TOUCH GESTURES CLASSIFICATION BY DEEP LEARNING METHODS

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

Master Of Science

in Mechanical Engineering

by
Irmak EGE

July 2022
İZMİR

ACKNOWLEDGEMENTS

I would like to specially thank my supervisor, Assist. Prof. Kerem Altun, who made this work possible. His guidance and advice carried me through all the stages of writing my master's thesis.

Secondly, I would like to express my thanks to Mehmet Ogün Sarp for assistance and technical support in the design and production of sensor settings. His entertaining attitudes always keep me alive while facing hardship. Furthermore, I would like to thank Emin Özdemir for his friendship and mind-blowing discussions throughout the last semester of my master's degree.

I dedicate this thesis to my life partner, Ümmü Dilan Gökçe, who is the equilibrium point of all the randomness in my life. With her infectious enthusiasm, it was possible to complete this thesis. Your existence was what sustained me this far. I would like to give special thanks to her.

I would like to convey my gratitude to my family, who deserve better words to describe my gratefullness for their unwavering support. I would like to thank my mother, Şerife Ege, who nurtured me with love and support throughout my life. I would also like to express my gratitude to my father, Aydan Ege, who has always supported and encouraged me to succeed. Thank you to Burak Kaya Ege, my younger brother, for being a friend.

Finally, but certainly not least, a special thanks goes out to the many individuals who went unmentioned yet contributed in some manner to the success of this thesis.

ABSTRACT

TOUCH GESTURES CLASSIFICATION BY DEEP LEARNING METHODS

In this study, we carried out social touch gesture classification on two publicly available datasets, Corpus of Social Touch (CoST) and Human-Animal Affective Robot Touch (HAART), and our demo dataset. In order to classify touch gesture datasets, four different models are proposed: 3-dimensional convolutional neural network (3D-CNN), 3-dimensional convolutional-long term short term memory neural network (3D-CNN-LSTM), 3-dimensional convolutional-bidirectional long term short term memory neural network (3D-CNN-BiLSTM) + and 3-dimensional convolutional transformers network (3D-CNN-Transformer). The fundamental layer of the proposed deep neural network architectures is 3-dimensional convolution layer that enables to extract spatio-temporal features of touch gestures. In this regard, with the use of spatio-temporal features of touch gestures, generalization performance of proposed four models have been improved using data augmentation techniques by applying randomly shift and rotation, and ensemble learning. Additionally, We also found out that Stochastic Gradient Descent (SGD) optimization algorithm has better generalization performance than Adaptive Moment Estimation (ADAM), which is used more frequently in deep learning. The accuracy of classification results of three dataset is investigated in terms of proposed model. The results showed that the proposed methods, especially ensemble classifier and the ensemble classifier with data augmentation, are beneficial for obtaining more generalizable learning algorithms. The scripts of deep neural network architecture are available upon request.

Keywords: *Touch Gesture Classification, Touch Gesture Recognition, Deep Learning, 3-Dimensional Convolution, Long Term Short Term Memory, Transformers, Generalization, Data Augmentaion*

ÖZET

DOKUNMA HAREKETLERİNİN DERİN ÖĞRENME YÖNTEMLERİ İLE SINIFLANDIRILMASI

Bu çalışmada, açık erişime sahip iki veri seti, Corpus of Social Touch (CoST) ve Human-Animal Affective Robot Touch (HAART) ve oluşturduğumuz veri seti üzerinde sosyal dokunma hareketi sınıflandırması gerçekleştirdik. Dokunma hareketi veri setlerini sınıflandırmak için dört farklı model önerilmiştir: 3 boyutlu evrişimli sinir ağı (3D-CNN), 3 boyutlu evrişimli uzun süreli kısa süreli bellek sinir ağı (3D-CNN-LSTM), 3 boyutlu evrişimli çift yönlü uzun süreli kısa süreli bellek sinir ağı (3D-CNN-BiLSTM) ve 3 boyutlu evrişimli dönüştürücü ağı (3D-CNN-Transformer). Önerilen derin sinir ağı mimarilerinin temel katmanı, dokunma hareketlerinin uzamsal-zamansal özniteliklerini çıkarmayı sağlayan 3 boyutlu evrişim katmanıdır. Bu bağlamda, dokunma hareketlerinin uzamsal-zamansal özelliklerinin kullanılmasıyla, önerilen dört modelin genelleme performansı, rassal olarak uygulanan dönme ve öteleme gibi veri artırma teknikleri ve toplu öğrenme kullanılarak geliştirilmiştir. Ek olarak, Stokastik Gradyan İniş (SGD) optimizasyon algoritmasının, derin öğrenmede daha sık kullanılan Uyarlamalı Moment Tahmini (ADAM) algoritmasından daha iyi genelleme performansına sahip olduğu sonucuna ulaştık. Üç veri kümesinin sınıflandırma sonuçlarının doğruluğu önerilen modeller ışığında araştırılmıştır. Sonuçlar, önerilen metodların, özellikle toplu sınıflandırıcı ve veri büyütmeli topluluk sınıflandırıcı algoritmalarının, daha genelleştirilebilir öğrenme algoritmaları elde etmek için faydalı olduğunu göstermiştir. Derin öğrenme mimarilerinin kod betik dosyası istek üzerine temin edilebilir.

Anahtar Kelimeler: *Dokunma Hareketi Sınıflandırma, Dokunma Hareketi Tanıma, Derin Öğrenme, 3-Boyutlu Evrişim, Uzun Süreli Kısa Süreli Bellek, Dönüştürücü, Genelleştirme, Veri Büyütme*

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

CHAPTER 1

INTRODUCTION

Human behaviour can be defined as the set of actions of individuals to respond to internal and external stimuli from their environment. While these set of actions are complex, they play a substantial role in the construction of our experiences and understanding the world [49]. The complexity of human behaviour can be reduced into three interactive components: action, cognition and emotion of human. The action might be seen as a self-initiated, purposeful movement sequence. It is, of course, complemented by proper cognitive ability in a suitable state. Examples of such well equipped human actions might be body movements, hand gestures or touch gestures.

Touch is one of the actions that enables shaping the individual's physical environment and communication with the world around us. It is the only sense that allows physical contact with living and non-living beings. Through the instrumentality of multiple actions of touch, the individuals express their state of emotions to each other as well. Understanding such interactions between humans plays a significant role in the development of psychology.

In some sense, generally in the digital age, individuals often prefer to interact with machines as they do with other individuals [15], and understanding those interactions via computer analysis might be useful. The determination of human actions as behaviours by computer analysis leads to research areas such as human-computer interaction, affective computing [7], social signal processing [69]. These areas are commonly used in the development of social robots with the purpose of applications ranging from therapeutic robots for intervening with children and disabled individuals to improve the socialization of the elderly. Moreover, some of the current research is concentrated on implementing touch sense similar to humans on robotic grippers [37].

Social robots are able to sense the touch behaviour via tactile sensing modality [11] which provides unique communication to express an individual's emotions and intent through affective touch [27]. These robots also benefit from other modalities such as audio and facial expression. However, in the case of interaction with contact, perception of tactile

signals during social interaction is doable only with proper combination of robotic skin hardware with detection and recognition algorithms.

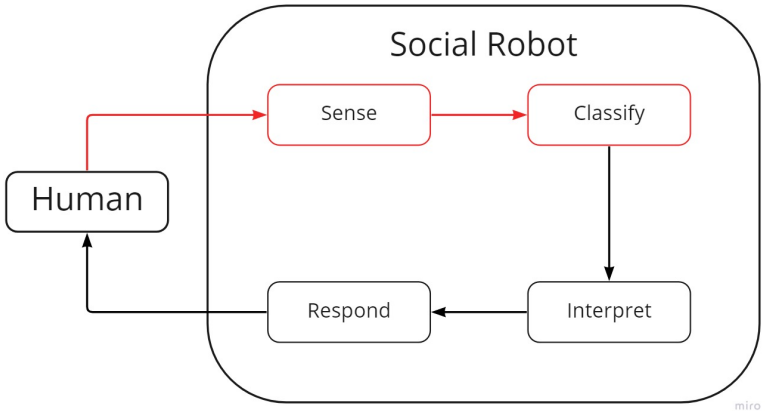


Figure 1.1: Interaction cycle for an robot with social intelligence to respond to human touch [31]

The interaction cycle shown in Figure 1.1 consists of two main components: the human and the robot. The human is the natural actor in the environment. The aim of the interaction cycle is to respond to the natural actor in the proper way. The social robot is equipped with tactile sensors and is responsible for sensing, classifying, interpreting and responding to the human. These four capabilities should be essential to understand the social meaning of the human touch.

Equipping a robot with touch sensors is the starting point of touch interaction. Once the sensor encodes a touch gesture, a specific algorithm must recognize what type of touch gesture is being performed. Recognition of touch can be examined in terms of the classification of touch gestures. A robust classifier must be designed in order to obtain accurate results. The other element is the interpretation of the social meaning of touch gestures. Other important capabilities of touch interaction include appropriate interpretation and thus responding to the human. Inferring the meaning from sensed gestures requires several sensor settings and cannot be easily done. The final step, responding, must be done in a way that makes sense to the human, not to confuse his/her mental state. These four elements of touch interaction are closely related to each other, and each of them is as important as the others.

In the following, we provide a summary of the existing literature on prominent studies in touch gesture classification with classical machine learning and deep learning methods, necessary details of touch gesture datasets used in this work, the motivation and problem definition of this work.

Chapter 2 is about methods that give a detailed information about deep learning architectures. This chapter focuses on the definition of the main building blocks of architectures and the mathematical operation inside these blocks. Additionally, this chapter answers the question of how the hyperparameter of proposed architectures is tuned and how is generalization ability of deep learning architectures are improved using several techniques: data augmentation, ensemble classifier and using Stochastic Gradient Descent algorithm instead of Adaptive Moment Estimation (ADAM).

Chapter 3 gives the results of deep learning methods introduced in Chapter 2 on three datasets. The obtained results are summarized, discussed and compared to the existing literature in Chapter 4. In Chapter 5, observations from classification results are discussed in detail and a research direction for further analysis of touch gestures is explained.

1.1 Literature Review

In earlier studies with touch gesture recognition, several social robots have been designed for communication with humans and interpret touch gestures. To the best of our knowledge, the very first social robot PARO [58] is able to recognize two touch gestures, stroke and hit. Haptic Creature [72] can sense various emotional states of humans through interaction with them. There are plenty of applications such as Huggable [61], Probo [56], AIBO [16], Miro [48], Lovot [73], etc. Comprehensive and detailed reviews can be found in [59], [57] and [20].

The Social Touch Gesture Challenge, which was organized as a satellite event in 2015 [29], has sparked further studies on the recognition of social touch gestures. The challenge has focused on recognition of touch gestures for two datasets: Human-Animal Affective Robot Touch (HAART) [14] and Corpus of Social Touch (CoST) [32]. The touch gestures in these datasets represent the problem of touch gesture recognition as an 8x8 grid

of pressure sensors to sense various social touch gestures. There were 4 papers accepted to the challenge as a benchmark results in social touch gesture classification.

In [4], researchers used statistical properties, image features and autoregressive model coefficients to extract features from touch gestures. In addition to feature extraction, several feature selection methods were performed in order to increase the accuracy of both datasets. The accuracy, which was obtained after applying those methods with random forest classifier, varied from 26% to 95% of CoST and from 60% to 70% of HAART. Another paper in the challenge, [17], inputted high level features of touch gestures, statistical distribution of pressure surface at each frame level, binary motion history to capture shape of gestures, statistical properties of each signal in an 8x8 grid, spatial multi-scale motion history histogram to capture touch dynamics to random forest classifier. They obtained the accuracy of 59% and 67% for CoST and HAART, respectively.

In [63], the authors implemented random forest classifier as well as with overall statistics of the gestures as global features, spatial relationship over temporal features as channel-based features and some signal processing method such as Fast Fourier Transform and Discrete Cosine Transform as sequence features. The overall accuracy they obtained was 61.34% for CoST and 70.91% for HAART as recorded a state of art results in touch gesture literature.

In [25], authors approached the problem by extracting features of touch gesture and calculating likelihood score of them, and finally using these scores as inputs to simple logistic regression. Three different methods were performed for extracting features to identify touch gestures: 1) Using a deep auto-encoder to obtain spatial features in low dimensions. 2) Determining geometric moment of each gesture frame. 3) Calculating summary statistic for both spatial and temporal features to obtain high level features of gestures. They got 56% accuracy on CoST, 71% accuracy on HAART.

As we have mentioned earlier, the social touch gesture challenge has accelerated researches on social touch gesture recognition. The detailed review of papers until the year 2017 can be accessed in [30]. Additionally, authors in [30] published their own results on the CoST dataset. They compared results of several classifiers such as Bayesian classifier, Decision tree, Support Vector Machine and Neural Network in four different mode; gentle, normal, rough, and combination of all. They did not obtain the state of the art result but discussed their results on the recognition of touch gestures in a detailed way.

A model that classifies touch gestures using spatio-temporal feature fusion was proposed in [42]. First, a set of coefficients of each sensor channel is obtained by Discrete Wavelet Transform (DWT). Then, these sets of wavelet coefficients were reduced to five statistical features for classification tasks such as norm, standard deviation, skewness and kurtosis which were inputed to four classifiers. The maximum obtained accuracy from classification was 64.17% on CoST which is the state of the art result for the CoST dataset.

The very first deep learning models to classify touch gestures were performed for both datasets in the paper of [26]. Three different models were considered: 2-dimensional convolutional neural network (2D-CNN) to capture spatial features, 2-dimensional convolution and recurrent neural network (RNN) to capture both spatial and temporal features, and autoencoder (AE) following with recurrent neural network to capture reduced noise spatial features and temporal features. These promising models in the deep learning field did not outperform the state of the art results but gave similar results compared to classical machine learning models.

Zhou and Du [74] analyzed different deep neural networks architectures including 2D-CNN, 3-dimensional convolutional and long term short term (LSTM) neural network, LSTM with extracted features by geometric moments, combination of RNN, LSTM, 2D-CNN and 3-dimensional convolutional neural network on HAART dataset. Their 3D-CNN architecture was achieved the state of art result 76.1% accuracy.

Bani and Chetouani [13] proposed another deep learning model named as Attentive Touch Model (ATM). The model consists of three building blocks; spatiotemporal encoding with convolutional block and positional encoding, intra-attention and inter-attention to optimize the learning process. The results are 60.9% and 67.8% for both dataset CoST and HAART, respectively.

General motivations behind touch gesture recognition literature focused on classification of all recorded gestures and their variants as accurately as possible and such classification must be independent from subjects. Previously reviewed papers have followed the touch gesture challenge protocol. However, Albawi et al. [2] have obtained 63.7% mean accuracy with leave-one-subject-out cross validation for all subjects on CoST using 2D-CNN.

