RECOGNITION OF NON-MANUAL SIGNS IN SIGN LANGUAGE

by

Müjde Aktaş

B.S., Computer Engineering, Bilgi University, 2015

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2019

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Prof. Lale Akarun for the continuous support, kindness and immense knowledge that she has provided me. I would not be able to complete my graduate study if she did not believe in me in the first place, and gave the unique chance to do research with her.

My sincere thanks also go to Dr. Berk Gökberk, who has shown the patience to answer each and every question that I come up with during our collaborations. His precise and clear explanations contributed significantly to my progress. I would also like to thank Dr. Fatma Başak Aydemir and Dr. Berk Gökberk for accepting to participate in my thesis jury and for their insightful comments and valuable ideas.

I would like to thank my colleagues in Perceptual Intelligence Laboratory, Ahmet Alp Kındıroğlu and Oğulcan Özdemir for taking the time to share their knowledge and experience with me generously. I appreciate the support that my friends and colleagues Hüseyin Temiz and Sena Sanıoğlu have provided, both technically and spiritually.

I am grateful to Dr. Elena Battini Sönmez, who always motivates me to not to be afraid to set high goals and go after them.

I would like to thank my family and my friends for supporting me spiritually throughout the writing of this thesis and my life. The greatest gratitude goes to my grandmother Nuran Gümüştargaç, who has raised me and prepared me for ups and downs of life. I would like to thank my dear friend Elifsu Karakaya for providing her support in every possible way.

Last but not least, I would like to thank my boyfriend Bayram Cevdet Akdeniz for always being there for me.

ABSTRACT

RECOGNITION OF NON-MANUAL SIGNS IN SIGN LANGUAGE

Recognition of non-manual components in sign language has been a neglected topic, partly due to the absence of annotated non-manual sign datasets. We have collected a dataset of videos with non-manual signs, displaying facial expressions and head movements and prepared frame-level annotations. In this thesis, we present the Turkish Sign Language (TSL) non-manual signs dataset and provide a baseline system for non-manual sign recognition. A deep learning based recognition system is proposed, in which the pre-trained ResNet Convolutional Neural Network (CNN) is employed to recognize the question, negation side to side and negation up-down, affirmation and pain movements and expressions.

483 TSL videos performed by five subjects, who are native TSL signers were temporally annotated. We employ a leave-one-subject-out approach for performance evaluation on the test videos. We have obtained annotation-level accuracy values of 55.77 %, 14.63 %, 72.83 %, 10 % and 11.67 % for question, negation-side, negation-up-down, pain and affirmation classes respectively in the BosphorusSign-HospiSign non-manual sign datasets.

Question, negation-side, negation-up-down and affirmation movements and expressions in 87 clips from the TSL translation video of a Turkish movie are temporally annotated for cross-database experiments. The models that are fine-tuned on BosphorusSign-HospiSign set are tested with the clip frames. The best performing model classifies 66.67 % of question annotations and 42.31 % of negation-up-down annotations correctly, while the remaining class labels could not be predicted.

ÖZET

İŞARET DİLİNDE YÜZ İFADELERİ VE KAFA HAREKETLERİNİN TANINMASI

İşaret dilinde yüz ifadeleri ve kafa hareketlerinin tanınması konusu ihmal edilmektedir. Bu ihmalin nedenlerinden biri olarak etiketlenmiş veri seti eksikliği gösterilebilir. Bu çalışmada, yüz ifadeleri ve kafa hareketlerinin yer aldığı, manuel olmayan işaretleri içeren bir Türk İşaret Dili (TİD) veriseti toplanıp, video kareleri seviyesinde işaretleme yapılmıştır. Bu tezde Türk İşaret Dili kafa hareketleri ve yüz ifadeleri veri seti sunulmakta ve manuel olmayan işaretler için bir temel tanıma sistemi önerilmektedir. Derin öğrenmeye dayalı tanıma sisteminde, önceden eğitilmiş ResNet konvolüsyonel sinir ağı kullanılarak soru, olumsuzluk, tasdik etme ve acı hareket ve ifadeleri tanınmaya çalışılmıştır.

Ana dili Türk İşaret Dili olan beş öznenin işaretleri yaptığı 483 video zamansal olarak işaretlenmiştir. Deney testleri, bir özneyi dışarda bırakma tekniği kullanılarak yapılmıştır. Doğru sınıflandırılan işaretlemelere göre başarım, soru, olumsuz-sağ-sol, olumsuz-yukarı-aşağı, acı ve tasdik sınıfları için sırasıyla % 55.77, % 14.63, % 72.83, % 10 ve % 11.67 olarak ölçülmüştür.

Sırasıyla farklı özneleri dışarda bırakarak eğitilen beş farklı model ve yeni bir verisetinden alınan işaret dili videoları ile çapraz veriseti deneyleri yapılmıştır. Etiketlenen 87 kısa klipten, acı sınıfı dışındaki dört sınıfa ait işaretlemeler elde edilmiştir. En iyi performans gösteren model soru işaretlemelerinin % 66.67'sini ve olumsuz-yukarı-aşağı işaretlemelerinin % 42.31'ini doğru sınıflandırmakta, geri kalan sınıflara ait tahmin yapamamaktadır.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF TABLES . . . x
LIST OF SYMBOLS . . . xi
LIST OF ACRONYMS/ABBREVIATIONS . . . xiii
1. INTRODUCTION . . . 1
2. LITERATURE SURVEY . . . 4
2.1. Sign Language Recognition . . . 4
2.2. Data Acquisition Technologies . . . 6
2.3. Sign language Recognition Using Non-Manual Features . . . 7
2.4. Facial Features and Expressions in Sign Language . . . 7
2.5. Sign Language Recognition In the Wild . . . 10
3. METHODS . . . 12
3.1. Ground-truth Annotation Correcting . . . 12
3.2. Data Preprocessing Steps . . . 13
3.2.1. Keypoint Extraction with OpenPose . . . 13
3.2.2. Face Cropping . . . 14
3.3. Training and Validation . . . 16
3.3.1. Transfer Learning with ResNet . . . 16
3.3.2. ResNet . . . 17
3.3.2.1. Intuition of ResNet Architecture . . . 18
3.3.2.2. Finetuning the ResNet . . . 19
3.3.3. Model . . . 19
3.3.4. Hyperparameters . . . 21
3.3.5. Stochastic Gradient Descent Optimizer . . . 21
3.3.6. Weight Regularization . . . 22
3.3.7. Data Augmentation . . . 23

3.3.8. Validation and Testing 23
3.3.9. Prediction Analysis 23
3.3.9.1. Temporal Filtering 23
4. TURKISH SIGN LANGUAGE CORPUS FOR FACIAL EXPRESSION AND HEAD MOVEMENT RECOGNITION 26
4.1. Datasets 26
4.1.1. BosphorusSign 26
4.1.2. HospiSign 27
4.1.3. Bosphorus Facial Signs Dataset 27
4.1.4. Audio Description Association (SEBEDER) Film Archive Dataset 30
4.1.4.1. Obtaining Non-Manual Sign Clips from SEBEDER Film Translation Videos 31
4.2. Video Annotation 33
4.2.1. ELAN Multimedia Annotation Tool 33
4.2.2. Selecting Sign Videos Semantically 34
4.2.3. Signer Related Diversity 35
4.2.4. Transition 36
4.2.5. Parsing Annotations 36
5. EXPERIMENTS AND RESULTS 38
5.1. Experimental Setup 38
5.1.1. Data Augmentation 38
5.1.2. Validation and Testing 39
5.1.3. Evaluation Metrics 39
5.2. Results 41
5.2.1. Spotting the Question, Negation, Pain, and Affirmation in Bosphorus Facial Signs Dataset Videos 42
5.2.2. Cross-database test with the SEBEDER dataset videos 54
6. CONCLUSION 61
REFERENCES 64

LIST OF FIGURES

Figure 3.1 Pipeline of the recognition system. 12

Figure 3.2 Video frames of User 3 performing “How can I help you ?”. Frames are sampled with 1/15 frame rate. 13

Figure 3.3 OpenPose face keypoints [1]. 14

Figure 3.4 Two face cropping approaches, tight bounding box (left) and bounding box calculated as a function of IOD (right). 15

Figure 3.5 A building block of ResNet [2]. 18

Figure 3.6 ResNet18 architecture. 20

Figure 3.7 The effect of median filtering with kernel size k = 9 on selected consecutive frames of User 2. c_0, c_2, c_3, and c_5 represent other, negation-side, negation-up-down and affirmation classes, respectively. 24

Figure 4.1 Representative frames of each class. 28

Figure 4.2 SEBEDER sign language translation clip frame. 31

Figure 4.3 ELAN linguistic annotation software interface 33

Figure 4.4 Video frame samples of User 3 (top) and User 5 (bottom) performing the sign “Not available”. 35

Figure 5.1 Effect of data transforms when applied seperately on selected frames of User 1 (top) and User 3(bottom). 39

Figure 5.2 IoU score calculation. 41

Figure 5.3 Selected key frames of each class label. 42

Figure 5.4 Histogram of annotation tuples with ground truth positive class frame counts. 44

Figure 5.5 Overall confusion matrix and confusion matrices of each test user. 45

Figure 5.6 Colored comparison of ground truth vs. precited frame labels with class prediction probabilities. 48

Figure 5.7 Prediction and ground-truth labels given with prediction probabilities of each class. 49

Figure 5.8 Prediction and ground-truth labels given with prediction probabilities of each class. 50

Figure 5.9 Misclasified frames of the video represented in Figure 5.8a. . . . 51

Figure 5.10 Negation-up-down frames that are misclassified as question frames, from the video represented in Figure 5.8b. 51

Figure 5.11 Selected test videos without positive ground-truth (left) and without positive prediction (right). 51

Figure 5.12 Accuracy of ground truth annotations versus different t-IoU values. 53

Figure 5.13 Selected key frames of each class label in SEBEDER. 54

Figure 5.14 Confusion matrices of each test fold in SEBEDER experiment. . 56

Figure 5.15 Signer performs 'This does not exist, that does not exist', labeled as negation-up-down. 58

Figure 5.16 Prediction probability plots of selected videos from SEBEDER. 58

Figure 5.17 Accuracy of ground truth annotations versus different t-IoU values. 59

LIST OF TABLES

Table 4.1 Subjects in the BosphorusSign and HospiSign datasets. 28

Table 4.2 Average video durations (in sec.) per sign phrase and per user. . 29

Table 4.3 Class distribution of videos from HospiSign (HS) and BosphorusSign (BS) subsets. 30

Table 4.4 Frame count of class labels after parsing annotations. 32

Table 5.1 Training and test splits for each user fold. 43

Table 5.2 Frame-level performance measurement for non-manual sign recognition. Average values are calculated excluding the null class label "other". Balanced accuracy is denoted with *. 46

Table 5.3 Count of ground truth annotations per class label in each test user fold. 52

Table 5.4 Training and test splits for each partition fold in SEBEDER experiments. 55

Table 5.5 Frame level average performance measurement for non-manual sign recognition in SEBEDER. Balanced accuracy is denoted with *. 57

Table 5.6 Total number of ground truth annotation tuples per class label in SEBEDER video clips. 59

LIST OF SYMBOLS

a	nth ground truth annotation tuple in the video
A_v	Set of ground truth annotation tuples in video v
$\hat{a_n}$	nth predicted annotation tuple in the video
$\hat{A}_v$	Set of predicted annotation tuples in video v
B_i	Bounding box of the ith frame in the video
C_h	Center in the horizontal axis between the eye pupils in face image
exp	Exponential function
$\mathcal{F}$	Residual function
fcn	Fully connected layer with n outputs
f_s	Start frame index of the annotation
f_e	End frame index of the annotation
f_{min_c}	Minimum frame count for class label c
g	Gradient of the loss function in Stochastic Gradient Descent step
g_n	Time gap between the nth with the $n+1$th ground truth annotations in milliseconds
G_{v_i}	Set of time gap between each consecutive ground truth annotation tuple in ith video v
IOD_i	Inter-occular distance between the eyes in the ith face image in the video
$IoU(a, \hat{a})$	Intersection over union between annotation tuples a and $\hat{a}$
l_n	Class label of the nth annotation
lr	Learning rate
p	Parameters of the neural network
$p(class)$	Class probability
s	Constant scale of IOD_i for bounding box calculation
t_g	Threshold for ground-truth annotation smoothing
t-IoU	Threshold for Intersection Over Union score
t_{ns}	Starting time of the nth annotation in milliseconds

t_{ne}	End time of the nth annotation in milliseconds
t_{pc}	Frame count threshold for class label c
v	Velocity in Stochastic Gradient Descent step function
W	Weight of the neural network layer
ρ	Momentum in Stochastic Gradient Descent step function
σ	Rectified Linear Unit activation function
$\sum$	Summation

LIST OF ACRONYMS/ABBREVIATIONS

3D	Three Dimensional
API	Application Programming Interface
ASLR	Automatic Sign Language Recognition
ASL	American Sign Language
AAM	Active Appearance Model
CTC	Connectionist Temporal Classification
CRF	Conditional Random Field
FACS	Facial Action Coding System
FPS	Frames Per Second
FER	Facial Expression Recognition
HCI	Human Computer Interaction
HMM	Hidden Markov Model
IOD	Inter Occular Distance
iDT	Improved dense trajectories
JSON	JavaScript Object Notation
LBP	Local Binary Pattern
MEI	Motion Energy Image
MHI	Motion History Image
R-CNN	Regions with CNN
RGB	Red Green Blue
SLR	Sign Language Recognition
TSL	Turkish Sign Language

1. INTRODUCTION

Sign languages are the means of communication of the deaf and hearing-impaired society. Sign language uses hand and body gestures, hand shapes as well as facial expressions to convey meaning. Each culture has its own sign language, which is independent of the languages spoken in that region.

