle or magnitude information is normalized, later they are directly added just before and after the relevant frame as an extra channel. The model in which the motion angle of all pixels is stacked without masking increases the performance from 77% to 80%. Looking at the masking ways,

noise reduction with the second strategy gives better results, achieving 85% accuracy with a significant 8% increase when it is applied on only angle knowledge. Additionally, grayscale model inputs in single-frame classification give worse results because they don't take advantage of RGB's additional information. On the other hand, stacking the magnitude channel with the masked angle produces more poor results than the model using only the extra angle information. This obviously indicates that the angle is more distinctive than the vector norm in recognizing motion. It is a multi-frame classification as a secondary CNN suggestion using neighbor frames with the center frame together. In contrast to single-frame classification, grayscale and RGB formats have similar outputs in multi-frame classification because the number of frames, that is the amount of spatial information, increases. Since the second masking solution is more successful in this multi-frame CNN too, only the angle information masked by that strategy is exploited as supplemental knowledge in all other architectures. It is expressly discovered that increasing the number of frames in the input and additionally using the masked angle information without being over-volume input produces tremendous outcome on that problem, accomplishing 91% accuracy.

Unlike CNN, LSTM-based models, which have high temporal inference ability instead of spatial, are suggested in an end-to-end model combined with CNN. Since there is a strong temporal dependency between frames, LSTM aims to capture this dependency that CNN cannot perform. The DenseNet model gives the feature maps that it extracts from the temporal data to the LSTM network, and the LSTM decides the class of the last moment in time by looking at the previous actions. Fed by only pure images, the CNN+LSTM architecture provides competitive results with the best CNN model (91%, including angle), while the inclusion of angle information in this joint structure jumps the accuracy to around 94% and beats all CNN-based models. In contrast to the traditional LSTM, although the Bi-LSTM model processes the timeline forward and backward, it generates worse results than uni-LSTM. Because of the Bi-LSTM architecture, an added LSTM layer increases the model complexity, which can lead to a decrease in the generalization power of the model and to be the tendency of overfitting. Also, the forward flow already has a trend, and the reverse flow is also the opposite of this trend. In other words, expressing the same trend in opposite directions can cause overfitting.

Due to the spectacular success of vision transformers in recent studies on other topics, we experiment with ViT-based instead of CNNs. This idea explores the

relations between the patches obtained from an image thanks to the self-attention mechanism and decides according to this context. The ViT model, run on the original single frame, very slightly surpasses the baseline CNN model, producing similar performance with 78%. It is a remarkable result that using the CNNs combined with the ViT model by feeding this hybrid structure only with the original single frame, provides an advancement of 8% over the baseline model (%85). This result catches the best single-frame CNN model utilizing optical flow. Finally, feeding multiple frames containing the angle information in this hybrid architecture outperforms CNN-based at the accuracy of 92%, competing with LSTM-based models.

To sum up, using motion information acquired by the optical flow with pure images has been proven to thrive the predictive and discrimination power of models on the problem. In addition, there is no doubt that the usage of hybrid or joint architecture enables crucial enhancements to this problem. Therefore, we can easily say that unity is strength. As a future study, an algorithm may be developed, which makes predictions by looking at the coordinates of certain skeleton points selected on a hand, and thereby temporal dependence can be better extracted with these coordinates. Furthermore, this proposed CNN+Vit-based model can be modified to be able to handle temporal progression besides spatial knowledge.

# REFERENCES


[1] WHO, (2009), “WHO Guidelines on Hand Hygiene in Health Care: First Global Patient Safety Challenge Clean Care Is Safer Care, Transmission of pathogens by hands”, Geneva.

[2] Lian K. Y., Chiu C. C., Hong Y. J., Sung W. T., (2017), "Wearable armband for real time hand gesture recognition," 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2992-2995.

[3] Savur C., Sahin F., (2015) "Real-Time American Sign Language Recognition System Using Surface EMG Signal," 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), 497-502.

[4] Abreu J. G., Teixeira J. M., Figueiredo L.S., Teichrieb V., (2016) "Evaluating Sign Language Recognition Using the Myo Armband," 2016 XVIII Symposium on Virtual and Augmented Reality (SVR), 64-70,

[5] Benalcázar M. E., Anchundia C. E., Zea J. A., Zambrano P., Jaramillo A. G., Segura M., (2018), "Real-Time Hand Gesture Recognition Based on Artificial Feed-Forward Neural Networks and EMG," 2018 26th European Signal Processing Conference (EUSIPCO), 1492-1496.