There are various machine learning algorithms in the social touch gesture literature, especially concentrated on classification because of The Social Touch Gesture Challenge

in 2015 [29]. In the Challenge and the following years, researchers mainly focused on machine learning methods with manually extracted features. Thereafter, with the rise of the deep learning field, the attempts have been made to increase classification accuracy in the touch gesture literature.

1.2 Touch Gesture Datasets

Three datasets CoST, HAART, our demo dataset were classified in this thesis. The first dataset, CoST, was introduced in 2014 and there was no touch gesture dataset available for research and benchmarking purposes before it. The HAART dataset was also introduced in the following year. Both datasets are publicly available which contain labeled training and test set. After their introduction, they became very popular among gesture recognition, affective computing and human-robot interaction research. The summary of datasets' properties can be found in the Table 1.1.

Table 1.1: Dataset properties of CoST and HAART

Dataset Properties	**CoST Dataset**	**HAART Dataset**	**Our Demo Dataset**
Number of social touch gestures	14	7	6
Size of sensor grid	8x8	8x8	15 x15
Sensor sampling rate	135 Hz	54 Hz	10 Hz
Sensor values	0-1023	0-1023	0-1023
Duration of gestures	Variable	8s	20s
Touch surface	Mannequin arm	Various	Cotton Fabric Condition
Variants	Gentle, normal, rough	Substrates and covers	No Variants
Number of subjects	31	10	5
Training and test split	21 subjects/ 10 subjects	7 subjects/ 3 subjects	5 subjects/ 1 Subjects
Number of gesture	7,805	829	300

1.2.1 CoST: Corpus of Social Touch Dataset

CoST stands for Corpus of Social Touch that was introduced in order to transfer the tactile modality from interpersonal touch interaction to Human-Robot Interaction (HRI)[31].

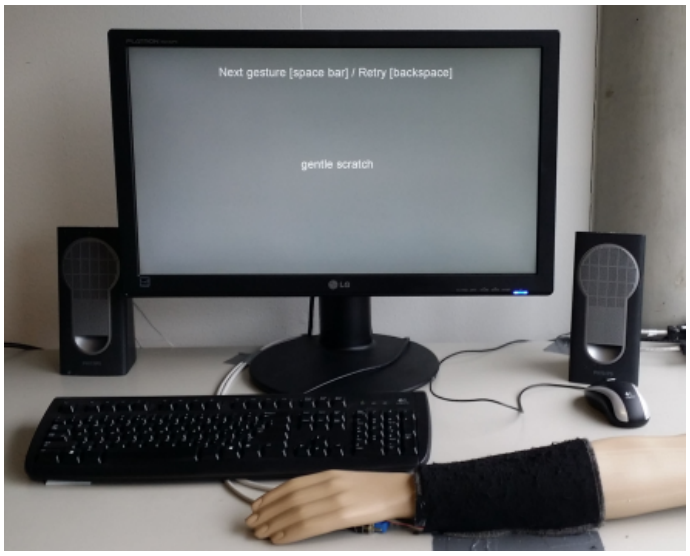


Figure 1.2: The experimental setting of CoST dataset reprinted with permission from [31]

Touch gesture data in CoST were collected from a sensor grid wrapped around a mannequin arm as in Figure 1.2. The arm was chosen as the touch surface intentionally because it was less invasive and neutral than other parts of the human body such as head, shoulders and legs [23]. The sensor grid contains 64 pressure sensors placed as 8x8 sensor array. There were 14 touch gestures recorded which were taken from the touch gesture dictionary of [72]. The definition of each touch gesture can be seen in Table 1.2. The 14 gestures were performed by 31 subjects in 3 variations: normal, gentle and rough.

Table 1.2: Recorded touch gestures in CoST dataset

Touch Gesture	**Gesture Definition**
Grab	Grasp or seize the arm suddenly and roughly.
Hit	Deliver a forcible blow to the arm with either a closed fist or the side or back of your hand.
Massage	Rub or knead the arm with your hands.
Pat	Gently and quickly touch the arm with the flat of your hand.
Pinch	Tightly and sharply grip the arm between your fingers and thumb.
Poke	Jab or prod the arm with your finger.
Press	Exert a steady force on the arm with your flattened fingers or hand.
Rub	Move your hand repeatedly back and forth on the arm with firm pressure.
Scratch	Rub the arm with your fingernails.
Slap	Quickly and sharply strike the arm with your open hand.
Squeeze	Firmly press the arm between your fingers or both hands.
Stroke	Move your hand with gentle pressure over arm, often repeatedly.
Tap	Strike the arm with a quick light blow or blows using one or more fingers.
Tickle	Touch the arm with light finger movements.

Other important properties of the dataset are that there are 7805 gesture sequences that belong to 21 training and 10 test subjects. The encoded sensor values range from 0 to 1023 which is sampled at 135 Hz.

1.2.2 HAART: Human-Animal Affective Robot Touch Dataset

The aim of collecting touch gestures in the HAART dataset is to mimic human-animal interaction. The dataset contains 7 social touch gestures: no touch, constant, pat, contact without movement, rub, scratch, stroke and tickle. The definition of gesture can be found in Table 1.3. The reason behind selecting these gestures is that they were found to be the most often used gestures in the touch gesture dictionary [72]. Ten subjects performed the touch actions on a 10x10 pressure sensor in Figure 1.3. Each instance is captured during 10 seconds and sampled at 54 Hz with different conditions: all permutations of 3 substrate conditions (firm and flat; foam and flat; foam and curve), and 4 fabric cover conditions (none; short minkee; long minkee; synthetic fur).

Figure 1.3: The experimental setting of HAART, reprinted with permission [14]

The HAART dataset was provided with an 8x8 array in each frame by trimming from a 10x10 array. The pressure values range from 0 to 1023. The resulting dataset includes 829 gestures constituted by 12 conditions, 7 gestures and 10 subjects. There were 11 faulty gestures in the dataset that were removed.

Table 1.3: Recorded touch gestures in HAART dataset

Touch Gesture	Gesture Definition
No touch	No contact on sensor.
Constant	Exert a steady force on the arm with your flattened fingers or hand.
Contact without movement	Any undefined form of contact.
Pat	Gently and quickly touch the arm with the flat of your hand.
Rub	Move your hand repeatedly back and forth on the arm with firm pressure.
Scratch	Rub the arm with your fingernails.
Stroke	Move your hand with gentle pressure over arm, often repeatedly.
Tickle	Touch the arm with light finger movements.

1.2.3 Our Demo Dataset

Our sensor setting is made from conductive silver strips, resistive fabric and paper of acetate. The conductive silver strips are sprinkled on the acetate paper at certain intervals. The resistive fabric is placed between the layer of conductive silver strips and its rotated version by $90°$. The sensor setting is settled on a pillow and covered with fabric cotton as shown in Figure 1.4. The piezoresistive sensors are preferred among other alternatives such as piezoelectric and capacitive sensors in our sensor setting since the major reason is that they are quite sensitive to force impacts [18]. The minor reasons are their low cost, flexibility and bending ratio under variety of conditions.

The sensor properties in Table 1.1 has 15x15 sensor grid. The pressure value of gesture range in our dataset is in the range of 0 to 1023 and the gestures are performed during 20 second at sampling rate of 10 Hz. The gesture sequences in our dataset belong to 6 classes: grab, tap, hit, pat, scratch, stroke. These gestures are performed by Turkish

subjects. Thus, Turkish translation of these gestures can be found in Table A.1 in Appendix A. There are no additional variants of gesture. They are performed 10 times in normal pressure by 5 subject. In total, our dataset contains 300 touch gestures.

(a) The sensor setting (b) The tactile sensor

Figure 1.4: Our sensor setting

1.3 Motivation

The main motivation of the thesis is how touch perception works in human-robot interaction, especially how the touch gestures can be classified. To understand the mechanism of touch gesture recognition as broadly as possible, a tactile sensor is developed and the recorded touch gestures with appropriate experimentation settings are classified. Additionally, the two publicly available touch gesture datasets are classified in order to increase the benchmark results.

The classification of touch gesture sequences in CoST, HAART, and our demo dataset is done using various deep learning methods by extracting spatio-temporal features of touch gestures. The current literature has focused on increasing the classification accuracy of test sets of dataset. This direction might be helpful in some sense but it is

surely not sufficient for the recognition task. During recognition of touch gestures, the motivation should take as much benefit as possible from training set. The aim should be obtain generalizability of given data at hand.

The classification begins with how the touch gesture sequence is represented. Formally, a touch gesture sequence $\mathbf{x}_i \in \mathbb{R}^{H \times W \times L}$ can be represented as:

$$\boldsymbol{x_i} = \begin{bmatrix} x_{j,(1,1)} & \dots & x_{j,(1,w)} \\ \vdots & \ddots & \vdots \\ x_{j,(h,1)} & \dots & x_{j,(h,w)} \end{bmatrix} \tag{1.1}$$

where $x_{j,(h,w)} \in \mathbb{R}^n$, for $h = 1, 2, \dots, H$, $w = 1, 2, \dots, W$ and L is the duration in terms of number of samples in touch gesture sequence $\mathbf{x}_i$. The approach to the problem of gesture recognition is as follows: The touch gestures data were collected from the $H \times W$ sensor channels at each time step j. With the entire collection of one touch gesture, the shape of data is $H \times W \times L$ which represents a $H \times W$ dimensional time series array where L is the total number of samples. An instance of touch gesture sequence has $H \times W$ dimensional spatial size and L dimensional temporal size. An example of gesture array that is mapped to range from 0 to 255 can be seen in Figure **??**. White-colored squares are represented by value of 0 which corresponds to lowest pressure in sensor array and black-colored squares are represented by value of 255 which corresponds to highest pressure in sensor array.

The second step is how the ground truth of the class label of the gesture sequence is established. Even if the culture of an individual has a significant effect on the representation of touch gestures, the optimal label can be defined by who makes the experimentation settings.

The third step is how the classification of touch recognition is approached. There are various alternatives to approaching the classification of gestures [3]. The approach to the problem is as follows: given a set of n training touch gesture samples $x \in \mathcal{X}$ and $y \in C$ is the class label of x from the set of classes $C = \{1, ..., c\}$, it is aimed to design a classifier based on deep neural network by extracting spatio-temporal features such that $f_\theta : \mathcal{X} \rightarrow C$ denotes a transformation with learned parameters θ from the input space $\mathcal{X}$ to predict the class of an unseen touch gestures.

1.4 Scope

The general idea behind the recognition of tactile signals is to classify gestures from captured data via tactile sensors. Other methods such as unsupervised or semisupervised learning may be performed to provide robust recognition. However, in this work, the classification of touch gestures is performed by improving the generalization capability of deep neural networks. The main objective of learning touch gestures is not only to minimize errors in the validation and test set of gestures as the current literature suggests but also to minimize generalization errors that is to have higher test accuracy than validation accuracy. Eventually, obtain more generalizable deep neural network model that takes as much as benefit from the training set.

In this thesis, we propose deep learning models to classify social touch gestures by learning spatiotemporal features of these gestures with the help of 3-dimensional convolution operation. There are plenty of models in the deep learning literature, the fundamental idea is to preserve natural local features of touch gestures and classify them. Hence, we propose four different deep neural network models to capture both spatial and temporal features together with the blocks of 3-dimensional convolution, long term short term memory and transformer. Using these four deep neural networks, the literature required to have unified model for touch gesture classification problem. In that regard, the CoST, the HAART, and our demo dataset are classified with changing only few parameters such as epoch, class labels, etc.

Several techniques are also proposed to improve generalization performance of networks by augmenting data with randomly shifted and rotated gestures, switching well-known optimization algorithm Adaptive Moment Estimation (ADAM) with Stochastic Gradient Descent (SGD) and ensemble classifier method to benefit from solution space of each networks. This way, we seek to test the performances of classification methods for shifted and rotated versions of touch gesture data.

In summary, the sensing and classification elements of touch gestures in Figure 1.1 are investigated with developed sensor settings using deep neural networks. The well-known publicly available datasets are classified to obtain an unified classifier.

CHAPTER 2

METHODS

Models that have been proposed in this section consist of two fundamental building block: the feature extractor and the classifier. The main feature extractor of the models is 3-dimensional convolution and the classifier is neural network. The models except 3-dimensional convolutional networks have additional temporal extractor just before classified. Eventually, the touch gesture sequences, which are resampled if necessary, are fed into the models to extract spatial and temporal features. Then, a neural network classifier makes a decision to assign labels to gestures.

2.1 Resampling

The proposed models require fixed sized inputs in temporal axis. Thus, the input gestures need to be resampled in the given length, otherwise it is not possible to train our models. The HAART and our dataset does not require resampling operation since they already have fixed length sequence in the temporal axis. But, the CoST dataset has varying size of gesture sequences. It is essential to make all gestures sequences in CoST dataset same length in temporal axis. Before the data in CoST fed into deep neural network, they were resampled to 200 frames which is the mean length of all gestures obtained by fitting Rayleigh distribution to length of all gesture sequences in CoST as shown in Figure 2.1. Rayleigh distribution is chosen since it is used for continuous probability distributions with non-negative valued random variables [46]. Some of gestures are downsampled and some of them are upsampled to obtain desired sequence length.

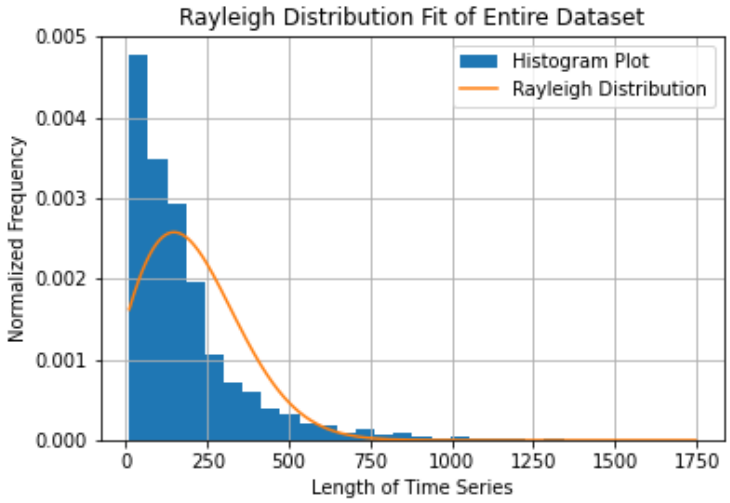


Figure 2.1: The Rayleigh distribution fit for length of entire CoST dataset

Converting a gesture sequence to a higher equivalent sampling rate is referred to as upsampling. The reversed form of upsampling is downsampling that is to convert a gesture sequence to a lower equivalent sampling rate. The upsampling process first starts by inserting intermediate points with zero amplitude between each of the values in the sequence. Then, an interpolation operation [70] is performed by lowpass filtering [45]. The downsampling operation is reversed process of upsampling. After resampling, there may be a loss of information in the touch gesture sequence.