According to the World Health Organization, over 5% of the world's population has a disabling hearing loss and it is estimated that the deaf society population will double up by 2050 [3]. There have been several improvements in the past years to fulfill the needs of deaf and hearing-impaired society. One improvement is the establishment of laws requiring translation services, such as the provision of real-time translation from spoken language to sign language during the television broadcast. Education that is specialized for the deaf community has also become widespread. Most of the efforts take place in developed countries while developing countries are following behind the same trend.

Sign languages around the world have a considerably big vocabulary. Every nation has its sign language as opposed to the common misconception. Deaf and hearing-impaired communities encounter major drawbacks in daily life activities, education and health fields. Employing sign language translators in institutions and organizations for public service would be a costly investment to sustain. A feasible and affordable solution to this problem would be to automatize the recognition and translation procedure for sign languages.

The development of automatic sign language translation and recognition systems (ASLR) would improve the integration of the hearing impaired into society. Lots of research has been done in the field of sign language recognition. However, when it comes to real-time translation of signs to spoken or written language, or vice-versa, we cannot easily proceed from the research phase to the production phase.

The vast majority of ASLR research has been focusing on manual features; these are hand gestures, orientation and shape of hands and fingerspelling. However, a significant amount of information lies also in non-manual features; which are head and torso movements, facial expressions and mouth movements. Facial expressions and head movements play an essential role in grammatical markers, such as question, negation, topic, assertion, doubt, and condition. Grammatical markers are specific to each sign language. Non-manuals are specifically distinctive for question and negation markers in Turkish Sign Language (TSL).

In this thesis, we develop a frame based recognition system for head movements and facial expressions in Turkish Sign Language. Turkish Sign Language is indigenous to Turkey, as opposed to American Sign Language and British Sign Language, which are used as first sign language in several other countries as well. The aim of this thesis is to contribute to TSL recognition using deep learning technology, which has proven successful in many other research fields. Research in non-manual sign recognition exists but is limited. In the context of sign language recognition, the lack of datasets with labeled non-manuals is a challenge. To the best of our knowledge, a TSL dataset with labeled grammatical facial expressions and head movements does not exist at the time of writing this thesis.

In this thesis, we create a non-manual sign annotated TSL dataset. First, sign videos from BosphorusSign TSL dataset [4] and the HospiSign Project [5] with a specific corpus of the question, negation, affirmation and pain phrases are selected.

We label facial expressions and head movements in the videos temporally. A terminal application is developed to automatize the annotation procedure and process the sign videos in batches. We employ the OpenPose keypoint detection system [6] to extract facial landmarks of the signers. Face images are extracted from video frames, using the landmark coordinates.

We first challenge the classification of interrogatory expressions in sign videos. Conventional feature extraction methodologies were not considered at any step of this task. Instead, raw face images have been fed into the pre-trained ResNet18 convolutional neural network as input. We use the learned weights of the ImageNet training set (from ILSVRC 2015 classification challenge) to initialize the model. We gradually enlarge the training and test set after each batch of video annotation.

Using a 2D CNN to classify the video frames separately, we lose the temporal relation between the frames. In order to preserve the temporal integrity of videos, we use Intersection over Union (IOU) based evaluation technique in addition to frame-based precision, recall, and accuracy. Performance evaluation of proposed systems is based on leave one out method. In each experimental setup, we leave all samples of one user out, train the network with the rest of the dataset and test the network on leaved out user samples. We repeat this procedure for all users and report the results separately. This guarantees the reliability of the evaluation procedure, as there are several repetitions for each video in the dataset. We post-process the ResNet prediction results to prepare for IoU based performance evaluation.

The thesis is organized as follows: Chapter 2 summarizes the literature in this field. Chapter 3 briefly describes the methods and techniques that are employed for classification of the non-manual signs in the prepared TSL dataset. Chapter 4 introduces the annotated datasets and explains the data annotation and pre-processing phases in detail. Experimental results are reported and discussed in Chapter 5. Finally, the conclusion and future work are presented in Chapter 6.

2. LITERATURE SURVEY

2.1. Sign Language Recognition

Sign languages consist of structured hand gestures; combined with facial expressions, head movements and upper body movements. Hand gestures have an essential role in sign languages. Fingerspelling, signs of isolated words and continuous signs are carried out by the hands.

Hands of the subject are in motion when performing a gesture. Motion Energy Image (MEI) and Motion History Image (MHI) can be used as temporal templates [7] to detect the area of motion, thus extract the hand region in the image. MEI holds binary information; whether a motion appears in an image sequence or not. MHI holds the scalar intensity information in terms of recency of motion. Akyol and Alvarado [8] make use of the latter to find out where manual signs take place in image sequences of hand gestures. Algorithms like Continuously Adaptive Meanshift (CAMshift) -which are based on mean shift technique- and particle filtering are used for tracking hands throughout the image sequence, as reviewed in [9].

The increase in accessibility of powerful GPUs was followed by the increase in popularity of deep learning techniques in computer vision tasks. A breakthrough study was [10] in 2012; the authors have trained a deep convolutional neural network on the challenging ImageNet dataset to classify images from 1000 different categories and have significantly outperformed existing methods with a top-5 error rate of 17%.

CNNs are found to be successful not only for image recognition but also for video recognition tasks. In 2014, [11] introduced a baseline single frame CNN and three novel approaches for fusing the information within a time window.

They propose an early fusion model using convolutional layer filters of size $W \times H \times 3$ where T is the temporal length, a late fusion model in which two single frame CNNs are merged in the first fully connected layer, and a slow fusion model in which they apply both spatial and temporal convolutions. On 200,000 test videos of Sports 1M dataset, which the authors collected from YouTube videos of sports activities, they found that single frame baseline model performs well enough with 59.3% of videos having a correct prediction in top-1. The slow fusion model outperforms the single-frame model by a small margin, classifying 60.9% of videos correctly in top-1 prediction.

By the nature of the videos, unless recording a still scene -e.g. recording indoors of a closed market at night– acquired visual data change in time. An exception for this is fingerspelling recognition, where it is possible to capture each letter in particular frames of the video. However, when it comes to signs, all manual and non-manual cues should be captured continuously. Contributions in ASLR research enabled the transition from processing isolated signs to continuous sign videos. For continuous recognition, one should detect the boundaries of each sign in a sentence. Video understanding, therefore, requires a temporal modality. Designing the model to best represent the spatiotemporal feature of sign videos is a common research question in SLR literature.

Attacking the problem of motion modeling, [12] compares the performance of 3D CNN on multiple action benchmarks. The authors have found that for modeling temporal information, best-performing kernel temporal depth of convolution layers is three and uses $3 \times 3 \times 3$ convolution kernels. They combined the C3D pre-trained I380K dataset and fine-tuned on the Sports-1M dataset with a simple linear classifier and obtained 85.2% recognition accuracy following the standard three train/test splits of UCF101. However, [13] performed better with 88.2% on the same set of videos using convolution pooling on long clips. Integrating improved Dense Trajectories as an additional feature to RGB frames, the authors [12] increased the accuracy to 90.4%.

Another study challenged the temporal segmentation problem in continuous SLR, to skip the costly and error-prone procedure of temporal localization of each sign in a sign video [14]. A 2-stream 3D CNN is employed for video feature extraction and an LS-HAN is used for sign sentence generation from the video. First, a global 16-frame video clip and a local tightly cropped hand region are fed into separate streams. Global and local information is combined with late fusion at fully connected layers. They described each video with a sequence of 4096-dimensional feature vectors per clip. Separately, each word in a sign sentence is one-hot encoded. Two feature vectors are represented in the same latent space to preserve the video-sentence relationship. The authors collect the CSL dataset with 25,000 videos of 178 sentences. On 2,000 test videos of the CSL dataset, they obtained up to 82% accuracy.

In [15], authors have released an ASL dataset which contains sign videos of word sequences and sentences, labeling each word in videos temporally. They proposed a hybrid network model that uses the C3D network to extract spatiotemporal features and feeds these to RNNs to extract sequential information. Separate C3D networks, all pretrained on Sports 1M dataset [11] have been finetuned with RGB, depth and optical flow input channels for feature extraction. FC-RNN classifies several clips for each video and prediction results are fused for the final decision. The authors have tested C3DRNN with 27 words from their collected ASL dataset. The comparison with one of the state-of-the-art systems [12], on the same dataset has shown that C3DRNN outperforms C3D by ~10% margin, achieving 65.8% accuracy on the person independent scenario.

2.2. Data Acquisition Technologies

In SLR, data acquisition techniques can be grouped into sensor-based and vision-based approaches. Sensor-based approaches use items such as data gloves, markers and motion sensors. In vision-based approaches, data acquisition can be accomplished with a single RGB camera, stereo camera, as well as new sensors such as Microsoft Kinect [16] and Leap Motion Controller technologies [17].

Microsoft Kinect sensor can be considered as both sensor and vision-based technology, which enables the collection of RGB, skeletal and depth data simultaneously. Data acquired with monocular cameras suffer from lack of depth information, whereas 3D scanning with Kinect or Leap Motion provides accurate depth and position information.

2.3. Sign language Recognition Using Non-Manual Features

Head pose, facial expressions, head and mouth gestures are essential components of a sign language, other than the hands. Majority of the studies in ASLR concentrate on manual features [18], [14], [8]. For non-manual sign language recognition, the tendency is towards classifying facial features solely and then integrating the results to the main framework of recognition system [19], [20]. However, the role of non-manual cues should not be neglected. It is highly probable to encounter signs that are only distinguishable by facial cues or upper body movements of the signer. Authors highlight the necessity of disambiguating such signs and provide a solution based on mouth analysis [21].

2.4. Facial Features and Expressions in Sign Language

In the broad context of automatic analysis of facial expressions, some researchers use the convention of six basic facial expressions that are introduced by Darwin [22]. These expressions are anger, fear, happiness, disgust, surprise, and sadness. Facial Action Coding System (FACS) [23] introduces a standardized way of coding the basic expressions with atomic facial muscle actions. However, in the context of ASLR, we are interested in facial expressions, which co-occur with hand gestures and head movements. For this reason, facial expressions and head movements in sign language are often handled together with hand gestures in the literature.

LBP [24] is a popular descriptor technique that is being used for facial recognition since 2006 [25]. As introduced in [26], it is possible to combine spatial and temporal information using LBP based calculations called Volume LBP and LBP of Three Orthogonal Planes (LBP-TOP). The authors successfully applied both techniques to facial expression recognition problem.