[6] Simao M., Neto P., Gibaru O., (2019), “EMG-based online classification of gestures with recurrent neural networks”, Pattern Recognition Letter, Elsevier, 128, 45-51.

[7] Côté-Allard U., Nougarou F., Fall C. L., Giguère P., Gosselin C., Laviolette F., Gosselin B., (2016), "A convolutional neural network for robotic arm guidance using sEMG based frequency-features," 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2464-2470.

[8] Tam S., Boukadoum M., Campeau-Lecours A., Gosselin B., (2020). “A Fully Embedded Adaptive Real-Time Hand Gesture Classifier Leveraging HD-sEMG & Deep Learning”, IEEE Transactions on Biomedical Circuits and Systems, 14(2), 232-243.

[9] Côté-Allard U., Fall C.L., Drouin A., Campeau-Lecours A., Gosselin C., Glette K., Laviolette F., Gosselin B., (2019), "Deep Learning for Electromyographic Hand Gesture Signal Classification Using Transfer Learning,", IEEE Transactions on Neural Systems and Rehabilitation Engineering, 27(4), 760-771.

[10] Naguri C. R., Bunescu R. C., (2017), “Recognition of Dynamic Hand Gestures from 3D Motion Data Using LSTM and CNN Architectures”, 16th IEEE International Conference on Machine Learning and Applications (ICMLA).

[11] Pisharady P. K., Saerbeck M., (2015), “Recent methods and databases in vision based hand gesture recognition: A review”, Computer Vision and Image Understanding, 141, 152–165.

[12] Alnaim N., Abbod M., Albar A., (2019), “Hand Gesture Recognition Using Convolutional Neural Network for People Who Have Experienced A Stroke”, 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT).

[13] Chung H. Y., Chung Y. L., Tsai, W. F., (2019), “An Efficient Hand Gesture Recognition System Based on Deep CNN”, 2019 IEEE International Conference on Industrial Technology (ICIT).

[14] Krizhevsky A., Sutskever I., Hinton G. E., (2017), “ImageNet Classification with Deep Convolutional Neural Networks”, Association for Computing Machinery, 60(6), 84-90.

[15] Simonyan K., Zisserman A., (2015), “Very Deep Convolutional Networks for Large-Scale Image Recognition”, 3rd International Conference on Learning Representations, San Diego, USA.

[16] Bao P., Maqueda A. I., del-Blanco C. R., García N., (2017), “Tiny Hand Gesture Recognition without Localization via a Deep Convolutional Network”, IEEE Transactions on Consumer Electronics, 63(3), 251–257.

[17] Li G., Tang H., Sun Y., Kong J., Jiang G., Jiang D., Tao B., Shuang X., Liu H., (2017), “Hand Gesture Recognition Based on Convolution Neural Network”, Cluster Computing.

[18] Lin H. I., Hsu M. H., Chen W. K., (2014), “Human Hand Gesture Recognition Using a Convolution Neural Network”, 2014 IEEE International Conference on Automation Science and Engineering (CASE).

[19] John V., Boyali A., Mita S., Imanishi M., Sanma N., (2016), “Deep Learning-Based Fast Hand Gesture Recognition Using Representative Frames”, 2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA).

[20] Wu X. Y., (2019), “A hand gesture recognition algorithm based on DC-CNN”, Multimedia Tools and Applications, 79, 9193-9205.

[21] Howard A. G., Zhu M., Chen B., Kalenichenko D., Wang W., Weyand T., Andreetto M., Adam H., (2017), “Mobilenets: Efficient convolutional neural networks for mobile vision applications”, arXiv.

[22] Park H., Lee J. S., Ko J., (2020), “Achieving Real-Time Sign Language Translation Using a Smartphone’s True Depth Images”, 2020 International Conference on COMmunication Systems & NETworkS (COMSNETS).

[23] Ward M. A., Schweizer M. L., Polgreen P. M., Gupta K., Reisinger H. S., Perencevich E. N., (2014), “Automated and electronically assisted hand hygiene monitoring systems: a systematic review”, American Journal of Infection Control, 42(5), 472–478.

[24] Kulkarni S., (2019), “Hand Hygiene Monitoring System for Hospital”, 3rd National Conference on Emerging Trends in Computer Engineering and Technology, School of Computer Engineering and Technology.