2.2 Deep Neural Networks Architectures

This section contains detailed information about the building blocks of deep neural network architecture and how they are combined to classify touch gestures.

2.2.1 The Building Blocks of Architectures

The building blocks of deep neural network consist of two parts: feature extractor and classifier. 3-dimensional convolution, LSTM, BiLSTM and Transformer are the feature extractor in the deep neural network with the classifier of neural network. There are additional layer that play a significant role in the proposed architectures: 1) Pooling layer is to preserve invariant features in spatio-temporal axis 2) Dropout and spatial dropout layers are to regularize the proposed networks.

2.2.1.1 Artificial Neural Networks

The field of *artificial neural networks* has a rich history which is inspired from cognitive science neuroscience. The artificial neuron was firstly originated by McCulloch and Pitts in 1943 as a mathematical model of the biological neuron [43]. Their mathematical model was consisting of a perceptron with a bias, but did not have learnable weights in constrast to model which was invented by Rosenblatt [52] in 1958 had learnable weight which pioneered deep learning field as a subfield of machine learning in 2012 [36]. **Deep learning** uses neural networks as function approximator, with a stacking many layers of structurally similar components [51].

Artificial neural networks which nowadays are referred to as **neural networks** can be described as a particular set of composable functions derived from a overly simplified model of the brain. The **neural networks** start with a flexible set of functions $\{f(x;\theta\}$ constructed from parameterized by adjustable model parameters θ_μ in order to approximate original function $f(x;\theta) \approx f(x)$ where $f(x)$ represent the function that takes as input an dxd gesture sequence x_i and outputs the label of the gesture sequence.

The fundamental component of the neural networks is the neuron that consists of two operation: **preactivation** and **activation**.

$$z_i(s) = b_i + \sum_{j=1}^{n_{in}} W_{ij} s_j \tag{2.1}$$

$$\sigma_i \equiv \sigma(z_i) \tag{2.2}$$

where $i = 1, \ldots, n_{out}$ and $j = 1, \ldots, n_{in}$. The preactivation z_i of a neuron in Equation 2.1 is a linear combination of input signals of s_j which are weighted by W_{ij} and biased by b_i. Then, each neuron is activated in Equation 2.2 according to the value of the preactivation z_i. Here, the scalar-valued function $\sigma(\cdot)$ is defined as the **activation function**. Eventually, a layer of the neural networks is parameterized by a matrix of weights and a vector of biases together with a proper activation function.

Using the components at above, flexible set of functions can be constituted by organizing many neurons into a layer and stacking such layers in such a way that the outgoing vector of activation of the neurons in one layer feed into the neurons in the following layer. Such organization is known as **neural network architecture**, in a more modern name **fully-connected network**.

2.2.1.1.1 Activation Functions

The activation function at each layer of neural networks is chosen to be a nonlinear function in order to convey as much information as possible from one layer to another.

The **sigmoid** activation function is a smoothed version of the **perceptron** activation function [43]. The behaviour of original perceptron, which is just a step function, either fires and outputs 1 or does not fire and outputs 0. The sigmoid activation function is a logistic function as follows:

$$\begin{aligned}\sigma_{Sigmoid}(z) &= \frac{1}{1+e^{-z}} \\ &= \frac{1}{2} + \frac{1}{2}\tanh\left(\frac{z}{2}\right)\end{aligned} \tag{2.3}$$

It is not only a continuous function that maps from the domain of $(-\infty, \infty)$ to the range $[0, 1]$ but also it preserve information about the magnitude of preactivation. However, the sigmoid activation function is not a best choice for deep neural network architectures due to the fact that it does not pass through the origin.

The hyperbolic tangent or **tanh** activation function in Equation 2.4 is scaled and shifted version of the **sigmoid** activation function such that $\sigma(0) = 0$.

$$\begin{aligned}\sigma_{Tanh}(z) &= \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \\ &= \frac{e^{2z} - 1}{e^{2z} + 1}\end{aligned} \tag{2.4}$$

One of them rectifier linear unit **ReLU** [19][44] in Equation 2.5 is the most popular activation function used in deep neural networks and the reason behind its popularity is being a scale invariant activation function and its behavior at the origin. These properties allow the ReLU to create nonlinear relationship between inputs and outputs.

$$\sigma_{ReLU}(\cdot) = \begin{cases} x & \text{if} \quad x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.5}$$

Despite the popularity of ReLU, it is not a smooth activation function. The Gaussian Error Linear Unit (GELU):

$$\sigma_{GELU} = \left[\frac{1}{2} + \frac{1}{2}\text{erf}\left(\frac{z}{\sqrt{2}}\right)\right] \times z \tag{2.6}$$

where the error function erf(z) is given by

$$\text{erf}(z) \equiv \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt \tag{2.7}$$

The graph of error function of GELU is similar to the $\tanh$.

2.2.1.1.2 Pooling

Pooling operation is another important building block in Deep Neural Networks. The idea is firstly introduced and used in [39], [38]. The pooling operation reduces the size of feature map by using window and the purpose of window is to summarize information in the border of that window by taking the average or the maximum value.

The way of summarizing information works by sliding a window across the input. There are three parameter that affect the output size: input size, pooling window size and stride which means the distance between two consecutive window operations.

2.2.1.1.3 Dropout

Dropout is a regularization technique to reduce overfitting problem of neural networks [60]. It does not only reduce overfitting, but it also improves generalization error of model. During training phase of neural networks, some nodes in layers are randomly ignored or dropped out. Such random dropping activity offers reduced computational complexity as well. It can be used with fully connected layers, convolutional layers and recurrent layers.

Spatial dropout [65] is an extension of dropout technique which make sense in convolutional layer since the feature map is a 2-dimensional matrix. In case of applying spatial dropout to the feature map, a neighbour values in the gesture array can provide a highly correlated gradient. With the same principle as in the dropout, the spatial dropout randomly drops feature maps.

2.2.1.1.4 Loss Function

The **cross-entropy** loss is a measure of similarity between discrete distributions. In our models, these discrete distributions are the ground truth of gesture sequences and the estimated distributions at the out of deep neural networks.

$$\mathcal{L} = -\sum_{\delta \in \mathcal{D}} \sum_{i=1} p(i|x_\delta) log[q(i|x_\delta)] \tag{2.8}$$

where $p(i|x_\delta)$ is a discrete distribution of the ground truth or true output and $q(i|x_\delta)$ is a discrete distribution of the network's output. x is the input of network and δ is the index of

input. i is the component of output.

$$p(i|x_\delta) = \frac{\exp\left[y_{i;\delta}\right]}{\sum_{j=1} \exp\left[y_{i;\delta}\right]}$$
$$q(i|x_\delta) = \frac{\exp\left[z_{i;\delta}\right](t)}{\sum_{j=1} \exp[z_{i;\delta}(t)]} \tag{2.9}$$

The equation 2.9 which are the components of Equation 2.8 are sometimes referred as **softmax** function.

2.2.1.2 3-Dimensional Convolution

In order to learn spatio-temporal features, 3-dimensional convolution operation is proposed. The 2-dimensional convolution operation can only capture spatial features, whereas the 3-dimensional convolution operation preserves spatial and temporal information. Therefore, the feature learning is extended through temporal axis. As shown in Figure 2.2, applying $k \times k \times d$ dimensional filter enables us to feature mapping both spatial and temporal features.

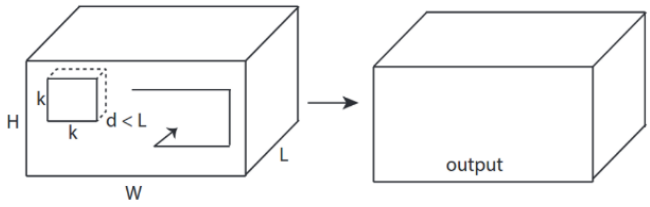


Figure 2.2: The input and output of 3D convolution operation reprinted by permission from [66]

The k represents the spatial size of convolution filter and d represents the temporal size of convolution filter. Each touch gestures in the datasets are able to represent with a size of $H \times W \times L$ where H and W are the height and width of the frame and L is the number of frames in the gesture sequence, respectively. During applying 3-dimensional convolution operation to touch gesture, the value at position (x, y, z) on the jth feature map in the ith layer is given by,

$$v_{i,j}^{x,y,z} = f\left(\sum_{m} \sum_{h=0}^{H_i-1} \sum_{w=0}^{W_i-1} \sum_{l=0}^{L_i-1} w_{i,j,m}^{h,w,l} v_{(i-1),m}^{(x+h),(y+w),(z+l)} + b_{i,j} \right) \tag{2.10}$$

where H and W represent spatial size of 3-dimensional kernel, L is the temporal size of 3 dimensional kernel, $w_{i,j,m}^{h,w,l}$ is the (h, w, l)th value of the kernel of the mth feature map. If padding is not used and the stride is 1, then the dimension of feature map is $(W - k + 1) \times (H - k + 1) \times (L - d + 1)$ generated by 3-dimensional convolutional layer.

The inherited disadvantages of the 3-dimensional convolution are computational complexity and excessive memory usage. However, with cutting-edge technology such as parallel computing, Graphic Processing Units, and Tensor Processing Units, those disadvantages can be overcome to a certain degree.

2.2.1.3 Long Term Short Term Memory

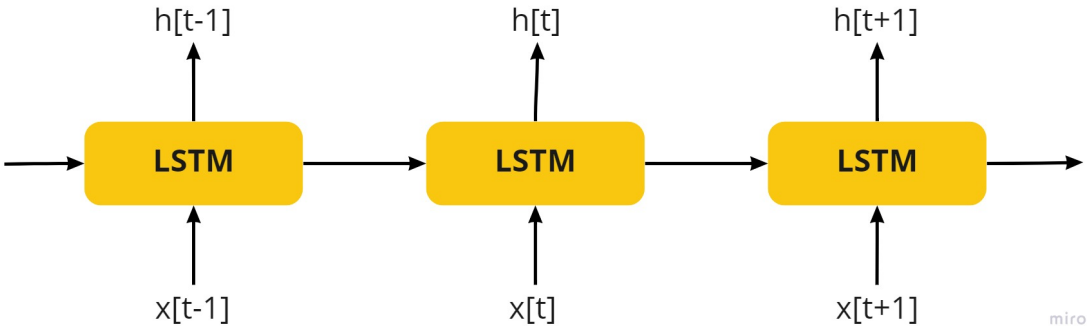


Figure 2.3: Structure of the LSTM

Long term short term memory networks (LSTMs) [24] are a special kind of recurrent neural networks (RNN) [53] that are capable of capturing long term dependencies in a sequence. The idea is quite straightforward and comes from autoregressive models, in order to understand present information the previous information might be useful. For instance, using previous frame of gesture sequence might inform the understanding of the present frame of the gesture sequence.

LSTMs have chain like structure composed of cells, similar to the RNNs. However, the repeating module of LSTMs has a different embedded structure. The key part of the structure is cell state that has ability to remove and add information along cell state by a structure called gates. The gates optionally enable information flow operated by activation function and element-wise multiplication operation.

There are three main gates in the LSTM structure: input gate, output gate, forget gate. The input gate determines the extent of information to be written onto the cell state, the forget gate determines to what extent to forget the previous data and finally, the purpose of the output gate is to decide which part of the cell state to output.

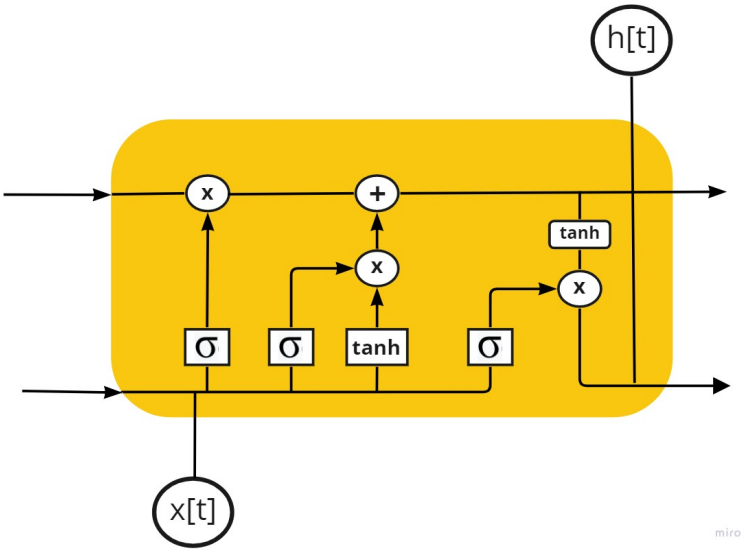


Figure 2.4: Inner structure of the LSTM

In Equation 2.11, f_t represents forget gate where x_t is the current input vector, h_{t-1} is previous hidden vector, W_f is weight matrices, b_f is bias vector. The forget gate makes a decision to keep information in the cell state with certain degree calculated by activation function. In the same equation, i_t represent input gate where what new information in the current cell is going to store. W_i is the weight matrices of input gate and b_i is the bias vector and $\widetilde{C}_t$ creates new candidate values that may be added to current cell state which is driven by $\tanh$ activation function. Combining the f_t, i_t and $\widetilde{C}_t$, the old state C_{t-1} is updated into C_t. As a final step at the cell, to decide what the output is going to be, o_t is employed which is also driven by an activation function and h_t. The summary of mentioned operation can

be found in the Figure 2.4

$$
\begin{aligned}
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
\widetilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\
C_t &= f_t * C_{t-1} + i_t * \widetilde{C}_t \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
h_t &= o_t * \tanh(C_t)
\end{aligned}
\tag{2.11}
$$

LSTMs are designed to overcome the long term dependency problem. That means, it can store the information over long periods of time. But, such feature heavily depends on the length of sequence. It also encodes long term dependencies by encoding one element at each time. Encoding in this way makes the LSTM more resistant to the vanishing gradient problem but make it impossible to parallelize the training process. So, LSTM requires more training time due to its of nature of encoding.

2.2.1.4 Transformer

The **transformer** architecture is used to process sequential input in such a way that takes advantages from correlation between elements in the sequence relying on the attention mechanism [67]. There are important elements in transformer architecture that provide us to capture long term dependencies in the sequence: positional encoding, residual connection, layer normalization, multi-head attention.