In [27], the authors describe the non-manual cues of sign language as head pose, facial expression, and lip patterns. The head pose helps with the interpretation of the sign performed; whether there is an affirmation, a negation, a question or a conditional situation. Facial expressions are mostly related to grammar. Combined with the head pose, facial expressions can determine the sentence structure. The most significant components of facial expressions are lifting or frowning the eyebrows and changing the shape of the mouth. Lip patterns are another strong representative which help for solving the ambiguity between similar signs.

More techniques that are used in expression recognition can be listed. However, there are few studies in which facial expression recognition serve for sign language interpretation [28], [29]. Facial expression recognition in the context of ASLR remains an open challenge.

It is common practice to categorize facial features into appearance-based and landmark-based features. In [27], authors extract both features by computing interesting areas in the face graph with landmark points in addition to the Active Appearance Model (AAM). Authors of [30] use appearance-based features from one of their previous studies [20], for analyzing hand shape and head motions. They employ a sequential belief-based Hidden Markov Model (HMM) which consists of two stages; manual and non-manual HMMs in the first stage and non-manual HMMs in the second stage. The second stage is activated only for signs with ambiguous manual features.

In a landmark-based approach, authors of [19] extract non-manual features with AAM and do the classification with SVM. Liu et al. [29] focus on eyebrow gestures by extracting geometric and appearance features from the region of eyebrows in face images. Their feature-set contains temporal information as they employ CRF to recognize eyebrow and periodic head gestures.

In [27], facial features extracted from AAM have shown to improve signer independent recognition performance from 78.7% to 80.2% for 450 isolated signs. Meanwhile, the recognition rate of the same vocabulary in continuous videos was increased from 60.6 % to 65.1 %.

The study [31] suggests that non-manual cues represent activities in three tiers: Repetition to emphasize; eyebrows, eyes, and mouth tier to display facial expressions and head and eye gaze tiers to focus on specific action or objects while signing.

The importance of facial features is emphasized in [32] where Hidden Markov Models (HMMs) were used to encode eyebrow, eye (widen and squint) and head movements. It is stated that there are seven different types of grammatical markers in sentences, which are wh-question (WH), yes-no question (YN), rhetorical question, topic (TP), conditional clause, relative clause and negation (NEG). In this study, WH, YN, NEG and TP expressions are formulated as a composition of the eyebrow, eye, and head movements.

In [33] grammatical facial expressions are considered as one of the key aspects for recognition at the syntax level. Similar to the [32], this study formulates WH questions, conditional, negative or affirmative expressions using eyebrows, eyes, mouth and head. An example sentence for conditional clause would be “If it’s sunny, I go to the beach.”. In this study, experiments are conducted on frame level; the authors aim to spot the existence of a question, negation or any other grammatical markers in a video frame. They have encoded the face in video frames with several vector representations; which are either directly taken or derived from face coordinates in x, y and z-axis.

Temporal information is also taken into account with sliding windows approach. Windows of size between three and six were found to perform best in their experiments.

Another study demonstrates the significance of facial expressions to distinguish fundamental grammatical markers in LIBRAS sign language, by only using facial keypoints to train a custom feed-forward network [34]. In each video in the GFE dataset used, signers perform a single marker sentence. Without explaining the details of the frame selection process, the authors have located the attribute points on faces of signers, on a total of 27,965 keyframes from the videos and labeled the grammatical marker in frames with the help of a sign language expert. The feed-forward network is trained with face attribute points as inputs. For a binary classification model, overall mean accuracy for recognizing 9 different grammatical markers is 98.04%.

2.5. Sign Language Recognition In the Wild

In SLR literature, collection of sign language data is accomplished in laboratory environments, typically with high contrast background [9]. Ideally, automatic sign language recognition systems should be designed in an in-the-wild manner, considering real-life scenarios and use cases. A hearing-impaired person consulting the front office, or a deaf patient arriving at the hospital are very intuitive examples for such scenarios. ASLR systems that are robust to illumination, occlusion, pose changes and independent users are yet to be developed. This research area still needs contributions.

To the best of our knowledge, there is currently not any published research about sign language recognition in the wild. The study in [18] is the only one that adapts deep learning techniques to ASL fingerspelling videos in the wild. First, they have collected various ASL videos from YouTube, a website of an ASL organization that publishes educational videos and an ASL social media website. After several linguists annotated the start and end of each letter fingerspelling in videos using ELAN [35], 7,304 fingerspelling sequences are obtained.

In another study, the authors adopt Faster R-CNN [36] for hand region prediction. For fingerspelling prediction, they feed RGB hand image concatenated with the optical flow to the AlexNet pre-trained on ImageNet as [10], and a single layer LSTM recognizes the fingerspelling. They achieved 41.9% accuracy on 868 test sequences using the hand regions and CTC based LSTM.

3. METHODS

In this chapter, the setup of classification experiments is described. The details of training, validation and test phases are explained. Face images that are prepared as explained in Chapter 4 are organized in a leave-one-subject-out manner for training and testing. The pre-trained ResNet convolutional network is trained on the face images to recognize particular head movements and facial expressions. Finally, the prediction results of frames are post-processed for interpretation and enhancement. An overview of the system is given in Figure 3.1.

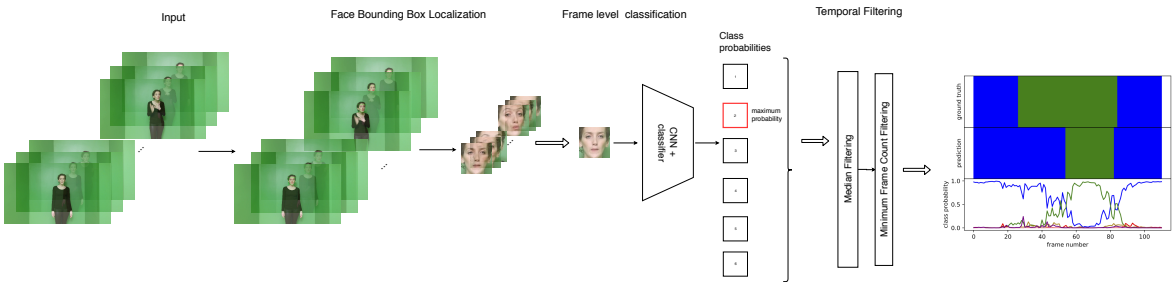


Figure 3.1: Pipeline of the recognition system.

3.1. Ground-truth Annotation Correcting

Sign videos in BosphorusSign dataset has the similar negative-positive-negative pattern in terms of movements and expressions, as can be seen in Figure 3.2. In this context, we are using:

- *Positive* notation to represent the frames with expression and/or movement of interest
- *Negative* notation to represent neutral/background frames

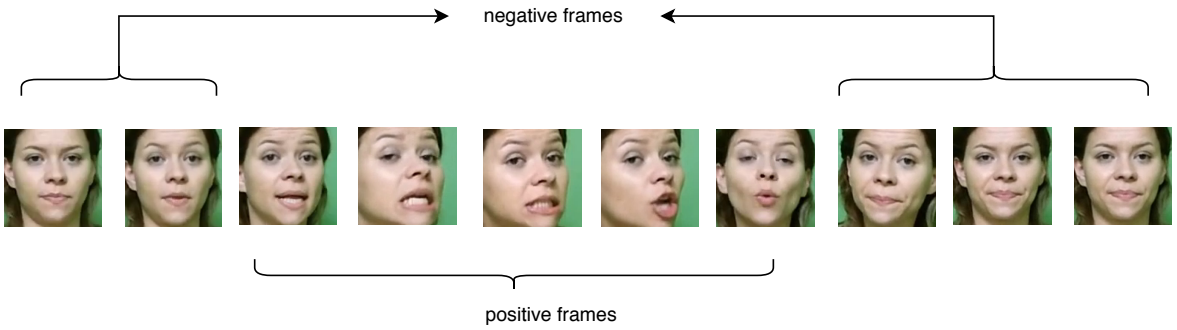


Figure 3.2: Video frames of User 3 performing "How can I help you ?". Frames are sampled with 1/15 frame rate.

Using this prior knowledge, small time gaps between consecutive positive annotations were filtered to avoid possible human error. In other words, we eliminate the negative ground truth labels if the number of consecutive frames with negative labels is less than the threshold t_g.

For this, we calculate the time gap between each consecutive annotation tuple in A_v. We obtain the set of $G_{v_i} = \{g_n = t_{n+1_s} - t_{n_e}\}_{n=1}^{N-1}$.

To avoid the small gaps, we define the threshold t_g milliseconds and we merge all consecutive annotation tuples with $g_n < t_g$ if $l_{n+1} = l_n$.

3.2. Data Preprocessing Steps

3.2.1. Keypoint Extraction with OpenPose

OpenPose is an open source multi-person keypoint extraction library, which estimates face, pose, hand and foot keypoints in real-time. This library is available with C++ and Python APIs. It also provides a demonstration tool for those who do not need to modify the default configuration. We have used the demonstration tool in order to extract face, hand and pose keypoints of signers in BosphorusSign TSL dataset.

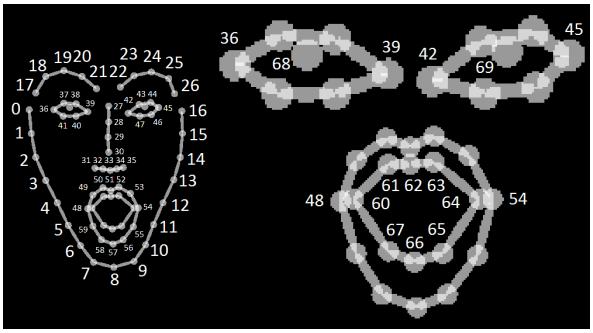


Figure 3.3: OpenPose face keypoints [1].

OpenPose outputs the keypoint estimation to JSON files. After obtaining the keypoint JSON files in a single run, files were parsed and serialized into a single Python pickle object for later and repetitive uses.

3.2.2. Face Cropping

For every frame in each video, the face bounding box is calculated using the corresponding OpenPose face keypoints. The bounding box is calculated using two different approaches.

(i) A tight bounding box is calculated using border key points of the face. This approach guarantees the smallest possible face region in each frame. The average size of face images is 115×113.

(ii) A square bounding box B, with size proportional to the interocular distance of the eyes and a scale s. The average size of face images is 135×135.

$$Width(B_i) = Height(B_i) = IOD_i \times s \tag{3.1}$$

where B_i is the bounding box of ith frame in the video, IOD is the interoccular distance ans s is a constant scalar. Examples are given in Figure 3.4.

For (ii), s is experimentally set to 2.7. B is located on the frame such that its center in horizontal axis C_h is the midpoint of eye pupils. Accordingly, in vertical axis, 30% of $Height(B_i)$ is above C_h and 70 % of $Height(B_i)$ is below C_h.

Although both approaches guarantee to include face region in the frames, we have used the latter, more loose box to crop the faces in each frame. The intuition behind this choice is that eyebrows and forehead play an essential role in facial expressions, therefore bring distinctive information.

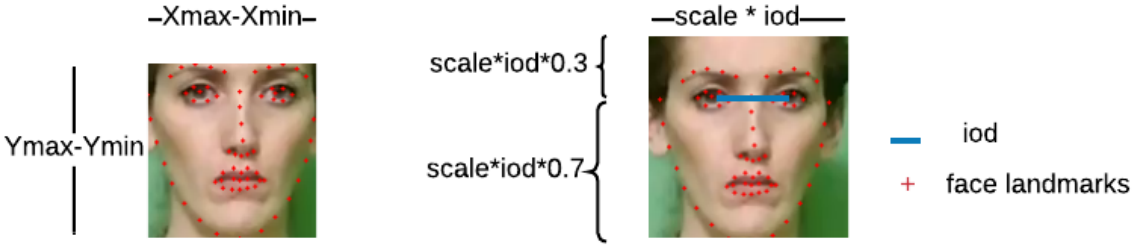


Figure 3.4: Two face cropping approaches, tight bounding box (left) and bounding box calculated as a function of IOD (right).

All video frames are extracted and saved as both raw images and face cropped images. Since OpenPose only *estimates* the facial landmarks, a cross-check is made to make sure each cropped face image indeed contains a face.

3.3. Training and Validation

3.3.1. Transfer Learning with ResNet

The training phase in neural networks has two directions: forward pass and backward pass.