[25] Kaiming H., Zhang X., Ren S., Sun J., (2016), "Deep Residual Learning for Image Recognition,", 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778.

[26] Haque A., Guo M., Alahi A., Yeung S., Luo Z., Rege A., Jopling J., Downing L., Beninati W., Singh A., Platchek T., Milstein A., Fei-Fei L., (2018), “Towards Vision-Based Smart Hospitals: A System for Tracking and Monitoring Hand Hygiene Compliance” ArXiv.

[27] Yeung S., Alahi A., Haque A., Peng B., Luo Z., Singh A., Platchek T., Milstein A., Fei-Fei L. (2016), “Vision-Based Hand Hygiene Monitoring in Hospitals” AMIA.

[28] Kim M., Choi J., Kim N. (2020), “Fully Automated Hand Hygiene Monitoring\\in Operating Room using 3D Convolutional Neural Network”, ArXiv

[29] Zhong C., Reibman A. R., Mina H. A., Deering A. J., (2021), “Designing a Computer-Vision Application: A Case Study for Hand-Hygiene Assessment in an Open-Room Environment”, Journal of Imaging, 7(9):170.

[30] Zhou B., Andonian A., Oliva A., Torralba A., (2018), “Temporal Relational Reasoning in Videos”, ArXiv.

[31] Turkmen B., (2021), “Continuous Vs Fixed-Window Temporal Data Processing For Hand Movement Analysis”, Master’s Thesis, Gebze Technical University.

[32] Simonyan K., Zisserman A., (2014), “Two-Stream Convolutional Networks for Action Recognition in Videos”, Proceedings of the 27th International Conference on Neural Information Processing Systems, 1, 568-576, MIT Press, Cambridge, MA, USA.

[33] Cui R., Liu H., Zhang C., (2019), “A Deep Neural Framework for Continuous Sign Language Recognition by Iterative Training”, IEEE Transactions on Multimedia, 1–1.

[34] Dosovitskiy A., Beyer L., Kolesnikov A., Weissenborn D., Zhai X., Unterthiner T., Dehghani M., Minderer M., Heigold G., Gelly A., Uszkoreit J., Houlsby N.,

(2021), “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, 9th International Conference on Learning Representations, Virtual Event, Austria, 3-7 May.

[35] D’Eusanio A., Simoni A., Pini S., Borghi G., Vezzani R., Cucchiara R., (2020), “A Transformer-Based Network for Dynamic Hand Gesture Recognition”, 2020 International Conference on 3D Vision (3DV), 623-632.

[36] Zhang X., Yin L., (2021), “Multi-Modal Learning for AU Detection Based on Multi-Head Fused Transformers”, IEEE Press, 1-8.

[37] Montazerin M., Zabihi S., Rahimian, Mohammadi A., Naderkhani F., (2022), “ViT-HGR: Vision Transformer-based Hand Gesture Recognition from High Density Surface {EMG} Signals”, CoRR, abs/2201.10060.

[38] Farnebäck G., (2003), “Two-Frame Motion Estimation Based on Polynomial Expansion” Lecture Notes in Computer Science, 363–370.

[39] Lucas B. D., Kanade T., (1981), “An Iterative Image Registration Technique with an Application to Stereo Vision”, Morgan Kaufmann Publishers Inc., Proceedings of the 7th International Joint Conference on Artificial Intelligence-2, 81, 674–679, Vancouver, BC, Canada.

[40] Huang G., Liu Z., Maaten L. V. D., Weinberger K. Q., (2016), “Densely Connected Convolutional Networks”, ArXiv.

[41] Goodfellow I., Bengio Y., Courville A., (2016), "Deep Learning”, MIT Press.

# BIOGRAPHY

Furkan Kasım took his bachelor's degree in Industrial engineering from Yıldız Technical University in February 2020. Since the following semester, he has been pursuing a master's degree in Industrial Engineering at Gebze Technical University. In 2020, he earned a scholarship from the Ministry of National Education to be sent abroad for PhD studies. Computer vision and machine learning are among his research interests.

# APPENDICES

## Appendix A: The Publications about the Thesis.

Kasım F., Budak A., Genç Y., (2022), “Optical Flow-based Temporal Video Analysis for Hand Hygiene Monitoring”, The 30th IEEE Conference on Signal Processing and Communications Applications, Turkey, 15-18 May.