Transformers models encode the entire sequence at once and that yields losing critical information in the sequence, the order of sequence. That's why transformer models need to be able to capture positional information of elements of the sequence and input such

positional information to the encoder of transformer. The model should encode a unique representation for each time step of the sequence and should capture longer dependencies as in the LSTM. Using the below equation, the positional information of elements in the sequence:

$$\begin{aligned} PE_{(\mathrm{p},2i)} &= \sin\left(\mathrm{p}/10000^{2i/d}\right) \\ PE_{(\mathrm{p},2i+1)} &= \cos\left(\mathrm{p}/10000^{2i/d}\right) \end{aligned} \tag{2.12}$$

where PE represent positional encoding of corresponding element, p is the position of element in the sequence and i is the dimension. Thus, each dimension of the positional encoding represented as a sinusoid function.

Residual Connection serve two main purposes: knowledge preservation and avoiding vanishing gradient problem. During forward propagation of neural network, the inputs are modified considerably by the time they reaching the last layer. Such modification may result in the loss of the information that would present early on at the beginning layers. To resolve this issue, one solution is to add **residual connection** that bypasses the intermediate layers and feed information to deeper layers. This helps the deeper layer not to forget relatively important information that was present early on the system.

The layer normalization makes the network faster and more stable while optimizing. The operation is simply standardizing the neuron activations along the axis of features as described below,

$$x_i = \frac{x_i^d - \mu}{\sigma^2 + \epsilon} \tag{2.13}$$

where x_i is neuron activations, μ is mean of features, σ is the standard deviation of features and ϵ is a small value not to obtain zero in denominator.

Another important element in transformers is multi-head attention which contains

certain number of self-attention mechanism. The purpose of the multi-head attention is to filter out unnecessary features of the sequence by concentrating on different combination of features in the same sequence using the Equation 2.14:

$$\text{Attention}(Q, K, V) = \sigma_{\text{softmax}}\left(\frac{QK^T}{\sqrt{d_k}}V\right) \tag{2.14}$$

where Q is query, K is key, V is value and d_k is the dimension of query or key. Using Q, K and V, the unnnecessary features in the gesture sequence are filtered out. The query, key and values are identical vectors that are flattened versions of patches of gesture sequences. By taking the dot product of query, key and values, we obtain a matrix that contains score of each patch in terms of other patches.

As a summary of transformer, the gesture sequences are divided into patches or 3-dimensional submatrices. The patches and its positional information calculated by the Equation 2.12 and only patches are normalized with their mean and variance. Then, multi-head attention compares each patches in the gesture sequence and discriminate them by taking the dot product of query, key and values. Another layer normalization is performed to stabilize the network during training. The encoder part is finalized with neural network by feeding all parameter extracting from gesture sequences. One additional mechanism is necessary which is the residual connection that allows the encoder not to encounter the vanishing gradient problem. As a result, the encoder of transformers takes the input sequence and converts it into vectorized representation. This vectorized representation then feeds into the neural network to obtain class labels. The advantage of the network is to draw global dependencies between inputs and their labels by allowing parallelization, unlike LSTM.

2.2.2 Proposed Models

Four models are proposed using the building blocks which are introduced in Section 2.2.1. The architectures can be found in from Figure 2.6 to 2.9. The main block of each architecture is 3-dimensional convolution where the spatio-temporal features of touch gestures are extracted.

In the first architecture, the block that consists of 3-dimensional convolution, GeLU activation function, 3-dimensional pooling layer and Spatial Dropout layer is used twice before the neural network classifier. These two blocks works as feature extractor and the obtained features are flattened by Global Average Pooling layer. Thus, The data will be ready to classify. The classifier in the first architecture contains one input, one hidden layer and one output layer. Not to encounter overfitting, the Dropout layer is added between each layer.

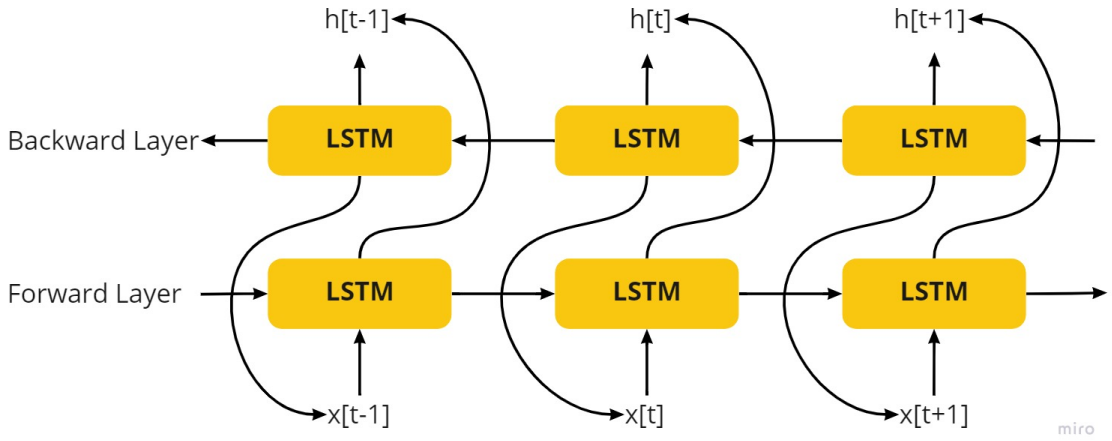


Figure 2.5: The structure of the BiLSTM

The second and third architectures in Figure 2.7 and 2.8 are similar to the 3D-CNN with one exception. The exception is the LSTM layer that is to extract additional temporal features from flattened layer. In the second architecture, LSTM layers is used, and in the third architecture the BiLSTM layer is used. BiLSTM layer in Figure 2.5 is capable of the

input flows in both directions, not only from first element to last element of the sequence but also from last element and first element of the sequence. The remaning architecture is the same as in the 3D-CNN.

In the fourth architecture, the combination of 3-dimensional convolution and transformers network is proposed to overcome the vanishing gradient problem of LSTM while the input sequence is longer as in the gesture sequence. The proposed architecture starts with 3-dimensional convolution, followed by transformers encoder and the neural network classifier. The intuition behind the 3D-CNN - Transformer model is to capture positional information of sequence, spatio-temporal features, global dependencies by attention mechanism together might be beneficial to increase generalization accuracy.

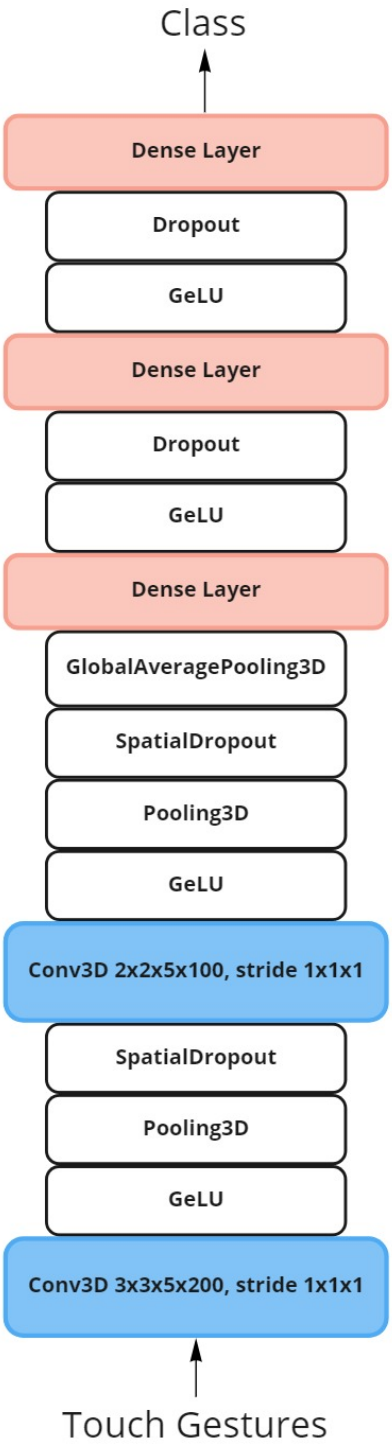


Figure 2.6: 3D-CNN

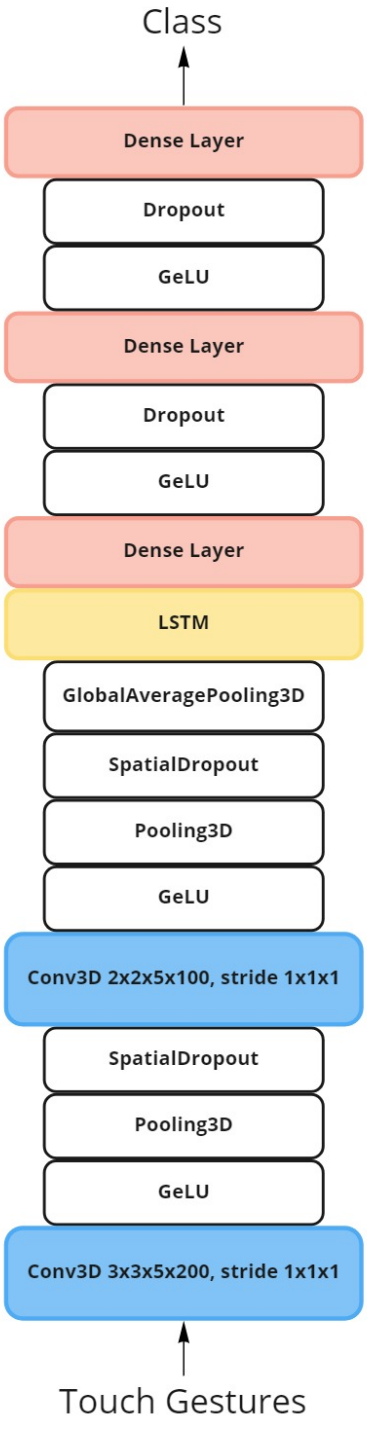


Figure 2.7: 3D-CNN + LSTM

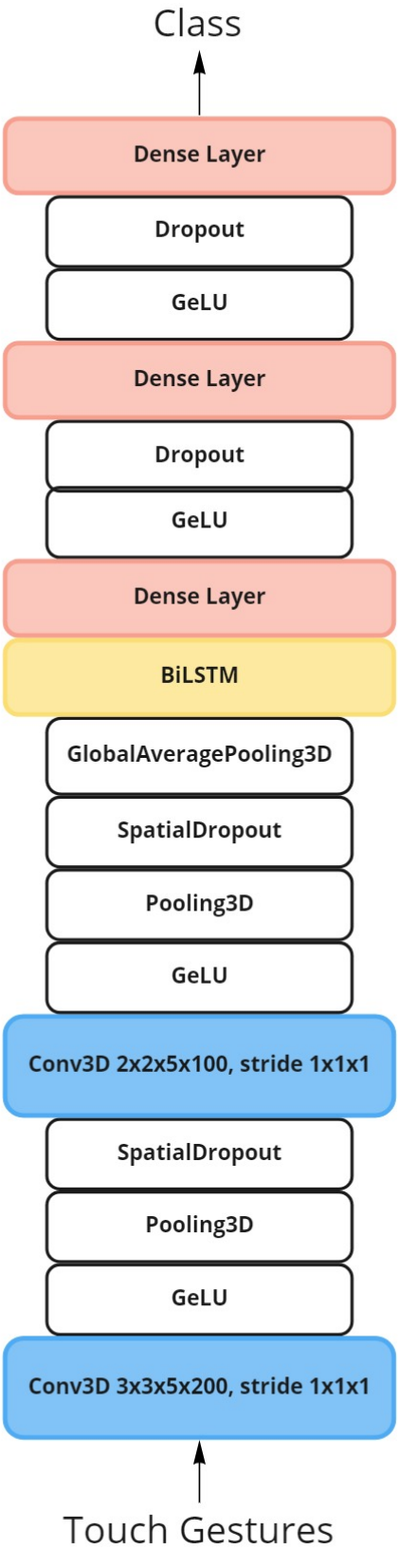


Figure 2.8: 3D-CNN + BiLSTM

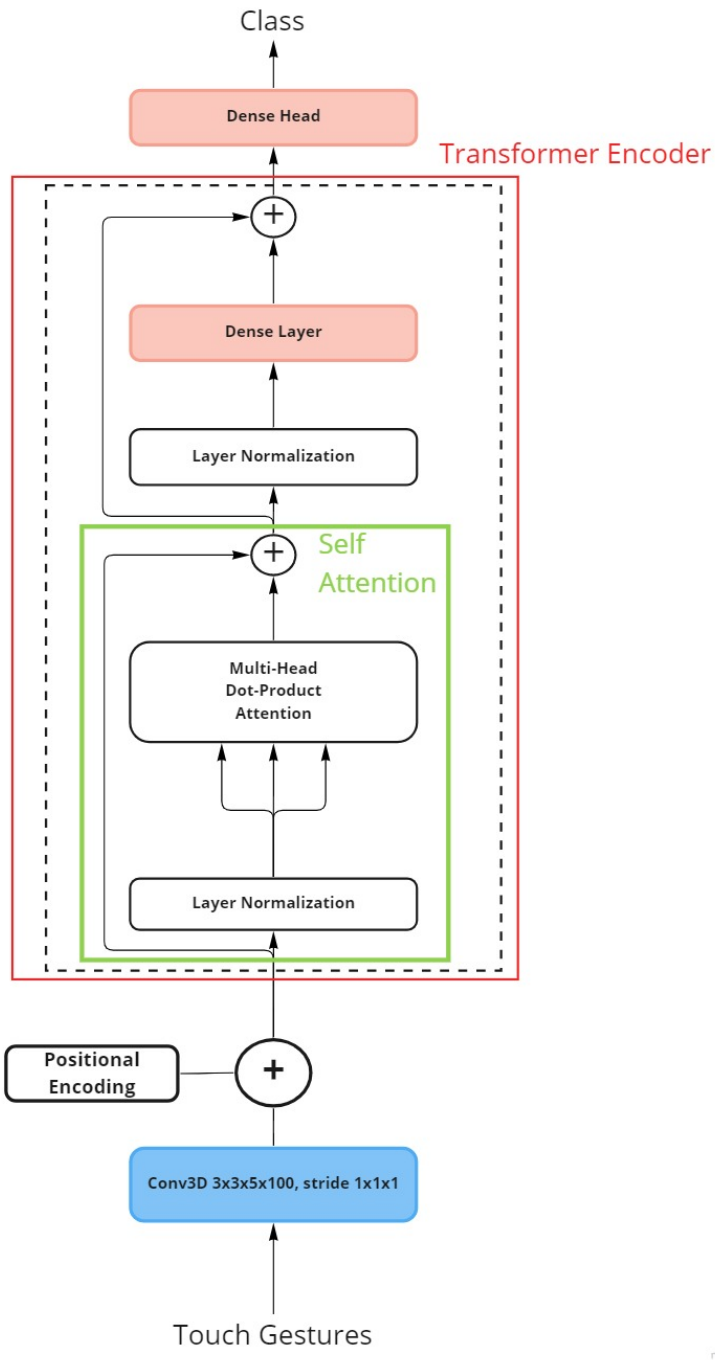


Figure 2.9: 3D-CNN + Transformer

2.3 Hyperparameter Tuning

The primary goal of hyperparameter tuning is to find the lowest generalization error of the learning algorithm, in other words, the effective capacity of a learning algorithm. The most important hyperparameter is learning rate since it affects the decision making by optimization algorithm [21]. Other hyperparameters of proposed models are filter size of 3-dimensional convolution, the pooling filter size, optimization algorithm, activation functions, batch size of training samples and number of epoch to train the models.