(i) Forward pass is the process of calculating the output given the input data, weights and biases. Given the output and the target, the loss is calculated using a predefined metric.

(ii) Backward pass is the process of recursively applying the chain rule to calculate the gradients of the loss function with respect to network parameters, i.e. weights and biases. This process is called backpropagation, short for backward propagation of errors. The learning process in neural networks is made possible by the backpropagation algorithm.

Aim of the learning process is to minimize the error so that the predictions are as close as possible to actual outputs. This cycle repeats until some pre-determined condition (i.e. the number of epochs, loss threshold) is satisfied.

One of the biggest challenges of training deep neural networks is the requirement of large and labeled datasets. Sign language recognition is a specific research domain that still lacks such huge amounts of labeled data.

For problem scenarios like this, instead of training an entire convolutional neural network from scratch, researchers commonly exploit the technology that is called transfer learning. Transfer learning is the technology of taking the advantage of learned weights of a model that is previously trained on a large dataset.

Two approaches are applicable in transfer learning with CNNs:

- Finetuning CNN
- Using CNN as a feature extractor

In the former, instead of random initialization, weights of the network are initialized with learned weights of the pre-trained network. And weights of all layers are updated during training. In the latter, only the weights of the last fully connected layer are trained, and all remaining layers are freezed. Both approaches require modification of the last layer with respect to the number of classes.

We employ the pre-trained ResNet model and finetune it. The motivation behind this model choice is explained in Chapter 3.3.2.

In this part of the thesis, generic ResNet architecture is going to be briefly introduced. Model design details of the specific ResNet versions used in our experiments are given in the following sections. The employed transfer learning technique is explained. Then, the hyperparameters and optimization methodology is discussed. Finally, our specific experimental setup for training, validation, and testing phases are given.

3.3.2. ResNet

In very deep neural networks, weights and biases of early layers cannot be updated effectively due to very small gradient values. This problem is caused by activation functions e.g. the sigmoid function which by nature squeezes the derivative values to a small range. This situation is called "the vanishing gradient" problem. ResNet architecture is a design solution to the vanishing gradient problem and accuracy degradation.

3.3.2.1. Intuition of ResNet Architecture.

(i) Use of identity mappings, so that the network becomes deeper without increasing the computation cost.
(ii) Use shortcut connections, that is, skip one or more layers to escape from squeezing activations.

In [2], authors took advantage of both ideas and used identity mapping in shortcut connections.

In a building block of ResNet, Rectified Linear Unit (ReLU) activation function is used for nonlinearity. A number of convolutional layers, typically 2 or 3 are stacked. Residual learning is adopted for every few stacked layers.

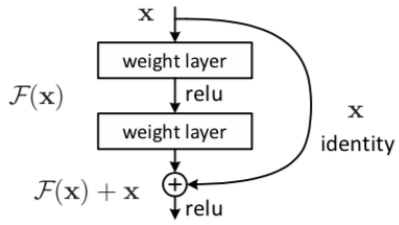


Figure 3.5: A building block of ResNet [2].

Building block in Figure 3.5 is formulated in [2] as follows:

$$y = \mathcal{F}(\mathbf{x}) + \mathbf{x} \tag{3.2}$$

$$\mathcal{F} = W_2\sigma\left(W_1\mathbf{x}\right) \tag{3.3}$$

Here, $\mathcal{F}$ is the residual function and $\mathbf{x}$ is the identity mapping. A "residual" is the amount to be added to prediction so that the prediction is equal to actual. When $\mathbf{x}$ is optimal, i.e. prediction is equal to the actual, weights get to zero, therefore $\mathcal{F}(\mathbf{x})$ get to zero. This causes a direct mapping of $\mathbf{x}$ to y. When $\mathbf{x}$ is not optimal, weights and biases of $\mathcal{F}(\mathbf{x})$ are learned to make it optimal.

3.3.2.2. Finetuning the ResNet. The fully connected layer denoted with fc1000 in default configuration of ResNet18 shown in Figure 3.6 gives one prediction value for each of the 1,000 class. This fully connected layer is replaced with an fc$n$, where n is the number of classes in our experiments.

The fully connected layer applies linear transformation to incoming data, and produces an output vector v of size $1 \times n$ where n is the number of classes. We then apply softmax function to v to get the prediction probability for each class.

Softmax function interprets its inputs as unnormalized log-probabilities and rescale them so that the they lie in range $[0, 1]$.

$$\text{Softmax}\left(x_i\right) = \frac{\exp\left(x_i\right)}{\sum_j \exp\left(x_j\right)}$$

$argmax(softmax(x))$, that is the class which has the maximum probability value, is the predicted class. Analysis and interpretation of prediction probabilities are reported in Chapter 5.

3.3.3. Model

Different ResNet architectures with 18, 34, 50, 101 and 152 layers are introduced by [2]. In our experiments, ResNet18 -the shallowest ResNet- with 18 layers were fine-tuned. Pytorch provides easy access to ResNet models which have been trained on 1.28 million images of 1,000 categories from ImageNet 2012 classification dataset [37]. Layers of ResNet18 are displayed in Figure 3.6.

Cross entropy loss is used to calculate the loss after going through every batch of samples. Formula 3.4 is taken from official Pytorch documentation [38].

$$\text{loss}(x,\ \text{class}\ ) = -\log\left(\frac{\exp(x[\ \text{class}\ ])}{\sum_j \exp(x[j])}\right) = -x[\text{class}] + \log\left(\sum_j \exp(x[j])\right) \quad (3.4)$$

3.3.4. Hyperparameters

The environmental setup of deep learning experiments requires careful model design and hyperparameter tuning. Hyperparameters differ from the model parameters in a way that they can be pre-determined before the training starts.

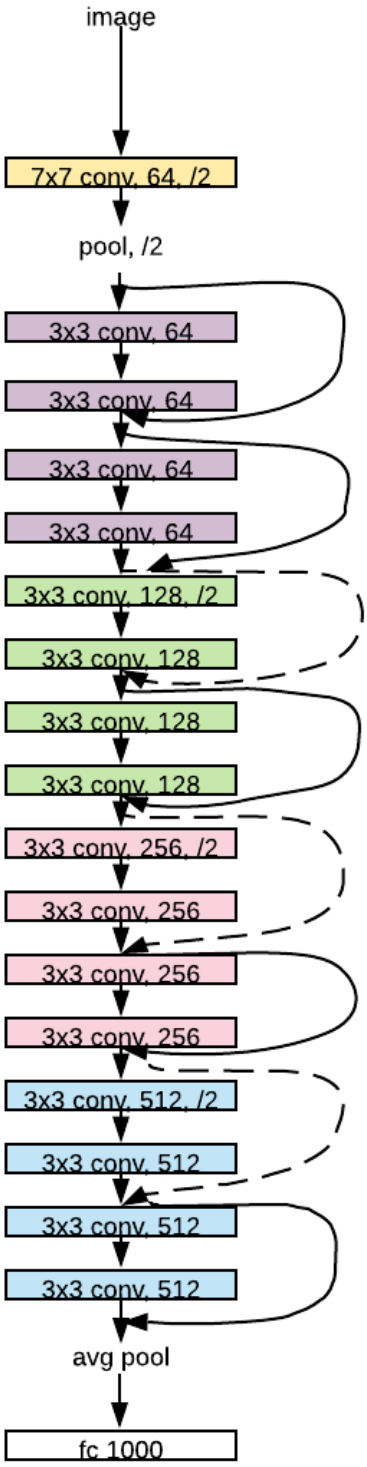


Figure 3.6: ResNet18 architecture.

The learning rate determines the magnitude of change in network weights. It is common practice to decrease the learning rate during training, to prevent overfitting. Overfitting is an issue about weights being too specialized for the training data, therefore lacking the ability to adapt to unseen data.

In order to effectively use the computational resources, samples in the training set can be loaded and processed in batches. The batch size determines the number of samples to process, before calculating loss and updating weights. Recently, mini-batch sizes between m=2 and m=32 have been found to give the best performance [39].

3.3.5. Stochastic Gradient Descent Optimizer

Once the hyperparameters are set, one can use one of the several optimization algorithms available for achieving the model weights which gives the best performance on the training set. This is achieved by updating model parameters, i.e. network weights. Optimization algorithms differ in the way of how the weights are updated.

Stochastic Gradient Descent Optimizer (SGD) is a learning algorithm that works iteratively to optimize the internal parameters of the neural network. For each training sample, the gradients of loss function -in our case Cross Entropy Loss given in Equation 3.4- with respect to the weights are calculated, and SGD updates the weights accordingly.

$$\begin{aligned} v &= \rho \times v + g \\ p &= p - lr \times v \end{aligned} \tag{3.5}$$

The step function of SGD is given in Equation 3.5, which is taken from the official Pytorch documentation [38].

p, g, v and ρ represents parameters, gradient, velocity and momentum respectively. lr is the learning rate. In some cases, SGD fails to progress further; ρ is used for avoiding to get stuck in local minima.

In regular *SGD*, the batch size is one by default, that is, network weights are updated after each sample in the training set is processed. One may also use the *batch gradient descent* and update the weights after processing all training samples.

Alternatively, there is *mini-batch gradient descent*, where we process a constant number of training samples before each update. *Mini-batch gradient* descent is used in our experiments.

3.3.6. Weight Regularization

Regularization is the optional operation of giving a penalty for greater values of weights during the training phase. It helps prevent overfitting and enables the network to generalize better. Regularization can be enabled with weight decay hyperparameter for Pytorch optimizers.

3.3.7. Data Augmentation

Data augmentation is the technique of adding variability to data samples to improve learning. In computer vision tasks, random cropping, random resized cropping, and random flipping are common augmentation operations.

We have employed crop, resized crop and flip for data augmentation in our experiments. Details are given in Chapter 5.

3.3.8. Validation and Testing

It is common practice to not apply data augmentation to image samples from validation and test sets. This procedure helps to report reliable results.

We have also followed the conventional method and only applied mandatory crop and resize operations.

3.3.9. Prediction Analysis

In order to get an insight into predictions, we have plotted the class probability of positive class and negative classes along video frames, with their respective ground truth labels and predicted labels. Ideally, a peak in probability value $p(question)$ is expected in question occurrences throughout the video.

With this analysis, we had the chance to investigate misclassifications along the time axis, the necessity of ground-truth smoothing and prediction postprocessing. Plots and their respective findings are given in Chapter 5.

3.3.9.1. Temporal Filtering. Frame-level classifications are often noisy and need to be post-processed. We observe the following phenomena:

- Head movements and facial expressions start and end instantly
- Instant faulty expressions or movements can occur (i.e. when the subject is distracted, could not catch up)

In both cases, such activities may be easily overlooked by the human eye, and by the annotator. However, the frame-level classification may switch between classes, and deviates from the ground truth.

Thus, determining the negligible amount of misclassified frames to be considered as noise and post-processing accordingly is essential. We have employed two kinds of filtering: Median filtering and minimum frame-count filtering.

(i) *Median Filtering*

Median filtering has been used to correct erroneous frame-level classifications.

Prediction	c_2	c_2	c_2	c_5	c_0	c_5	c_5	c_2	c_2	c_5
Filtered Prediction	c_2	c_2	c_2	c_2	c_2	c_2	c_2	c_5	c_3	c_3
Target	c_2	c_2	c_2	c_2	c_2	c_2	c_2	c_2	c_2	c_2

Figure 3.7: The effect of median filtering with kernel size k = 9 on selected consecutive frames of User 2. c_0, c_2, c_3, and c_5 represent other, negation-side, negation-up-down and affirmation classes, respectively.

Median filters of kernel sizes k = 3, 5, 7, 9, 11 and 13 are used for this purpose. Some misclassified frames of User 2 are given in Figure 3.7, before and after applying the median filter.

(ii) *Minimum Frame Count Filtering*

A predicted annotation tuple is defined as $\hat{a} = (l_c, f_s, f_e)$ where f_s and f_e are frame indices of start and end of the annotation and l_c is the annotation label of class c. We call an annotation tuple *positive* if $l_c \neq other$.

The set of predicted annotation tuples for the ith video sample v_i is $\hat{A}_{v_i} = \{\hat{a}_n = (l_c, f_s, f_e)\}_{n=1}^{N}$ where N is the total number of predicted annotation tuples in v_i.

We introduce the frame count threshold t_{p_c} for each positive class label l_c, which is determined after evaluating the class frame count statistics. Additionally, we evaluated the histograms of the annotation tuple frame counts. With the histogram calculation, we aim to determine a lower bound for annotation of each class label in terms of frame count. Then, videos in which shortest annotation occur were visually examined. Facial expressions and head movements in such videos were observed to be not less significant than the other videos with longer annotations. Thus, we set $t_{p_c} = f_{min_c}/2$ for each class c. For all positive annotation tuples in $\hat{A}_{v_i}$ with $f_e - f_s < t_{p_c}$, we set $l_c = other$.

4. TURKISH SIGN LANGUAGE CORPUS FOR FACIAL EXPRESSION AND HEAD MOVEMENT RECOGNITION

A subset of a Turkish Sign Language video dataset and collection of videos from two distinct TSL projects are studied in this thesis, each of which is explained in this chapter. Each step of the data preparation and preprocessing phases are explained in detail.

First, the corpus of sign videos to be studied is selected based on their semantics. Facial expressions and head movements in these sign videos are temporally annotated. A deep learning based keypoint detector is employed to find the face regions in video frames, before parsing the annotations to obtain labeled video frames. During the parsing operation, a naive approach is employed for merging the chunks of annotations with a negligible amount of time gaps, to prevent possible annotator errors.

4.1. Datasets

4.1.1. BosphorusSign

BosphorusSign [4] is a TSL dataset of sign videos with corpus from health, finance and general domains. BosphorusSign corpus consists of words, compounds and phrases.

There are 188 signs of commonly used phrases, 171 signs of banking and finance phrases and 496 signs of phrases that can be used in a hospital visit.

Six native sign language signers perform each sign for varying number of times, six times on average. A Microsoft Kinect v2 sensor is used to simultaneously capture RGB, depth and skeletal data. RGB videos in BosphorusSign have 960×1080 pixels resolution, which makes it a dataset suitable for studying facial expression recognition.

However, signs and sign phrases in BosphorusSign are mostly one-word, making it relatively challenging to spot facial expressions in short videos. This fact leads us to extend our experiment set to include videos from the HospiSign [5] project.

4.1.2. HospiSign

HospiSign [5] is a community-aid interaction platform designed for hearing-impaired people arriving at the hospital.

HospiSign corpus consists of compounds and sentences from the health domain. There are a total of 41 compounds and sentences, which can be used in a hospital visit scenario. Same users as in BosphorusSign perform the signs in the videos.

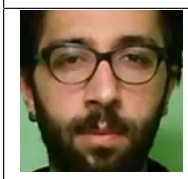
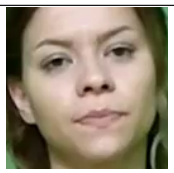
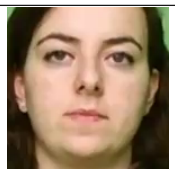
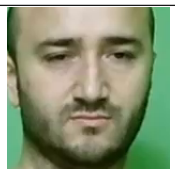
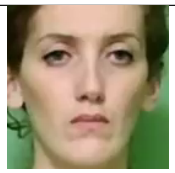
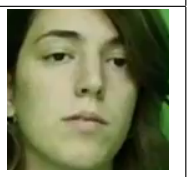
This platform provides three steps to communicate with the patient using a touch screen, a PC and a Microsoft Kinect v2 sensor. First, the question is displayed on the touch screen. Then, possible answers are displayed on the screen. Finally, the answer to the user as a sign is recognized. The first two steps are repeated until enough information is gathered from the patient.

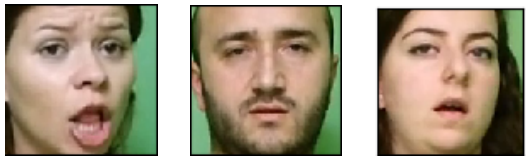
4.1.3. Bosphorus Facial Signs Dataset

A mixture of the corpus from HospiSign and BosphorusSign is used in our experiments. Originally, there are 547 videos in the selected set. However, during the annotation procedure, one of the subjects in the dataset, User 6, is found to be consistently neutral during the signing. After observing that facial expressions and head movements are not articulated in the majority of her videos, this user was excluded from the dataset.

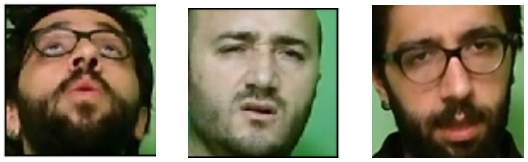
A total of 483 videos out of 547 videos were annotated gradually over time. Six common subjects of both datasets are displayed in Table 4.1. The frame representation of each class is given in Figure 4.1.

Table 4.1: Subjects in the BosphorusSign and HospiSign datasets.

User ID					
User 1	User 2	User 3	User 4	User 5	User 6
					



(a) Question, other and negation-side frames from left to right.



(b) Negation-Up-Down, pain and affirmation frames from left to right.

Figure 4.1: Representative frames of each class.

Each sign in dataset takes 3.50 seconds on average to be performed. Videos are 30 frames per second (FPS). Average duration of videos is given in Table 4.2.

Empirically, one-third of frames in the middle of each video belong to the positive classes. We have observed that the movements of the signer have a similar onset-peak-offset pattern in the videos. Although this pattern helps to accelerate the annotation process, speed variation or movement intensity of the signers and length of different sign sentences were taken into account.

Table 4.2: Average video durations (in sec.) per sign phrase and per user.

Sign Phrase (EN)	User 1	User 2	User 3	User 4	User 5	User 6	Avg. Of Avg. Per Phrase
Chest Pain	3.83	3.73	3.60	3.46	3.89	2.88	3.65
Do You Have an Appointment?	4.33	4.35	4.08	4.84	4.07	3.88	4.32
Headache	3.04	2.90	2.93	3.30	3.21	2.84	3.05
How can I help you	4.39	4.61	4.16	4.69	4.09	3.82	4.32
Insufficient	2.94	3.37	3.09	3.35	2.47	2.52	2.96
Is it urgent?	2.84	2.83	2.74	3.26	3.04	2.53	2.89
My stomach hurts	3.57	3.85	3.57	3.58	3.83	3.12	3.61
No, not available	3.22	3.08	2.86	3.03	3.12	2.78	3.03
No, not urgent	3.72	3.34	3.19	3.89	3.65	3.09	3.50
Not available	2.44	2.88	2.55	3.03	2.57	2.50	2.64
Reluctant	2.76	-	2.90	2.96	2.63	2.84	2.83
Waist Ache	3.78	3.76	3.48	3.72	3.35	3.03	3.55
What Information Do You Want	5.04	4.37	4.61	4.74	4.51	4.51	4.64
What is your complaint?	3.90	4.27	3.64	4.20	3.76	3.62	3.92
Yes, there is	3.03	3.05	2.92	3.26	3.17	2.61	3.03
Yes, It is an Emergency	3.19	3.21	3.04	3.18	3.27	2.73	3.13
Avg. Of Avg. Per User	3.50	3.65	3.35	3.71	3.45	3.10	3.48

Class distrubition and subset information of videos are given in Table 4.3. Videos with negation, affirmation, pain expressions and movements were relatively few in the corpus. This yields an imbalanced annotated dataset.

Table 4.3: Class distribution of videos from HospiSign (HS) and BosphorusSign (BS) subsets.

Sign ID	Sign phrase (TR)	Sign phrase (EN)	Source Dataset	Class Label	Number of Videos
7	Acil mi?	Is it urgent?	HS	question	35
69	Baş Ağrısı	Headache	HS	pain	34
85	Belim Ağrıyor	Waist Ache	HS	pain	34
221	Evet Acil	Yes, It is an Emergency	HS	affirmation	34
222	Evet Var	Yes there is	HS	affirmation	34
247	Göğüs Ağrısı	Chest Pain	HS	pain	40
277	Hayır, Acil Değil	No, not urgent	HS	negation	34
278	Hayır Yok	No not available	HS	negation	34
316	İsteksiz	Reluctant	BS-Health	negation	26
353	Karnim Ağrıyor	My stomach hurts	HS	pain	34
424	Nasıl Yardımcı Olabilirim	How can I help you	HS	question	36
535	Siz Ne Bilgisi İstiyorsunuz	What Information Do You Want	HS	question	37
536	Sizin Randevunuz Var Mı?	Do You Have an Appointment?	HS	question	40
537	Sizin Şikayetiniz Nedir	What is your complaint?	HS	question	36
637	Yetersiz	Insufficient	BS-Finance	negation	27
642	Yok	Not available	BS-General	negation	32

Class labels that are given in Table 4.3 are derived from semantics; they do not guarantee the occurrence of the movement in the video.

4.1.4. Audio Description Association (SEBEDER) Film Archive Dataset

Audio Description Association (SEBEDER) is the first and only association that is founded for delivering the written, visual and auditory media to the visually impaired and the deaf society, simultaneously with the rest of the public. The association is unofficially founded in 2006 at Boğaziçi University Mithat Alam Film Center and has been professionally active since 2010 as the SEBEDER.

SEBEDER has collaborated with Boğaziçi University Assistive Technology and Education Laboratory for Individuals with Visual Disabilities (GETEM) to publish the audio descriptions of various books via a catalog website [40].

Outside the university, SEBEDER has published its archive website [41] and provided audio descriptions, detailed subtitles and sign language translations of various films to the members of the association.

SEBEDER has shared 73 films from its archive with the Perceptual Intelligence Laboratory researchers for sign language and natural language processing research. There are 21 Turkish films in this dataset. Sign language translations of these films are available in the video format, in which a professional sign translator signs each cue while watching the film scenes.

4.1.4.1. Obtaining Non-Manual Sign Clips from SEBEDER Film Translation Videos. Sign language translation of one film is available in approximately three or four long clips. These clips include the hesitance moments and time-outs, thus requires trimming. Similar to the BosphorusSign dataset, SEBEDER clips are in green background. Video frames have 640×480 resolution and the frame rate is 25 FPS.

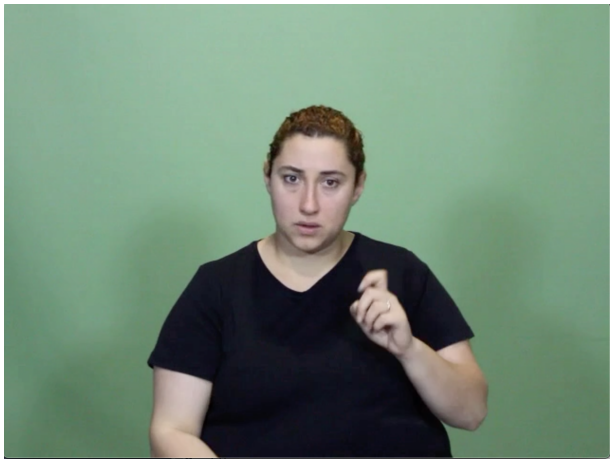


Figure 4.2: SEBEDER sign language translation clip frame.

Within the timeline of this thesis, only one of the available Turkish films were prepared, annotated and processed to conduct recognition experiments. A colleague in PILAB has used the keywords from movie subtitles to trim the sign language translation clips. 97 short clips are obtained with non-manual cues of negation and affirmation. The average duration of a clip is 4 seconds.

- For negation: ‘No’, ‘No, not available’, ‘Not available’, ‘No way’, ‘Did not happen’, ‘Does not come’, ‘Not’ keywords were used.
- For affirmation: ‘Yes’, ‘Yes, there is’, ‘There is’, ‘There exists’, ‘Okay’, ‘Alright’ keywords were used.

With the help of our colleague, we annotated the non-manual cues in obtained short clips with the same annotator tool and the terminal application that are described in Chapter 4.2.1. 9,722 video frames are obtained after parsing the annotations. Class distributions of these frames are given in Table 4.4.

Table 4.4: Frame count of class labels after parsing annotations.