The candidate hyperparameters for proposed models; convolutional filter size: ((2,2,2), (2,2,5), (2,2,10), (3,3,3), (3,3,5), (3,3,10)), pooling filter size: ((2,2,2), (2,2,5), (2,2,10), (3,3,3), (3,3,5), (3,3,10)), learning rate: (0.01, 0.05, 0.001, 0.005, 0.0001, 0.0005), optimization algorithm: (ADAM, SGD), batch size: (64, 128, 256), activation functions: (GeLU, ReLU), loss function: (categorical cross entropy), epochs: (200, 300, 500, 600, 800). A grid search performed to determine best hyperparameters inspired [5][6]. Additionally, the proposed models are trained with Tensorflow Platform [1] using Google Colaboratory's NVIDIA P100 and T4 GPUs.

2.4 Optimization Algorithms

From optimization perspective, training deep neural networks can be formulate as non-convex optimization problem

$$\min_{w} f(w) := \frac{1}{M} \sum_{i=1}^{M} f_i(w)$$

where f_i is a loss function for data point $i \in \{1, 2, \cdots, M\}$ which captures the deviation of the model prediction from the data and w is the vector of weights of corresponding deep

learning model being optimized. Such non-convex problem can be iteratively solved using Stochastic Gradient Descent (SGD) [50]:

$$w_k = w_{k-1} - \alpha_{k-1}\left(\frac{1}{|B_{k-1}|}\sum_{i\in B_{k-1}} \nabla f_i(w_{k-1})\right) \tag{2.15}$$

where w_k denotes weights in the k^{th} iteration, α_k is a step size sequence, also called the learning rate, $B_k \subset \{1, 2, \cdots, M\}$ is the batch sampled from the data set and $\nabla f(w_k)$ denotes the stochastic gradient computed at w_k. SGD uses only one learning rate for all gradient coordinates as in Equation (2.15) and could suffer from slow convergence rate since it scales uniformly the gradient of all parameters of the model during training [34].

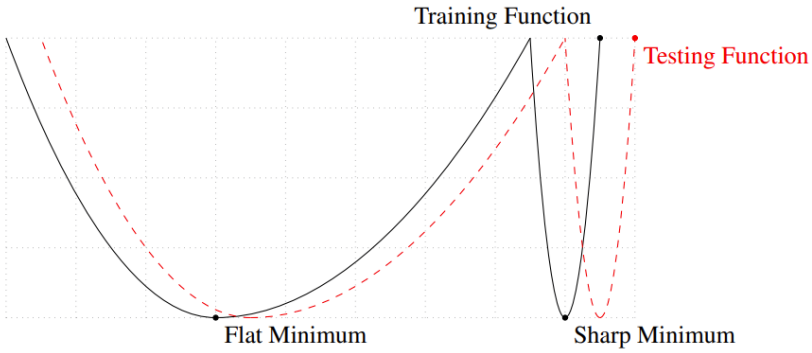


Figure 2.10: A conceptual sketch of flat and sharp minimum, reprinted with permission from [33]

In order to avoid shortcomings of SGD, several adaptive methods have been proposed that adjust learning rate for each gradient coordinates such as Adam [35], AdaGrad [12] and RMSprop [64]. These optimization methods scale the gradient diagonally using estimates of the function's curvature and much faster convergence rate than SGD. Adam and its

update equations can be formulated as follows:

$$w_k = w_{k-1} - \alpha_{k-1} \cdot \frac{\sqrt{1-\beta_2^k}}{1-\beta_1^k} \cdot \frac{m_{k-1}}{\sqrt{v_{k-1}+\epsilon}} \tag{2.16}$$

where

$$\begin{aligned} m_{k-1} &= \beta_1 m_{k-2} + (1-\beta_1)\hat{\nabla} f(w_{k-1}) \\ v_{k-1} &= \beta_2 v_{k-2} + (1-\beta_2)\hat{\nabla} f(w_{k-1})^2 \end{aligned} \tag{2.17}$$

In Equation (2.16) and (2.17), $\beta \in [0, 1)$ is a momentum parameter and v is the accumulator which is initialized to 0. ϵ is a constant to make sure the denominator not to equal 0. The momentum parameter β_1 is generally kept around 0.9 while β_2 is kept at 0.99. Epsilon is chosen to be 10^{-10} generally.

Adaptive Moment Estimation (Adam) has been used in many deep learning applications because of performing well with requiring minimal tuning. However, adaptive methods lack of ability to generalize compared to Stochastic Gradient Descent (SGD) since the loss surface of deep neural networks tends to have many local minima and many of these might be equally good in terms of training error, but they may have different generalization performance [71] [76]. The lack of generalization of adaptive methods is caused by tending to converge to sharp minimizers of the training function that is due to the number of large positive eigenvalues of $\nabla^2 f(w)$. In contrast, SGD converges to flat minimizers of training function [33] [68] since the loss surface indicates numerous small eigenvalues of $\nabla^2 f(w)$ as in the Figure 2.10 where the x-axis represents the parameters of

deep neural network model and the y-axis represents value of loss function.

2.5 Ensemble Classifier

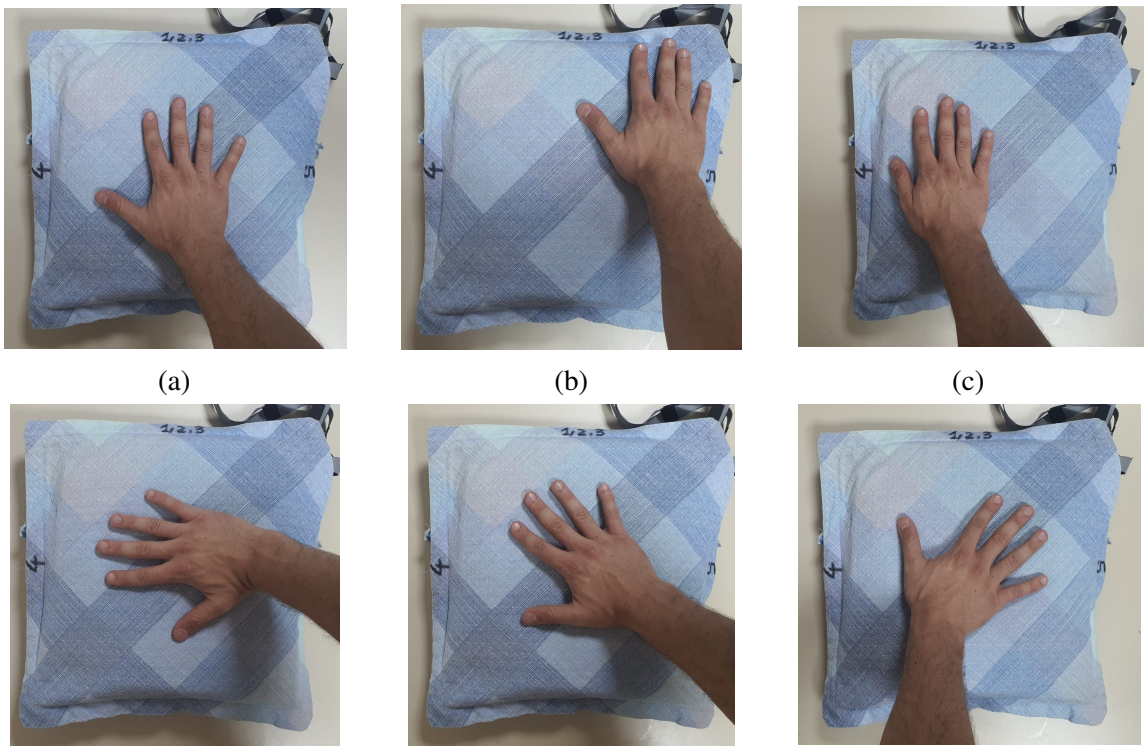
Ensemble classifier is to build a prediction model by combining the strengths of multiple single classifiers. The main advantage is that ensembles are often much more accurate than the single classifiers, but they are computationally infeasible, especially with deep neural networks having lots of parameters. The rationale behind ensemble methods is that single classifiers have the uncorrelated error rate.

To improve prediction performance and generalization of single model, a relatively simple search algorithm **grid search** is applied to ensemble the proposed classifiers. The search space was defined as a grid of weights of each model to achieve best combination of proposed 4 model that give maximum accuracy and feasible solution was obtained by evaluating every position in the grid.

The grid of ensemble classifier has four dimensional since there are four proposed model. Each dimension divided into 10 points and the best point that gives maximum accuracy is searched by grid search algorithm. Then, the weight of each proposed model is determined according to corresponding point in the axis of grid.

2.6 Data Augmentation

To obtain more generalizable deep neural networks, the best way to train the model with more data. In practice, there are limited amount of data. So, **data augmentation** is a technique that is used to increase the amount of data in the training dataset by adding modified copies of existing data [39]. Adding modified copies to the dataset helps reduce overfitting during the training phase as well.



(a) (b) (c)

(d) (e) (f)

Figure 2.11: Examples of rotation and shift of touch gestures

The neural network as a classifier encodes information by reshaping input data into a vector, so it is difficult to preserve equivariant features, even invariant features. However, the neural networks learn to classify data by training on a significant amount of data. It is natural to expect to give better results. But this highly depends on the quality of training data. With the data augmentation, the neural networks may learn some equivariant or invariant features [41]. In other words, the classifier is invariant to a wide variety of transformations.

There are various types of augmentation techniques such as rotation, shift, flip, saturation, cropping, etc. In this thesis, only the rotation and the shifted versions of touch gestures are taken into account and fed into deep neural networks. The augmentation should be applied only to limited number of samples in the training set of the datasets. In the ideal scenario, each transformation should be applied to randomly selected 25% of dataset. For instance, the augmentation by rotation should be applied to randomly selected 25% of dataset and same procedures should be repeated for the augmentation by shift. As a final step, all transformed touch gesture data is added to the training set with its labels

before it was transformed. Some of the examples of data augmentation can be seen in Figure 2.11: (a) Touch without, (b) Right and upper-shifted version, (c) Left-shifted, (d) -45° degree rotated version, (e) $+45^{\circ}$ degree rotated version.

CHAPTER 3

RESULTS

3.1 Results on CoST Dataset

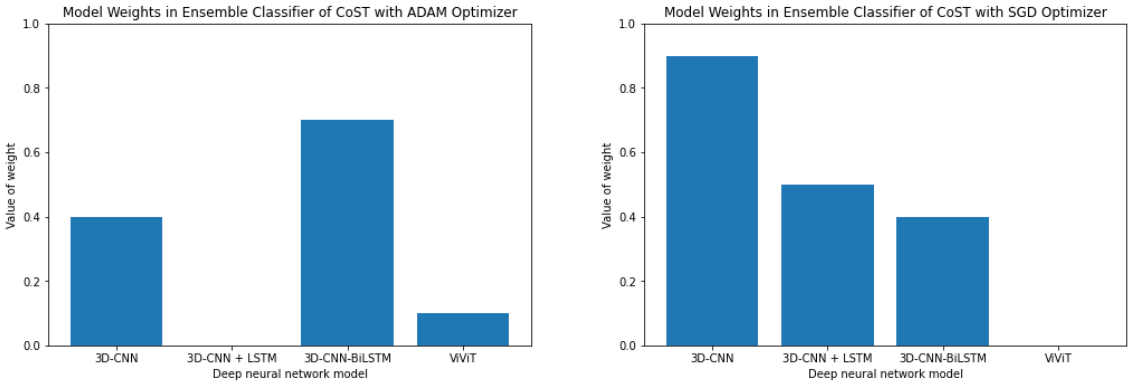
The Table 3.1 shows that the validation and test accuracies of each architecture and their ensemble accuracies with respect to ADAM and SGD optimization algorithm. The results are obtained by following the protocol in Social Touch Gesture Challenge. In the challenge, the training subjects are 1, 2, 3, 5, 6, 7, 10, 12, 13, 14, 16, 18, 19, 21, 22, 24, 25, 26, 27, 28, 30 and the test subjects are 4, 8, 9, 11, 15, 17, 20, 23, 29, 31. The variants that were taken into account in the challenge were gentle and normal gestures, not include gestures that are performed in rough manner.

Table 3.1: The percentage maximum accuracies for CoST dataset

	ADAM		SGD	
Architecture	Val. Acc.	Test Acc.	Val. Acc.	Test Acc.
3D-CNN	55.46	55.27	55.74	56.82
3D-CNN - LSTM	56.31	52.89	58.72	55.69
3D-CNN - BiLSTM	56.74	55.27	55.89	55.15
3D-CNN - Transformer	43.55	40.14	37.54	35.24
Ensemble Classifier	-	**57.83**	-	**59.14**

The above table contains the validation accuracies and test accuracies together. During training phase, validation set is always chosen randomly from training set by the amount of 20%. The test accuracy is always calculated after training phase is done. The purpose of giving validation results and test results together are to emphasize generalizability

of the proposed models. The validation accuracy of ensemble classifier cannot be reported since the ensemble classifier takes only account for best combination of networks which is not trainable due to choosing best weights of model. The obtained results in the table is the average accuracy of 10 randomized trial. The number of epoch is 200 for ADAM optimizer, 300 for SGD optimizer.



(a) CoST Dataset with ADAM Optimizer (b) CoST Dataset with SGD Optimizer

Figure 3.1: The model weights of ensemble classifier for CoST dataset

In the Table 3.1, among all proposed architectures, 3D-CNN-Transformers networks classified the gestures 40.14% with ADAM and 35.24% with SGD. These are the lowest accuracies among all reported result. However, 3D-CNN and 3D-CNN-LSTM results are the best accuracies. By examining generalizability of architectures, the 3D-CNN with SGD optimizer has passed the validation accuracy. Additionaly, it can easily seen that Ensemble methods are benefical to generalize in terms of both optimizer. In both cases, ensemble accuracies are better than any other reported validation accuracies.

Furthermore, the weights of the proposed classifiers while generating ensemble classifier are shown in Figure 3.1. Ensemble classifier with ADAM took advantage from 3D-CNN-BiLSTM, 3D-CNN and 3D-CNN - Transformer. 3D-CNN-BiLSTM classified as best, so it is expected to take big role in ensemble. However, The result of 3D-CNN - Transformer classifier was the lowest accuracy and took a little role in ensemble classifier. Although the 3D-CNN-LSTM obtains higher accuracy than 3D-CNN - Transformer, it has 0 weight in ensemble. Another ensemble classifier with SGD contains 3D-CNN, 3D-CNN-LSTM,

3D-CNN-BiLSTM architectures, not the 3D-CNN - Transformer architecture.