Class Label	Affirmative	Negation-side	Negation-up-down	Question	Other	Exceptional
Count	1328	191	499	376	7110	218
Total			9722			

The intended class labels of the videos were affirmation, negation-side and negation up-down. However, several occurences of the question class and a new class of non-manual sign was observed during the annotation. While the subject is signing the phrase ‘Is that so?’, she shakes her head confirmingly with an inquiring facial expression. Thus we call this movement ‘exceptional’ and discarded the related clips of this class.

As these signs are not originally performed for research purposes, additional challenges occur. These challenges can be summarized as:

(i) Sign videos are recorded in a challenging, less controlled setup.
(ii) The pace of movements of the signer is much higher when compared to BosphorusSign Facial Signs Dataset videos.
(iii) Naive method to trim the video clips provide limited data to be annotated.

Thus, partly due to the rapidly-developing incidents in movie scenes, facial expressions and head movements of the signer are not as precise.

4.2. Video Annotation

4.2.1. ELAN Multimedia Annotation Tool

For video classification tasks, videos can be annotated in several different ways depending on the specific problem. In the case of activity recognition tasks, people and actions of interest are localized in both space and time. In sign language videos, typically a single signer would stand in a stable location, only moving the upper body parts while signing. For the scope of this study, we are not interested in the spatial localization of the signer, which lies already in a restricted area. The temporal localization of manual signs is also out of the scope of this study. Instead, we are interested in the temporal localization of specific head movements and facial expressions of the signer.

ELAN, a linguistic annotation software developed by the Max Planck Institute for Psycholinguistics [35] is used for the annotation task.

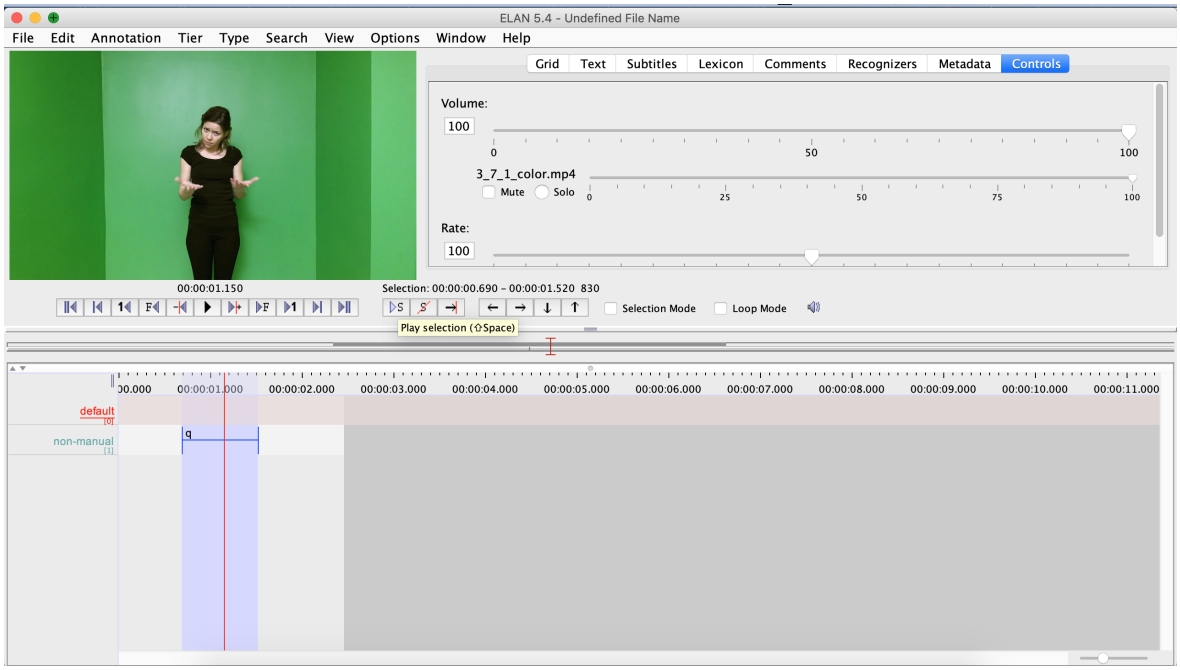


Figure 4.3: ELAN linguistic annotation software interface

This software provides a broad menu for selecting, playing and annotating a highlighted time interval within the video as seen in Figure 4.3. It allows annotation of more than one tier. Since we were interested in the signer's facial expressions and head movements, we have decided on the name "non-manual" for the tier of interest.

The video annotation process requires a careful evaluation of each video in the dataset. Pympi [42], a linguistic Python module for processing ELAN annotation files was used as an interaction tool with the ELAN software. With the help of this module, the non-manual tier is added to all files automatically. Although ELAN does not support multiple file annotations, with the help of Pympi, a terminal application is developed so that the desired number of videos can be annotated one after another in batches.

4.2.2. Selecting Sign Videos Semantically

Despite the recognition of signs and/or sign phrases goes beyond the scope of this study, we were still able to exploit the semantics of signs. Intuitively, most significant facial expressions and head movements were expected to occur in questions and negation utterances. As a starting point, literal questions in the dataset were found. Following this, with the same direct approach, literal negation and affirmation sentences were found.

The majority of the phrases in the HospiSign dataset, for its obvious design purposes, are suitable for hospital information desk scenarios. Therefore, there are multiple patient complaints. Although such phrases do not imply grammatical negation, a pattern with grimace and pain expressions was observed in signers' faces. The pattern was most visible when the subjects are signing to complain about a particular ache, or were signing the particular ache as a noun. All such videos were labeled as pain class.

In this context, searching for a question, negation and affirmation expression and movement is rather an objective task when compared to searching for pain expressions.

4.2.3. Signer Related Diversity

Before the annotation procedure, several random videos were visually examined to make sure that expected head movements and facial expressions occur. Despite this examination, challenges regarding the signers naturally existed. Major challenges were found to be the variety in terms of pace and intensity of the head movements and facial expressions. Considering the structure of sign language, hand movements are expected to be relatively standardized when compared to head movements. Similarly, as observed in our dataset, facial expressions vary from signer to signer. For instance, User 6 is the one who displays the least intense expressions within the dataset, with an almost always neutral expression on her face. User 6 is excluded from the dataset, to partially balance the dominating number of neutral *other* frames.

Another issue is about intra-user variance. Users who display high expression intensity do not necessarily perform similarly in all sign videos. For instance, User 5 -one of the most articulate signers as empirically observed throughout the dataset- displays rather neutral expressions and low-intensity head movements in a particular sign. Figure 4.4 displays one such comparison, where video frames of the same sign video are sampled with 1/10 frame rate from approximately equal length videos of the two users.

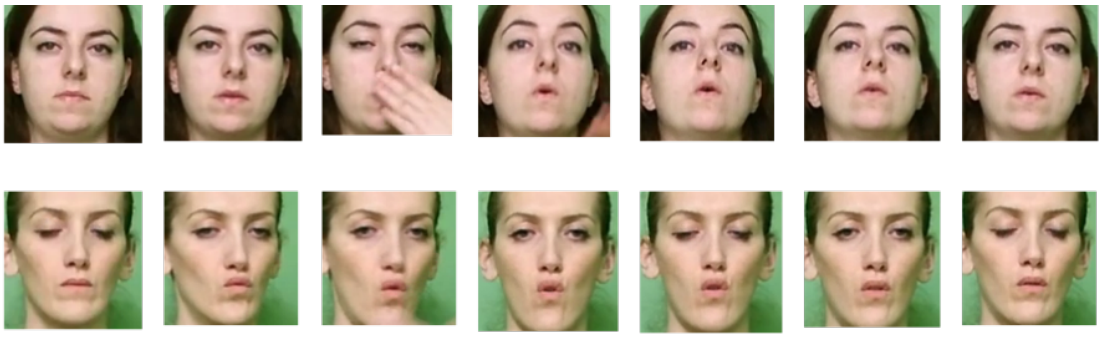


Figure 4.4: Video frame samples of User 3 (top) and User 5 (bottom) performing the sign “Not available”.

4.2.4. Transition

The annotation procedure requires multiple examinations on each video to avoid human error as much as possible. To prevent human errors, especially at the beginning and end phases of movements and expressions, we have revised the annotations of question class and introduced two new class labels: Transition to question and exception. Transition to question represents the phase in the video, where the subject is preparing for the upcoming question expression, therefore not displaying a neutral expression. The exception class label is used for rare cases, where the movement and expression of the signer could not be identified.

After investigation of the preliminary experiment results, question labels were merged with transition-to-question labels and exception labels were merged with other labels. Following this procedure, the next set of videos are annotated less tightly. Thus, the transition phase exists implicitly.

4.2.5. Parsing Annotations

An annotation tuple is defined as $a = (l, t_s, t_e)$ where t_s and t_e are starting and end time of annotation in milliseconds and l is the annotation label.

We define the set of annotation tuples for the ith video sample v_i as $A_{v_i} = \{a_n = (l_n, t_{n_s}, t_{n_e})\}_{n=1}^{N}$ where N is the total number of ground truth annotation tuples in v_i.

Approximately 2/3 of each video frame was regarded as background and labeled as *other*. This yields the vast majority of all frames to be from *other* class. For balance purposes, 1/3 of *other* frames were randomly sampled to be used in the first set of experiments. However, this approach violates the integrity of sign videos, therefore is deprecated in further experiments.

After each annotation procedure, all video frames are labeled with the label of their respective timestamp. Actual frame extraction was carried out after the face detection and face crop steps explained in Chapter 3.2.

5. EXPERIMENTS AND RESULTS

5.1. Experimental Setup

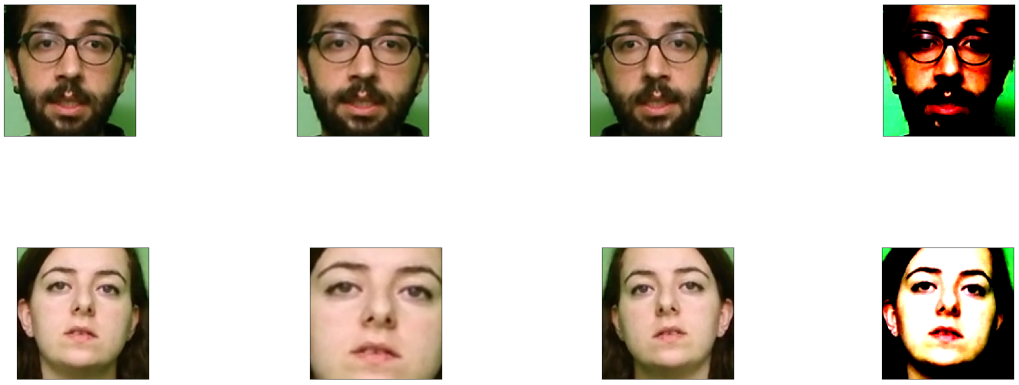
Learning rate, number of training epochs and training batch size are the hyper-parameters of our system.

In our experiments, the initial value of the learning rate is set to 0.001 and is decreased after every seven epochs by ratio 0.1. We have used the mini-batch gradient descent and set the batch size to 100 training samples.

5.1.1. Data Augmentation

Every image in the training set is randomly cropped to obtain patches of size 224x224, which is the input size of ResNet. Horizontal flipping is then randomly applied with a 50% probability on the training set, i.e. half of the training samples would be horizontal flipped. We do not employ a vertical flip as an upside-down face image is not a usual scene.

Images are then converted to Tensors. Tensors are normalized with respect to mean values 0.485, 0.456, 0.406 and standard deviaton values 0.229, 0.224, 0.225 for red, green and blue channels respectively.



Original Cropped Flipped Normalized

Figure 5.1: Effect of data transforms when applied seperately on selected frames of User 1 (top) and User 3(bottom).

5.1.2. Validation and Testing

Samples were resized to 256x256 (not randomly) and were cropped from the center to obtain images of size 224x224. Images are then converted to tensors and normalization procedure is the same with the training phase.

The same data preparation operations as in the validation phase are applied to the test samples.

5.1.3. Evaluation Metrics

For frame-level performance evaluation, precision, recall, and f1-score are calculated per class label. Accuracy values for each test user are also reported; however, these values reflect the inflated performance estimates considering the major number of negative class frames.

Micro, macro and weighted average values of f1-score, precision, and recall metrics are also reported. The micro average is calculated with the total true positives, false negatives and false positives; without considering the class labels separately.