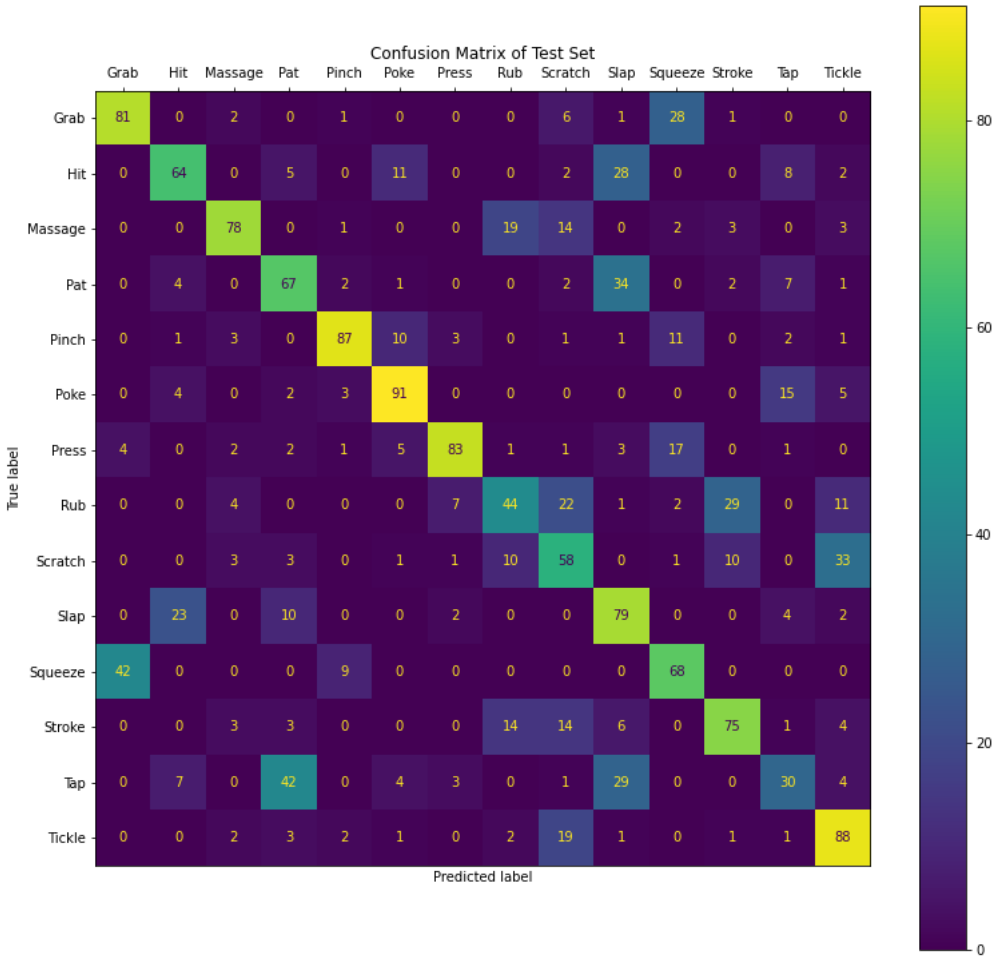


Figure 3.2: Confusion matrix for test set of CoST dataset

The Figure 3.2 depicts the confusion matrix of CoST test set with ensemble classifier with SGD optimization algorithm. The horizontal axis represents predicted labels and the vertical axis represents the ground truth of gestures. According to the confusion matrix at above, the most predictable touch gestures are poke, tickle and pinch, whereas

tap, rub and scratch are the most confusable gestures. The tap gesture is confused with pat and slap. The ensemble classifier identifies rub gestures as stroke, scratch and tickle. The third confused gesture is scratch which is classified mostly as tickle, rub and stroke.

Another method to increase generalizability was data augmentation of touch gestures. The test set of CoST dataset was identical in the experiment of augmented and non-augmented dataset. The only difference during experiment was training dataset. The results belongs to each classifier can be found in the Table 3.2. The results are similar to the non-augmented dataset except in 3D-CNN - Transformer architecture and Ensemble classifier. The ensemble classifier are better than ensemble classifier with non-augmented dataset.

Table 3.2: The percentage test accuracy of augmented CoST datasets

Architecture	Augmented CoST
3D-CNN	55.81%
3D-CNN - LSTM	56.68%
3D-CNN - BiLSTM	49.67%
3D-CNN - Transformer	53.96%
Ensemble Classifier	**63.58%**

In the Table 3.3, leave-one-out-subject cross validation results of ensemble classifier with SGD are reported. Accuracies are obtained by testing with only corresponding subject on trained remaining subjects. The best accuracy belongs to subjects 3 and 10 with 78.18% accuracy. The lowest accuracies are belongs to the subjects 23 and 14 with approximately 35%. The average accuracy of all subjects in the dataset is 59.72%.

Table 3.3: Leave-one-out-subject cross validation percentage accuracy for CoST

Test Subjects	1	2	3	4	5	6	7	8
Accuracy	67.48	50.81	78.19	71.04	79.38	54.97	68.46	72.23
Test Subjects	9	10	11	12	13	14	15	16
Accuracy	62.12	78.19	62.71	52.33	52.92	35.34	46.83	49.62
Test Subjects	17	18	19	20	21	22	23	24
Accuracy	65.10	69.47	45.11	50.50	50.50	66.88	34.14	58.54
Test Subjects	25	26	27	28	29	30	31	Average
Accuracy	45.71	75.81	62.71	51.40	56.16	57.95	63.90	**59.72**

3.2 Results on HAART Dataset

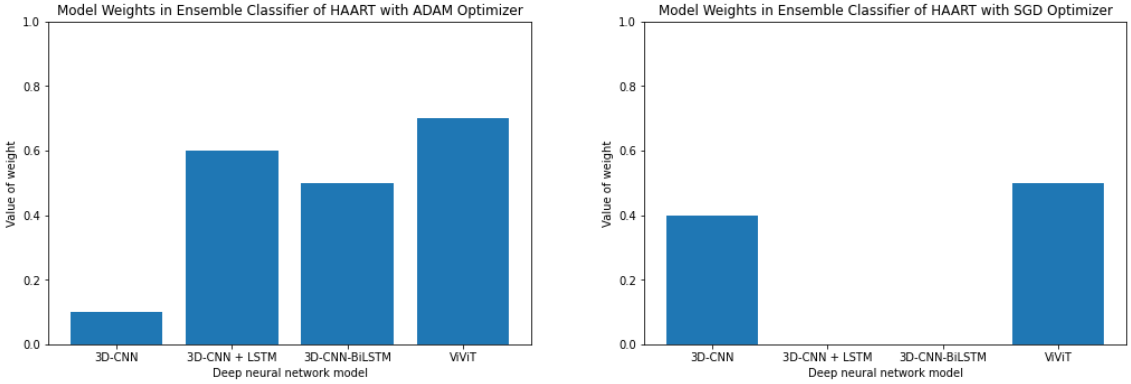
The Table 3.4 shows that that the validation and test accuries of each arhitecture and their ensemble accuracies with respect to two optimization algorithm for HAART dataset. The results are obtained by following the protocol in Social Touch Gesture Challenge. In the challenge, the training subjects are 3, 4, 5, 6, 7, 8, 10 and the test subjects are 1, 2, 9. All conditions and substrates are considered, none of them excluded from dataset.

Table 3.4: The percentage maximum validation accuracy and test accuracy obtained by proposed models in terms of ADAM and SGD Optimizer for HAART dataset

Architecture	ADAM		SGD	
	Val. Acc.	Test Acc.	Val. Acc.	Test Acc.
3D-CNN	64.66	62.15	67.24	63.35
3D-CNN - LSTM	68.97	61.35	62.93	53.78
3D-CNN - BiLSTM	50.86	49.40	63.79	53.39
3D-CNN - Transformer	63.79	66.14	69.83	70.52
Ensemble Classifier	-	**69.33**	-	**72.51**

The Table 3.4 contains the validation accuracies and test accuracies together. During training phase, validation set is always chosen randomly from training set by the amount of 20%. The test accuracy is always calculated after training phase is done. The purpose of giving validation results and test results together are to emphasize generalizability of the proposed models. The validation accuracy of ensemble classifier cannot be reported since the ensemble classifier takes only account for best combination of networks which is not trainable due to choosing best weights of model. The number of epoch is 500 for ADAM optimizer, 800 for SGD optimizer. The obtained results for HAART dataset are the average accuracies of 10 randomized trial.

The best accuracy of test set of HAART belongs to 3D-CNN - Transformer classifier with ADAM and SGD which is 66.14% and 70.52%. The lowest one is 49.40% that belongs to 3D-CNN-BiLSTM with ADAM. The test accuracies obtained by LSTMs layers are the lowest ones for both optimization algorithm. The ensemble method classified the test dataset 69.33% with ADAM and 72.51% with SGD. SGD again outperformed the results with ADAM.



(a) HAART Dataset with ADAM Optimizer (b) HAART Dataset with SGD Optimizer

Figure 3.3: The model weights of ensemble classifier for HAART dataset

The weights of ensemble method can be found in the Figure 3.3. The weights of models with SGD are correlated with the accuracies. 3D-CNN - Transformer classifier has major impact on ensembles. The 3D-CNN-LSTM has also higher weight in ensemble with ADAM, whereas no impact in ensemble with SGD. Another higher weight in ensemble

with ADAM belongs to 3D-CNN-BiLSTM and no impact with other optimizer.

The confusion matrix of test set of HAART is shown in the Figure 3.4. The matrix includes the classification result of ensemble classifier with SGD. By examining the confusion matrix, the most confusable gestures are rub and tickle. They are confused with stroke and scratch with almost same amount. The most successfully predicted gestures are constant, no touch and pat. The prediction of scratch and stroke are in the moderate level. Mostly, scratch has been confused rub and tickle. The gesture of stroke is rarely confused with pat and rub according to figure.

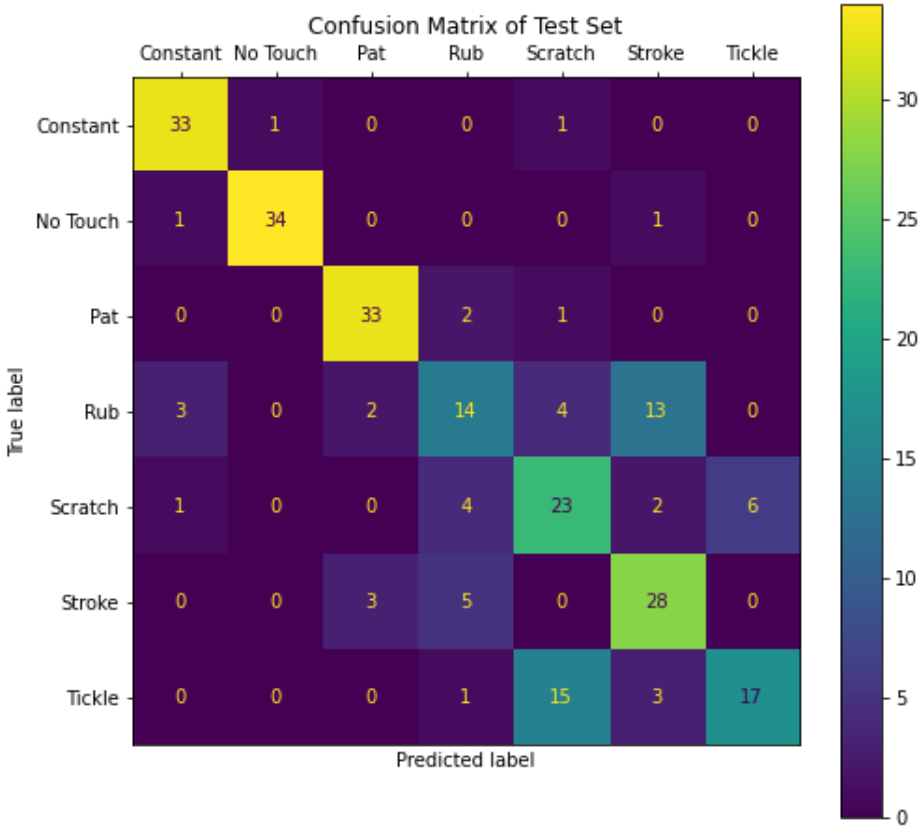


Figure 3.4: Confusion matrix for test set of HAART dataset

Table 3.5: The percentage test accuracy of augmented HAART datasets

Architecture	Augmented HAART
3D-CNN	62.55%
3D-CNN - LSTM	64.14%
3D-CNN - BiLSTM	60.95 %
3D-CNN - Transformer	67.73%
Ensemble Classifier	**74.10%**

The method of augmented training set increase the accuracy of ensemble classifier but decrease the 3D-CNN and 3D-CNN - Transformer accuracies and increase the 3D-CNN - LSTM and 3D-CNN - BiLSTM compared to pure training set. It is clearly be seen that generalizability property has been improved by proposed models in Table 3.5.

Table 3.6: Leave-one-out-subject cross validation percentage accuracy for HAART dataset

Test Subjects	1	2	3	4	5	
Accuracy	14.28	14.28	14.45	14.28	14.28	
Test Subjects	6	7	8	9	10	Average
Accuracy	14.45	14.46	14.28	13.25	14.46	**14.24**

The Table 3.6 shows that leave-one-out-subject cross validation results of entire dataset. The accuracies almost similar in all subjects. The difference in accuracies depends on the size of test set. Some of gestures were missing as default in the dataset. The average accuracy is 14.24 that is quite lower than intentionally selected test set in the Table 3.6. Adding test set into training set go down the overall accuracy. Due to the less number of samples in the HAART dataset, the proposed models was biased to specific gestures. They only recognize touch without movement and no touch.

3.3 Results on Our Demo Dataset

The results that reported in this section is classification results of our demo dataset. The Table 3.4 shows that the validation and test accuracies of each architecture and their ensemble accuracies with respect to two optimization algorithm. From Table 3.8 to 3.11, the leave-one-out-subject cross validation results are reported.

The first attempt of classification contained all collected gestures in our dataset and the accuracies of classification was almost zero. To overcome the lower accuracy problem, the cross validation is performed to find out what are the most efficient gestures in our dataset. Thus, the scratch and grab gestures are excluded from our collected data.