In other words, the sum of dividends of each metric is divided by the sum of denominators. The macro average is the unweighted average of each class label metric. The weighted average is the average of each class label metric, weighted with the support -number of samples- of each class.

In order to take this imbalance into account, we report the balanced accuracy score, which is defined as the macro average of recall values for each class label, excluding the *other* class label. Weighted average values are calculated for each metric, again excluding the *other* class label.

Annotation-based performance evaluation is essential in our experiments. As an additional evaluation metric, we calculate Intersection Over Union (IoU) of annotations for each video. We calculate the annotation level accuracy and report the results for each test user separately. This measurement is based on how well the ground truth annotations intersect with the predicted annotations. A demonstration of score calculation is shown in Figure 5.2.

For a given video v_i, IoU is calculated for each pair of ground truth annotation a in A_{v_i} and predicted annotation $\hat{a}$ in $\hat{A}_{v_i}$ where $l_a = l_{\hat{a}}$ as follows:

Let $a_i = (l_a, f_{s_a}, f_{e_a})$ and $\hat{a}_j = (l_{\hat{a}}, f_{s_{\hat{a}}}, f_{e_{\hat{a}}})$ where l, f_s and f_e denote the class label, starting frame index and ending frame index of the annotation respectively.

$$
\begin{aligned}
area_{intersection} &= min(f_{e_a}, f_{e_{\hat{a}}}) - max(f_{s_a}, f_{s_{\hat{a}}}) + 1 \\
area_{union} &= (f_{e_a} - f_{s_a} + 1) + (f_{e_{\hat{a}}} - f_{s_{\hat{a}}} + 1) - area_{intersection} \\
IoU(a_i, \hat{a}_j) &= area_{intersection} / area_{union}
\end{aligned}
\tag{5.1}
$$

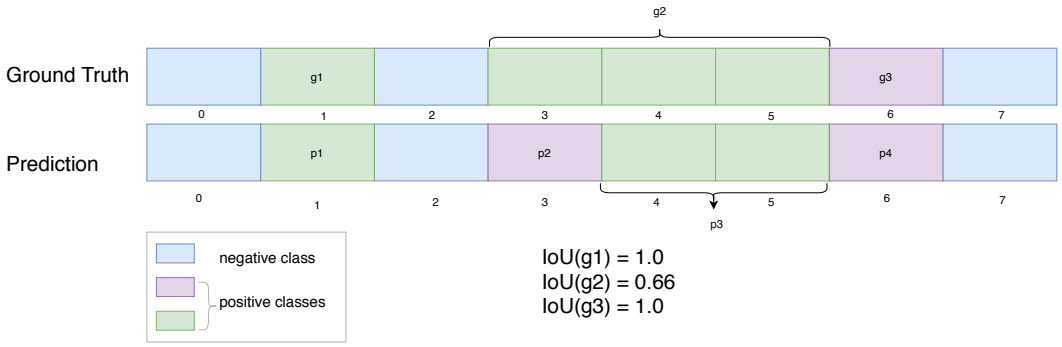


Figure 5.2: IoU score calculation.

$IoU_{a,\hat{a}^*}$ denotes the maximum IoU score that is calculated for the ground truth annotation a. Out of the candidate predicted annotations in $\hat{A}$, a is assumed to match with the prediction $\hat{a}^*$, which gives the greatest IoU score.

We assign the annotation a to be correctly classified if $IoU_{a,\hat{a}^*} > t\text{-}IoU$. We use the threshold range $T\text{-}IoU = [0.3, 0.7]$. and we calculate the annotation level accuracy for each threshold value $t\text{-}IoU$ in $T\text{-}IoU$.

5.2. Results

The aforementioned datasets introduced in Chapter 4 are not off-the-shelf and require careful temporal annotation. Annotation procedure required time and manual effort. Therefore the experiments were conducted simultaneously with the annotations and the set of labeled videos gradually enlarged. Results that are reported in this chapter are obtained by experimenting with the whole set of labeled videos from the aforementioned datasets.

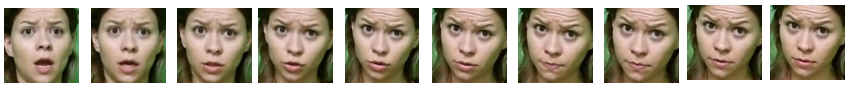
The results of two experiment setups are reported in this section:

(i) Transfer learning for spotting the question, negation, pain, and affirmation in BosphorusSign-HospiSign dataset videos

(ii) Cross-database test with the SEBEDER dataset videos

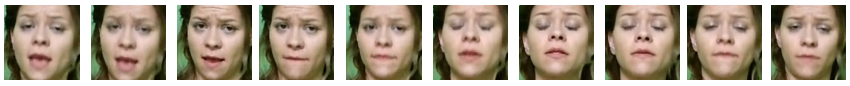
5.2.1. Spotting the Question, Negation, Pain, and Affirmation in Bosphorus Facial Signs Dataset Videos

This corpus contains 483 videos that are temporally annotated with respect to movements and facial expressions of interest. 51,186 labeled frames are obtained after the annotation procedure. All background frames are labeled as the "other" class. Additionally, two distinct movement patterns are observed in negation videos; side to side headshake and up-down head nod. Therefore, the negation class is split into two separate classes: Negation-Side and Negation-UpDown. A total of six distinct labels are obtained for the classification task. Class distribution of video frames in training and test sets are given in Table 5.1. Class labels are represented with selected keyframes given in Figure 5.3.

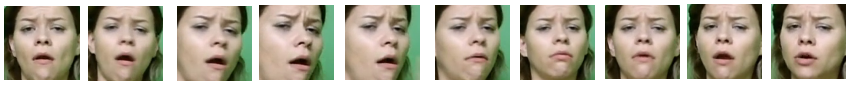


(a) Question key frames.

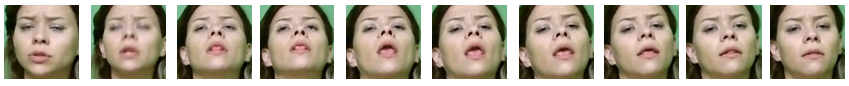
(b) Pain key frames.



(c) Affirmation key frames.



(d) Negation-side key frames.



(e) Negation-up-down key frames.

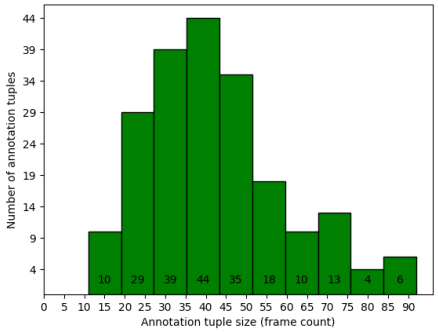
Figure 5.3: Selected key frames of each class label.

Table 5.1: Training and test splits for each user fold.

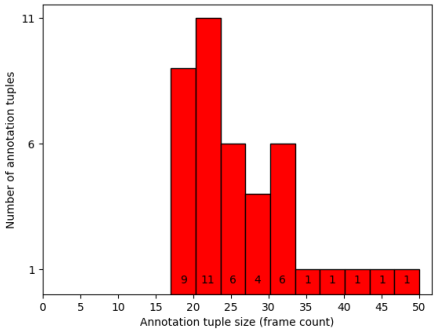
		Class Labels						
Partitions		Affirmative	Negation-Side	Negation-UpDown	Other	Pain	Question	Total
Test User 1	Train	1614	784	2099	26427	3094	6975	40993
	Test	456	293	741	5705	1141	1857	10193
Test User 2	Train	1571	749	2365	27038	3155	6683	41561
	Test	499	328	475	5094	1080	2149	9625
Test User 3	Train	1666	931	2124	25433	4115	7480	41749
	Test	404	146	716	6699	120	1352	9437
Test User 4	Train	1803	1054	2183	23688	3711	7280	39719
	Test	267	23	657	8444	524	1552	11467
Test User 5	Train	1626	790	2589	25942	2865	6910	40722
	Test	444	287	251	6190	1370	1922	10464
Total	Train	204744						
	Test	51186						

As mentioned earlier in Chapter 3, ground truth annotations are smoothed before extracting the video frames. For ground truth smoothing, we set $t_g = 100ms$ empirically, and merge the ith annotation pair, if $g_i < 100ms$. This smoothing helps to close the negligible gaps between ground truth annotations caused by possible human annotator errors.

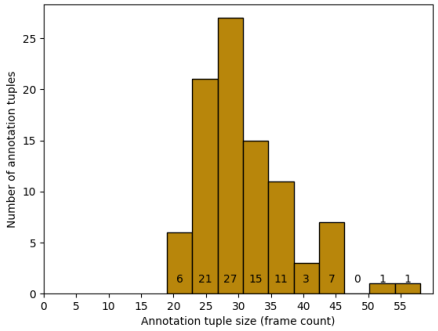
For minimum frame count filtering, annotation blocks of all classes except for the null class *other* are taken into account. First, frame count histograms for each class of annotations are calculated as given in Figure 5.4. We observe from Figure 5.4 that the lower bound is class-specific and that the duration of the action of interest varies widely. The sign videos which include the action with minimum duration were visually examined to make sure that the head movements and/or facial expressions are precise, despite the short durations. After this, we set the minimum frame count thresholds as follows: $t_{p_{question}} = 5$, $t_{p_{negation-side}} = 8$, $t_{p_{negation-up-down}} = 9$, $t_{p_{pain}} = 12$, $t_{p_{affirmation}} = 8$.



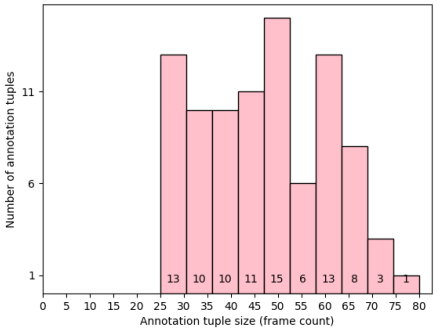
(a) Question.



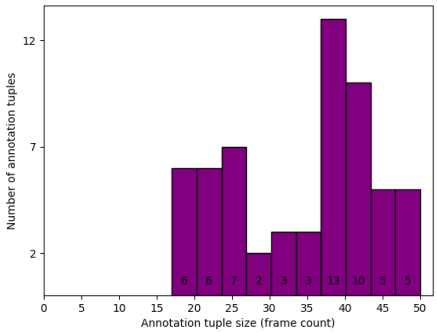
(b) Negation-side.



(c) Negation-up-down.



(d) Pain.

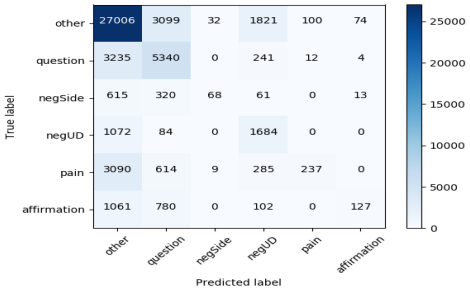


(e) Affirmation.

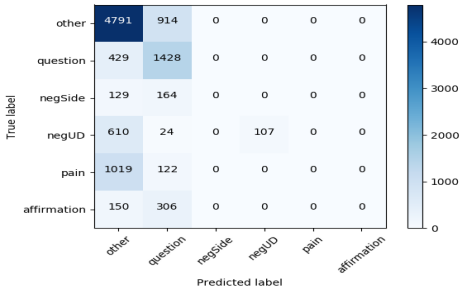
Figure 5.4: Histogram of annotation tuples with ground truth positive class frame counts.

All of the class labels do not necessarily appear in the predicted labels of each test user fold. For convenience, performance measurements given in Table 5.2 are calculated for all test users jointly. Reported results are obtained after applying median filtering with kernel size $k = 9$ and minimum frame count filtering with $t_p = f_{min}/2$.