The Table 3.7 contains the validation accuracies and test accuracies together. During training phase, validation set is always chosen randomly from training set by the amount of 10% because of having less gesture data. The test accuracy is always calculated after training phase is done. The purpose of giving validation results and test results together are to emphasize generalizability of the proposed models. The validation accuracy of ensemble classifier cannot be reported since the ensemble classifier takes only account for best combination of networks which is not trainable due to choosing best weights of model. The number of epoch is 50 for ADAM optimizer, 100 for SGD optimizer. The obtained results for HAART dataset are the average accuracies of 10 randomized trial.

Table 3.7: The percentage maximum validation accuracy and test accuracy obtained by proposed models in terms of ADAM and SGD Optimizer for our demo dataset

Architecture	ADAM		SGD	
	Val. Acc.	Test Acc.	Val. Acc.	Test Acc.
3D-CNN	25.00	45.00	55.00	50.00
3D-CNN - LSTM	45.00	25.00	35.00	25.00
3D-CNN - BiLSTM	40.00	37.50	65.00	52.50
3D-CNN - Transformer	37.50	25.00	40.00	20.00
Ensemble Classifier	-	**57.50**	-	**67.50**

The Table 3.7 shows the test set results of our demo dataset. The training set consist of subjects: 1, 2, 4 and 5. The test set only contains the gestures of subject 3. The ensemble classifier with SGD outperform other methods. It is not clearly seen the improved generalization from individual results but can be seen in the both ensemble classifier. The weights of ensemble classifier with ADAM optimizer 3D-CNN with 0.3, 3D-CNN - LSTM with 0, 3D-CNN - Transformer with 0.1, 3D-CNN - BiLSTM with 0.8. The weights of ensemble classifier with SGD optimizer 3D-CNN with 0.2, 3D-CNN - LSTM with 0, 3D-CNN - Transformer with 0, 3D-CNN - BiLSTM with 0.5. The confusion matrices belong to these ensemble classifiers reported in the Figure 3.6 and 3.5.

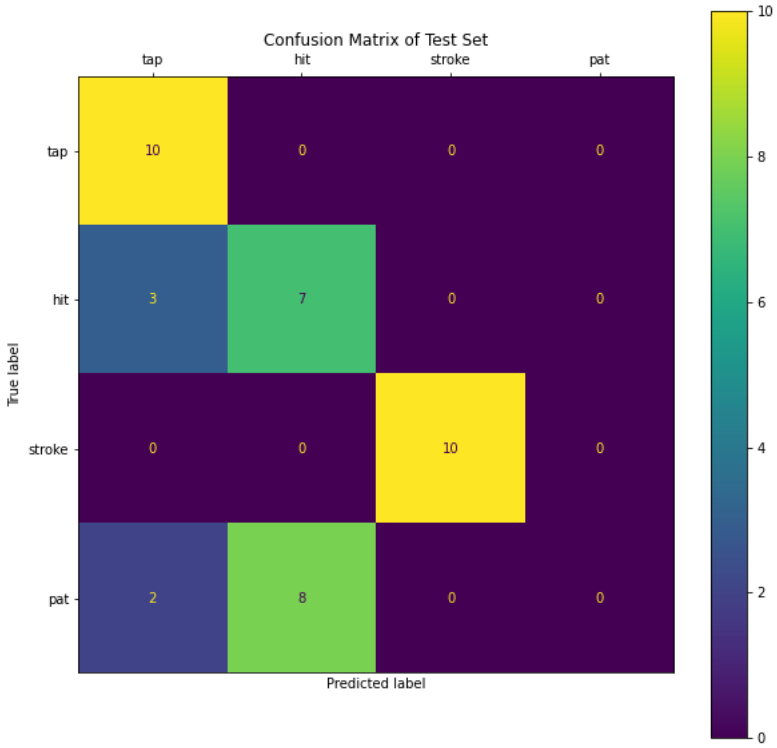


Figure 3.6: Confusion matrix for test set of our demo dataset in result of ensemble classifier with SGD

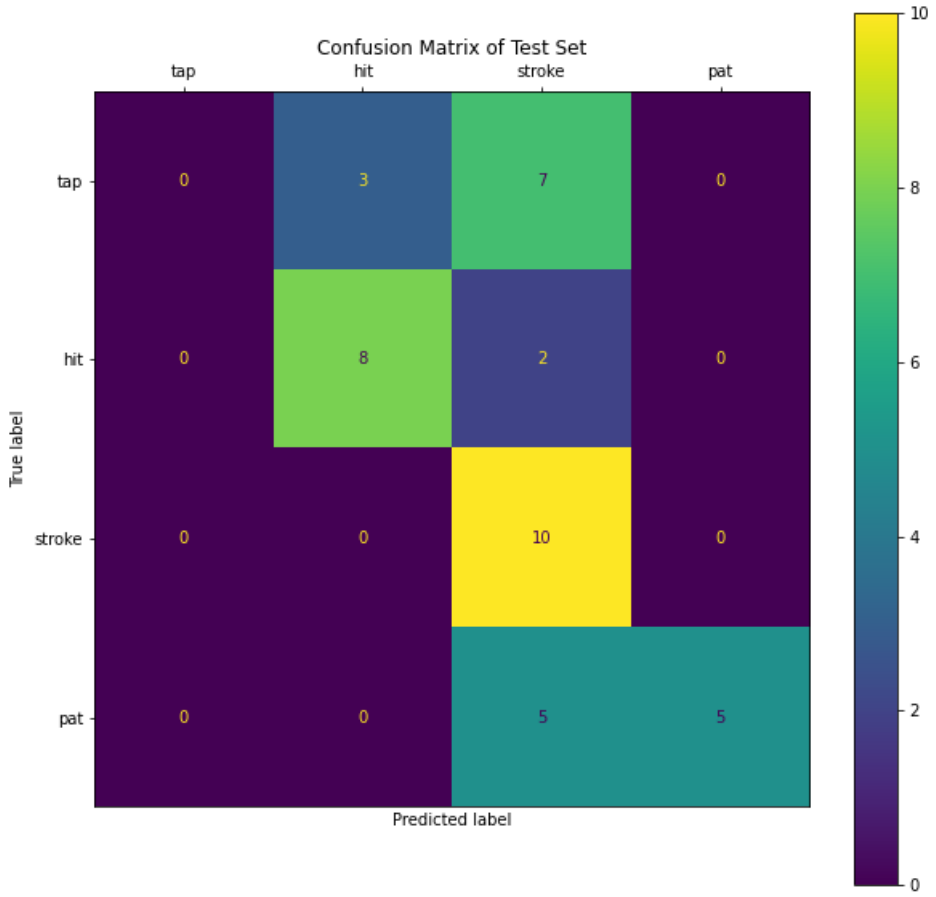


Figure 3.5: Confusion matrix for test set of our demo dataset in result of ensemble classifier with ADAM

The obtained confusion matrix have in common results which is the stroke gesture that is classified perfectly. Another perfect classification results is the hit gesture obtained with ensemble classifier with SGD. The pat gesture 50% classified with ADAM and totaly confused with the method of SGD where as the tap gesture are totaly confused with hit and stroke gestures.

Table 3.8: Leave-one-out-subject cross validation percentage accuracy for our demo dataset with ADAM optimizer

Test Subjects	1	2	3	4	5	Average
Accuracy	50.00	58.00	57.50	42.00	50.00	**51.50**

Table 3.9: Leave-one-out-subject cross validation percentage accuracy for our demo dataset with SGD optimizer

Test Subjects	1	2	3	4	5	Average
Accuracy	52.00	64.00	67.50	34.00	64.00	**56.30**

At the above Table 3.8 that is cross validation results of ensemble classifier with ADAM, the average accuracy is 51.50% percent ranging from 50% to 57.50%. The subject 2 is the best test set among other subjects. The results of leave-one-out-subject cross validation can be seen in the Table 3.9 where the subject 3 is the best test set.

Table 3.10: Leave-one-out-subject cross validation percentage accuracy for augmented our demo dataset with ADAM optimizer

Test Subjects	1	2	3	4	5	Average
Accuracy	52.00	58.00	64.00	50.00	50.00	**54.80**

Table 3.11: Leave-one-out-subject cross validation percentage accuracy for augmented our demo dataset with SGD optimizer

Test Subjects	1	2	3	4	5	Average
Accuracy	58.00	52.00	70.00	50.00	52.00	**56.40**

The cross validation results with augmented training set can be found in Table 3.10 and 3.11. The results with SGD are still the best results but average result is lower than without augmentation. The subject 3 is again the best test set among other subjects which has minimum accuracy of 52.00% and maximum accuracy of 70.00%.

CHAPTER 4

DISCUSSION

In order to evaluate proposed methods, a comparison is made with other results in the literature in Table 4.1 for CoST dataset and the Table 4.2 for HAART dataset. Two method have been evaluated, the ensemble classifier and ensemble classifier with data augmentation. In the comparison of literature results of CoST dataset, they did not achieve the best accuracy, but the ensemble classifier with data augmentation are in second place after Li et al. [42].

Table 4.1: Evaluation of CoST: Benchmark Results and Proposed Method

Paper	Classifier	Accuracy
Ta et al.[63]	Random Forest	61.3 %
Gaus et al.[17]	Random Forest	58.7 %
Hughes et al. [25]	Logistic Regression	47.2 %
Balli et al.[4]	Random Forest	26.0 %
Jung et al. [30]	SVM	60.0 %
Hughes et al. [26]	2D-CNN	42.3 %
Hughes et al. [26]	2D-CNN - RNN	52.8 %
Li et al. [42]	Random Forest	64.17%
Bani et al.[13]	ATM	60.9 %
Proposed method	**Ensemble Classifier**	**59.14**%
Proposed method	**Augmentation + Ensemble Classifier**	**63.58**%

In the evaluation of HAART dataset, both proposed methods took second place in terms of accuracy after Zhou et al. [75]. Zhou et al. have used 3D-CNN, same architecture with the proposed method in this thesis, and achieved the best accuracy. Bani et al. [13]

prefer to implement Transformers architecture and obtained 67.8 accuracy. Their method only took into consideration temporal features of touch gestures. The obtained result with 3D-CNN - Transformer architecture is better than theirs, as well.

Table 4.2: Evaluation of HAART: Benchmark Results and Proposed Method

Paper	Classifier	Accuracy
Ta et al.[63]	Random Forest	70.9 %
Gaus et al.[17]	Random Forest	66.5 %
Hughes et al. [25]	Logistic Regression	67.7 %
Balli et al.[4]	Random Forest	61.0 %
Zhou et al.[75]	3D-CNN	76.1 %
Hughes et al. [26]	2D-CNN	56.1 &
Hughes et al. [26]	2D-CNN - RNN	61.3 &
Bani et al.[13]	ATM	67.8 %
Proposed method	**Ensemble Classifier**	**72.51**%
Proposed method	**Augmentation + Ensemble Classifier**	**74.10**%

Classification results of our demo dataset and both publicly available dataset show that the confusable gestures have common features: the definition of gestures can be easily misunderstood by subject without strictly defined. Because, most of the mislabeled gestures might be confusable with similar gesture in terms of movement such as poke and tap, rub and stroke, hit and slap. Other factor rather than properly understanding the definition of gestures and having similar movements by performing them could be cultural differences among subjects.

While during experiment to achieve better accuracy and hyper-parameter tuning of proposed models, the networks came across with the problem of overfitting and underfitting. To overcome such problems, the dropout have been used in the proposed architectures; even if the popular solution to overfitting using the dropout layer and batch normalization layer together in the way that convolution layer, batch normalization layer, activation function and dropout layer [28][60]. In the touch gesture classification, batch normalization did not

work, it only increase training time.

On the one hand, from the results of CoST and HAART dataset several implications can be made. Selection of proper subject highly affects the classification result. The subject biases can be clearly observed in the leave-one-out-subjects cross validation results. It is seen that the test set of both CoST and HAART dataset are intentionally chosen which is significantly low accuracies and hardly generalizable. A lot of outlier gestures can be observed from confusion matrices.

On the other hand, the results of our demo dataset showed us that our sensor setting does not capture variety of touch gestures. Only specific gestures can be captured and classified moderately. The measurement of touch gesture probably contains noisy data, so they are not discriminated from each other.

We also trained with one dataset and tested with another dataset to figure out how beneficial to try out-of-distribution training. The result of this experiment are not reported since no significant results were observed.

The results show that the data augmentation is a beneficial method for three dataset, especially, with more data such as in case of CoST and HAART dataset. With the limited amount of data, the deep neural networks might turn biased classifier.

There are variety of ensemble methods in the literature [22]. One may decrease the generalization error of test set by using them. We have chosen one of the simple method, the grid search strategy, is just to show that the combination of proposed model can outperform compared to the single model. According to the results, ensemble classifier and ensemble classifier with data augmentation yields more generalizable accuracies then single model.

However, the purpose of choosing one of the simple method, the grid search strategy, is to show that the combination of models can outperform compared to one model at each time. According to result of ensemble classifiers, the both cases which are only the ensemble classifier and the ensemble classifier with data augmentation yields more generalized test set than one proposed model.

It is observed that Stochastic Gradient Descent (SGD) has well-generalizable optimization algorithm than Adaptive Moment Estimation. In most of the results, it outperformed the ADAM. However, it requires more epoch to train since its convergence rate is slow.

The another observation from the results is that LSTM and BiLSTM are better at relatively shorter sequences. The gestures in HAART dataset have longer sequences than other two dataset. The temporal size is more than double of HAART. Compared to other architecture, LSTM has lower accuracies in HAART that is why the 3D-CNN - Transformer classifier is proposed, to capture long term dependencies.

Even if the classification results of our demo dataset is promising, encoding gestures at the sampling rate of 10 Hz may not be sufficient. The actual sampling rate of the developed sensor is 5 Hz according to the Shannon-Nyquist Sampling Theorem [45] which is quite low in terms of the capability of number of finger movements can be done in one second. The gesture movements may be captured in effective and detailed way with the higher sampling rates.

CHAPTER 5

CONCLUSION

5.1 Summary and Conclusion

Studying how touch perception works in humans and animals is especially important in order to develop artificial touch perception systems for social robots. In this thesis, improving the classification of touch gestures and generalizability of the proposed methods have been attempted. The conducted experiments and analyses provided us to understand the problem of touch gesture classification.

The ultimate goal of this thesis was to create unified model to be successful in classification of touch gestures. In that regard, we classified two publicly available dataset with four deep learning model. Generalization error of these two datasets are minimized with ensemble classifier, data augmentation, switching ADAM optimization algorithm to SGD. The proposed models are also tested with our demo dataset with developed sensor setting. We performed same models as done in the two publicly available dataset to minimize generalization error.

We observed that a single model is not enough to learn all necessary features of touch gestures. Ensemble classifier by combining all of the proposed architectures achieved more accurate results. Attempts to increase capacity of deep learning models was quite effective. It is clearly observed that data augmentation and altering the optimization algorithm provide well generalizability. It can also be concluded that most of the proposed model are beneficial compared to the literature.

The number of gesture sequence in the datasets did not enough to improve generalization performance of the proposed model. The common approach for such improvement is collecting more data as much as possible. The results of research in [40] [62] delivers that the performance on deep learning classifier increases logarithmically based on volume of training data size and plateaus after certain number of training data is

seen. Our results show that classification performance of the proposed model increased by augmenting the training set of the datasets.

The tactile sensors and the sensors that mentioned in this thesis collects lower level indicators of touch gestures, not higher level of indicators due to the structure of sensor array. No possible way to capture 3-dimensional shape of hand and touch gesture through spatial and temporal axis on those sensor array. The pressure values of sensor can be counted as the third dimension but it still not enough to represent touch gestures

Deep learning algorithms are quite robust to errors in the training set, less robust to systematic errors. It is not easy to determine mislabeled touch gesture in the three dataset as commonly done in the image dataset. Despite all systematic errors in the datasets, the results are promising for three dataset compared to existing literature in term of accuracies.

5.2 Future Studies

The results have shown that the data augmentation methods improved the generalization model. Additional data augmentation method, scaling may be beneficial, since all the subjects in the dataset have different size of hands. However, data augmentation methods are not an efficient method to overcome generalization problem. The proposed models are able learn features of touch gestures that have variety of orientation. More advance methods can be applied in order to learn transformation of gestures during movement. Capsule Networks may resolve such transformation problem efficiently [54] [9], it can be used as future model instead of using data augmentation to capture equivariant features of the touch gestures.

Another powerful method to capture invariant features of touch gestures could be Dynamic Time Warping (DTW) [55]. DTW is a simpler approach and able to handle one dimensional time series determining discrepancy between them. We believe that combining geometric invariant properties with the neural network classifier may be a possible solution to classification of touch gesture problem, as in [10] [8].

The accuracy of our demo dataset may be increased by collecting more data and adding new blocks to proposed architectures. The one of aim of this thesis is to propose

unified deep neural network models for gesture classification. However, our demo dataset have more sensor channels in the experimental setting than in the CoST and HAART sensor settings. So, new blocks to models may be added since at the input of neural network the touch gestures spatial and temporal features could not be preserved due to flattening.

One of the aims of machine learning models to approach the level of classification accuracy as human did. In order to compare the performance of humans and the proposed models, an experiment can be designed to recognize touch gestures that are performed on individual skin. The individual whose skin is touched can try to predict the touch gestures without looking them. In this way, we can test the proposed models in terms of human performance. So that, we will have a metric to compare our models with real-touch feeling.

The error margin or uncertainty of model have not been investigated in this thesis. Creating neural network using probabilistic methods can be quite effective to understand the limitation of proposed models. Moreover, the recent studies have attempts to give a proof the underlying theory of neural networks [51] but they are still in the category of observational studies. In my opinion, there is a long way to give a proof of the theory neural networks.

Further research to fill in gaps in our understanding of classification of touch gestures might increase the capacity of models, especially concentration on learning of invariant and equivariant features of touch gestures. Additionally, as emphasized in [47], there are always only a finite sample from theoretically infinite population of individuals to record their touch gestures. In that regard, the literature has still need to understand underlying structure of individual gestures. Even if the datasets can classified to certain accuracies, gestures in out of distribution gestures may not be classified correctly.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Saad Albawi, Oguz Bayat, Saad Al-Azawi, and Osman N Ucan. Social touch gesture recognition using convolutional neural network. *Computational Intelligence and Neuroscience*, 2018, 2018.

[3] Ethem Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2nd edition, 2010.

[4] Tugce Balli Altuglu and Kerem Altun. Recognizing touch gestures for social human-robot interaction. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, page 407–413, New York, NY, USA, 2015. Association for Computing Machinery.

[5] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478. Springer, 2012.

[6] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

[7] Rafael Alejandro Calvo, Sidney K. D'Mello, J. Gratch, and Arvid Kappas. The oxford handbook of affective computing. 2014.

[8] Chien-Yi Chang, De-An Huang, Yanan Sui, Li Fei-Fei, and Juan Carlos Niebles. D3tw: Discriminative differentiable dynamic time warping for weakly supervised

action alignment and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3546–3555, 2019.

[9] Taco S. Cohen and Max Welling. Group equivariant convolutional networks. *CoRR*, abs/1602.07576, 2016.

[10] Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. In *International conference on machine learning*, pages 894–903. PMLR, 2017.

[11] Ravinder S Dahiya, Philipp Mittendorfer, Maurizio Valle, Gordon Cheng, and Vladimir J Lumelsky. Directions toward effective utilization of tactile skin: A review. *IEEE Sensors Journal*, 13(11):4121–4138, 2013.

[12] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12(null):2121–2159, jul 2011.

[13] Wail El Bani and Mohamed Chetouani. *Touch Recognition with Attentive End-to-End Model*, page 694–698. Association for Computing Machinery, New York, NY, USA, 2020.

[14] Anna Flagg and Karon MacLean. Affective touch gesture recognition for a furry zoomorphic machine. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*, TEI '13, page 25–32, New York, NY, USA, 2013. Association for Computing Machinery.

[15] Terrence Fong, Illah Reza Nourbakhsh, and Kerstin Dautenhahn. A survey of socially interactive robots : Concepts , design , and applications. 1992.

[16] Masahiro Fujita. Aibo: Toward the era of digital creatures. *The International Journal of Robotics Research*, 20(10):781–794, 2001.

[17] Yona Falinie A. Gaus, Temitayo Olugbade, Asim Jan, Rui Qin, Jingxin Liu, Fan Zhang, Hongying Meng, and Nadia Bianchi-Berthouze. Social touch gesture recognition using random forest and boosting on distinct feature sets. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, page 399–406, New York, NY, USA, 2015. Association for Computing Machinery.

[18] Pedro Silva Girão, Pedro Miguel Pinto Ramos, Octavian Postolache, and José Miguel Dias Pereira. Tactile sensors for robotic applications. *Measurement*, 46(3):1257–1271, 2013.

[19] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.

[20] Carina Soledad González-González, Verónica Violant-Holz, and Rosa Maria Gil-Iranzo. Social robots in hospitals: A systematic review. *Applied Sciences*, 11(13), 2021.

[21] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[22] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[23] R. Heslin, T. D. Nguyen, and M. L. Nguyen. Meaning of touch: The case of touch from a stranger or same sex person. *Journal of Nonverbal Behavior*, 1983.

[24] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[25] Dana Hughes, Nicholas Farrow, Halley Profita, and Nikolaus Correll. Detecting and identifying tactile gestures using deep autoencoders, geometric moments and gesture level features. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, page 415–422, New York, NY, USA, 2015. Association for Computing Machinery.

[26] Dana Hughes, Alon Krauthammer, and Nikolaus Correll. Recognizing social touch gestures using recurrent and convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2315–2321. IEEE, 2017.

[27] Gijs Huisman, Aduen Darriba Frederiks, Betsy Van Dijk, Dirk Hevlen, and Ben Kröse. The tasst: Tactile sleeve for social touch. In *2013 World Haptics Conference (WHC)*, pages 211–216. IEEE, 2013.

[28] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[29] Merel M. Jung, Laura Cang, Mannes Poel, and Karon E. MacLean. Touch challenge '15: Recognizing social touch gestures. *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, 2015.

[30] Merel M Jung, Mannes Poel, Ronald Poppe, and Dirk KJ Heylen. Automatic recognition of touch gestures in the corpus of social touch. *Journal on multimodal user interfaces*, 11(1):81–96, 2017.

[31] Merel M. Jung, Ronald Poppe, Mannes Poel, and Dirk K.J. Heylen. Touching the void – introducing cost: Corpus of social touch. In *Proceedings of the 16th International Conference on Multimodal Interaction*, ICMI '14, page 120–127, New York, NY, USA, 2014. Association for Computing Machinery.

[32] Merel M. Jung, Ronald Poppe, Mannes Poel, and Dirk K.J. Heylen. Touching the void – introducing cost: Corpus of social touch. In *Proceedings of the 16th International Conference on Multimodal Interaction*, ICMI '14, page 120–127, New York, NY, USA, 2014. Association for Computing Machinery.

[33] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

[34] Nitish Shirish Keskar and Richard Socher. Improving generalization performance by switching from adam to sgd. *ArXiv*, abs/1712.07628, 2017.

[35] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[36] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[37] Mike Lambeta, Huazhe Xu, Jingwei Xu, Po-Wei Chou, Shaoxiong Wang, Trevor Darrell, and Roberto Calandra. PyTouch: A machine learning library for touch processing. *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[38] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2, 1989.

[39] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[40] Suhua Lei, Huan Zhang, Ke Wang, and Zhendong Su. How training data affect the accuracy and robustness of neural networks for image classification. 2018.

[41] Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. *CoRR*, abs/1411.5908, 2014.

[42] Yun-Kai Li, Qing-Hao Meng, and Hong-Wei Zhang. Touch gesture recognition using spatiotemporal fusion features. *IEEE Sensors Journal*, 22(1):428–437, 2022.

[43] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[44] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.

[45] Alan V. Oppenheim, Alan S. Willsky, and S. Hamid Nawab. *Signals and Systems (2nd Ed.)*. Prentice-Hall, Inc., USA, 1996.

[46] Athanasios Papoulis and S Unnikrishna Pillai. *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education, 2002.

[47] Judea Pearl and Dana Mackenzie. *The Book of Why*. Basic Books, New York, 2018.

[48] Tony Prescott, Sheffield Robotics, Uni Sheffield, Ben Mitchinson, and Sebastian Conran. Miro: An animal-like companion robot with a biomimetic brain-based control system. 04 2017.

[49] Sara Price, Nadia Bianchi-Berthouze, Carey Jewitt, Nikoleta Yiannoutsou, Katerina Fotopoulou, Svetlana Dajic, Juspreet Virdee, Yixin Zhao, Douglas Atkinson, and Frederik Brudy. The making of meaning through dyadic haptic affective touch. *ACM Trans. Comput.-Hum. Interact.*, 29(3), jan 2022.

[50] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[51] Daniel A Roberts, Sho Yaida, and Boris Hanin. *The principles of deep learning theory*. Cambridge University Press, 2022.

[52] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[53] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. , 323(6088):533–536, October 1986.

[54] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *Advances in neural information processing systems*, 30, 2017.

[55] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49, 1978.

[56] Jelle Saldien, Kristof Goris, Selma Yilmazyildiz, Werner Verhelst, and Dirk Lefeber. On the design of the huggable robot probo. 2008.

[57] Thomas B Sheridan. A review of recent research in social robotics. *Current Opinion in Psychology*, 36:7–12, 2020. Cyberpsychology.

[58] T. Shibata, K. Inoue, and R. Irie. Emotional robot for intelligent system-artificial emotional creature project. In *Proceedings 5th IEEE International Workshop on Robot and Human Communication. RO-MAN'96 TSUKUBA*, pages 466–471, 1996.

[59] David Silvera-Tawil, David Rye, and Mari Velonaki. Artificial skin and tactile sensing for socially interactive robots. *Robot. Auton. Syst.*, 63(P3):230–243, jan 2015.

[60] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[61] Walter Dan Stiehl, Jeff Lieberman, Cynthia Breazeal, Louis Basel, Levi Lalla, and Michael Wolf. The design of the huggable: A therapeutic robotic companion for relational, affective touch. In *AAAI Fall Symposium: Caring Machines*, pages 91–98, 2005.

[62] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. *CoRR*, abs/1707.02968, 2017.

[63] Viet-Cuong Ta, Wafa Johal, Maxime Portaz, Eric Castelli, and Dominique Vaufreydaz. The grenoble system for the social touch challenge at icmi 2015. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, page 391–398, New York, NY, USA, 2015. Association for Computing Machinery.

[64] Tijmen Tieleman, Geoffrey Hinton, et al. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[65] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christopher Bregler. Efficient object localization using convolutional networks, 2014.

[66] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. C3D: generic features for video analysis. *CoRR*, abs/1412.0767, 2014.

[67] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[68] Sharan Vaswani, Aaron Mishkin, Issam Laradji, Mark Schmidt, Gauthier Gidel, and Simon Lacoste-Julien. Painless stochastic gradient: Interpolation, line-search, and convergence rates, 2019.

[69] Alessandro Vinciarelli. *Introduction: Social Signal Processing*, page 1–8. Cambridge University Press, 2017.

[70] Edmund Taylor Whittaker and George Robinson. The calculus of observations : an introduction to numerical analysis. 1967.

[71] Ashia C. Wilson, Rebecca Roelofs, Mitchell Stern, Nathan Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning, 2017.

[72] Steve Yohanan and Karon E. MacLean. The role of affective touch in human-robot interaction: Human intent and expectations in touching the haptic creature. *International Journal of Social Robotics*, 4:163–180, 2012.

[73] Naoto Yoshida, Shuto Yonemura, Masahiro Emoto, Kanji Kawai, Naoki Numaguchi, Hiroki Nakazato, Shunsuke Otsubo, Megumi Takada, and Kaname Hayashi. Production of character animation in a home robot: A case study of lovot. *International Journal of Social Robotics*, 14, 01 2022.

[74] Liang Zhang, Guangming Zhu, Peiyi Shen, Juan Song, Syed Afaq Shah, and Mohammed Bennamoun. Learning spatiotemporal features using 3dcnn and convolutional lstm for gesture recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[75] Nan Zhou and Jun Du. Recognition of social touch gestures using 3d convolutional neural networks. In *Chinese Conference on Pattern Recognition*, pages 164–173. Springer, 2016.

[76] Pan Zhou, Jiashi Feng, Chao Ma, Caiming Xiong, Steven Hoi, and Weinan E. Towards theoretically understanding why sgd generalizes better than adam in deep learning, 2020.

APPENDIX A

Table A.1: Touch gesture definitions in experiment

Gesture label	Gesture definition
Kavramak	Elinizin tamamını kullanarak nesneyi kavrayıp bırakınız. Hareketi tekrarlı bir şekilde sürdürünüz.
Parmak ile vurmak	Tek parmağınız ile nesneye birkaç defa vurunuz. Hareketi tekrarlı bir şekilde sürdürünüz.
El ile vurmak	Elinizin tamamını kullanarak nesneye vurunuz. Hareketi tekrarlı bir şekilde sürdürünüz.
Sıvazlamak	Elinizi obje üzerinde ileriye ve geriye hareket ettiriniz. Hareketi tekrarlı bir şekilde sürdürünüz.
Okşamak	Elinizi obje üzerinde, objeye bastırarak dairesel bir şekide hareket ettiriniz. Hareketi tekrarlı bir şekilde sürdürünüz.
Tırmalamak	Bütün parmaklarınızı kullanarak, tırnaklarınız ile fazla kuvvet uygulamadan objeyi tırmalayınız. Hareketi tekrarlı bir şekilde sürdürünüz.