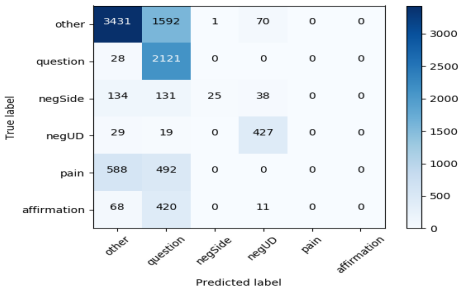
Accuracy values for each user fold that are represented with confusion matrices in Figures 5.7b, 5.7c, 5.7d, 5.7e and 5.7f are calculated as 60 %, 61 %, 75 %, 63 % and 69 % respectively. As these values are inflated with the true positives of the null class *other*, we calculate the balanced accuracy score excluding the null class. Macro average of the recall values that are calculated for each class label denote the balanced accuracy score as explained earlier in Chapter 5.1.3. The frame-level balanced accuracy is 28 %.



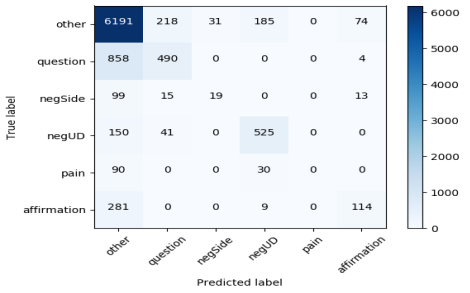
(a) Overall performance.



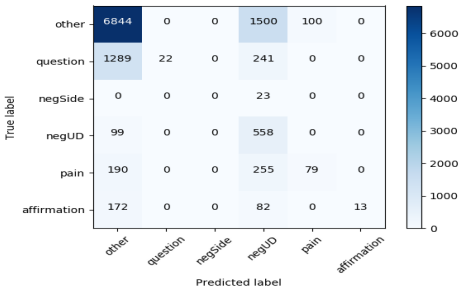
(b) Test User 1.



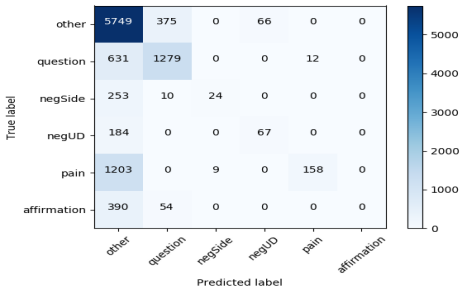
(c) Test User 2.



(d) Test User 3.



(e) Test User 4.



(f) Test User 5.

Figure 5.5: Overall confusion matrix and confusion matrices of each test user.

Table 5.2: Frame-level performance measurement for non-manual sign recognition. Average values are calculated excluding the null class label "other". Balanced accuracy is denoted with *.

	Class Labels						Averages without Null Class		
	other	affirmation	negation-side	negation-up-down	pain	question	Micro Avg.	Macro Avg.	Weighted Avg.
f1-score	0.79	0.11	0.11	0.48	0.10	0.56	0.44	0.27	0.37
precision	0.75	0.58	0.62	0.40	0.68	0.52	0.49	0.56	0.55
recall	0.84	0.06	0.06	0.59	0.06	0.60	0.39	0.28*	0.39

We observe from the high recall value of *other* class, that our recognition system accurately distinguishes background frames from the positive class frames. This is due to the dominating number of samples from *other* class, which naturally arises from signing patterns in the videos. An average sign video takes 3.53 sec. and the subject performs the sign or sign phrase in approximately 1.34 sec. Thus, positive frames make less than half of all video frames.

Confusion matrices for each test user and the overall confusion matrix are given in Figure 5.5. The effect of the imbalanced dataset can be observed in Figure 5.7a, where a remarkable amount of samples from each class is confused with the *other* class. The majority of the video frames of Test User 1, 2, 3 and 5 are assigned to the *question* class. It is seen that *question* class -which is the second most populated class- is learned well by the three out of five trained models. Specifically, the model trained for Test User 2 can successfully recognize the question frames.

From Table 5.2, we find that the negation-up-down class shows similar recognition rate with the question class, despite having only $\sim 1/3$ as many training samples as question class. Though, for question and negation classes (both of them), the subjects display relatively standardized movements and expressions. Slightly tilted head and raised eyebrows for the question; side to side headshake or head nod for negation are the characteristics of these classes. However, such precise characteristics are relatively hard to define for affirmation and pain classes.

We observe particularly low recall values for affirmation, negation-side and pain classes in Table 5.2. Ideally, we want high precision and high recall scores for each class. However, for some scenarios, a high recall value may be much helpful. Considering a hospital scenario where our classifier is employed for determining whether the patient is in pain or not. A high recall value for pain class means that the system correctly identifies the majority of the patients who are in actual pain. A high precision value for pain class means that, out of the patients whom the system classifies as in pain, the rate of patients who are in actual pain is high. Correctly identifying the patients who are in actual pain is more important in this case. Therefore an improvement in the training phase of these three classes is required.

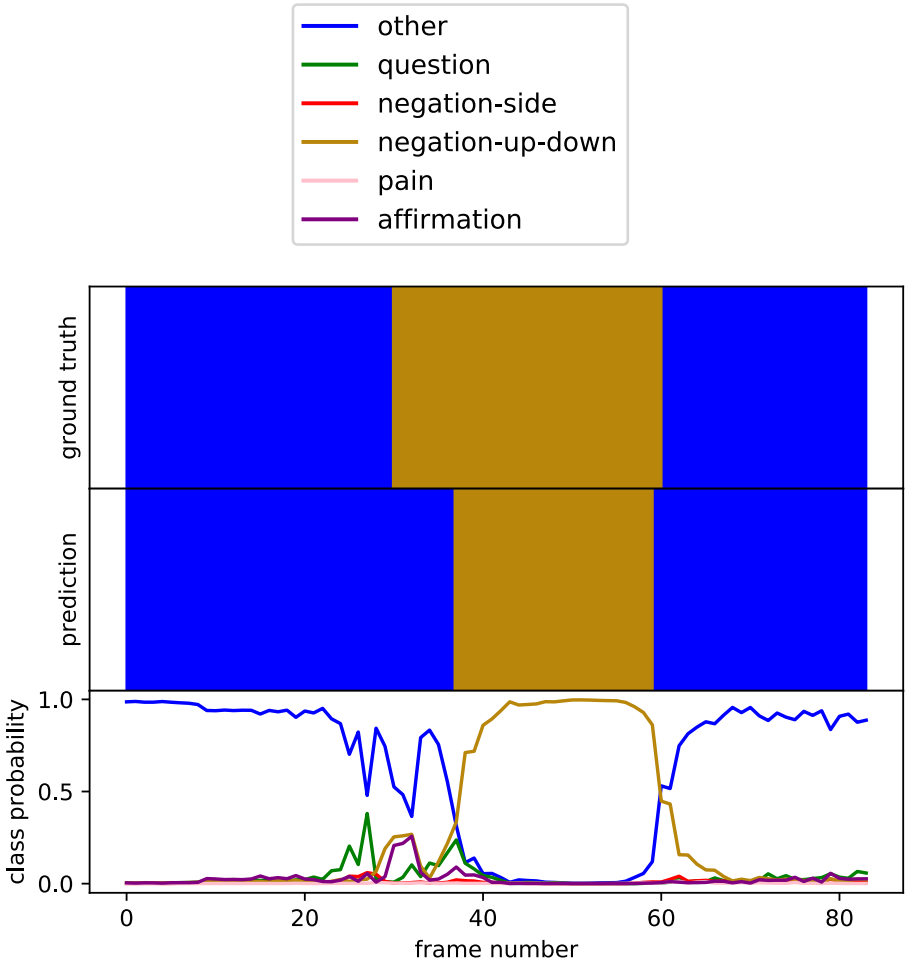
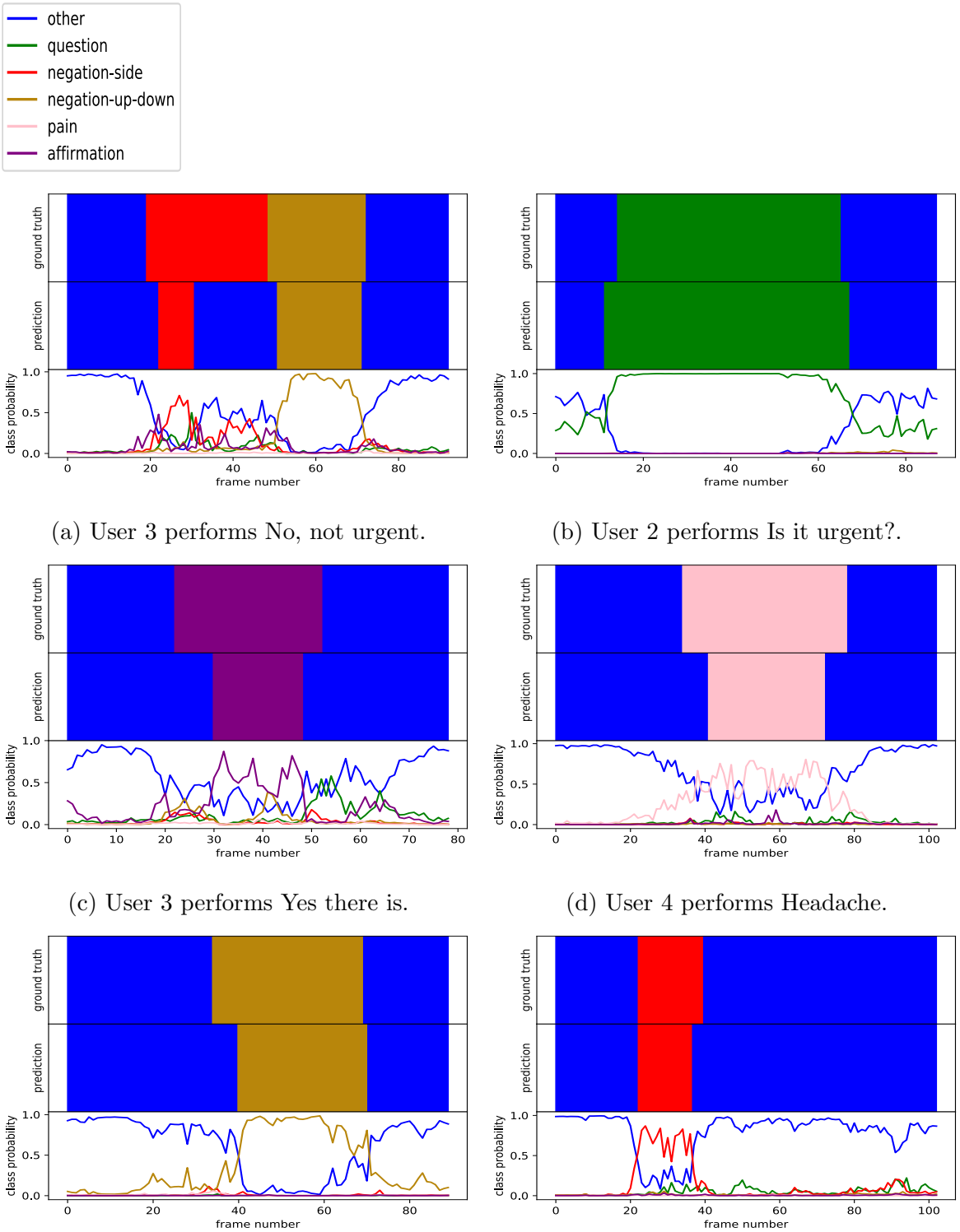


Figure 5.6: Colored comparison of ground truth vs. precited frame labels with class prediction probabilities.

For a clear interpretation of IoU based performance measurements, prediction probability bar plots of selected test videos are plotted as given in Figure 5.6. Selected test videos with different qualitative performances are given in Figure 5.7 and Figure 5.8. We observe that the frame-level predictions of given test videos in Figure 5.7 frequently intersect -either lie within or comprise- with the temporal borders of ground-truth annotations.

We observe the tendency of being misclassified as the question class in the negation-up-down, negation-side, pain, affirmation class frames in Figure 5.8. Also, more commonly in pain and affirmation videos, the system does not necessarily predict any positive labels for the video frames, as can be seen in Figure 5.11.

There are a total of 75 videos in the test set, which do not have any frames labeled with positive classes. That is, the signers in these videos do not display any non-manual sign, causing all video frames to be automatically labeled as the other class. Such videos were not discarded from the dataset, to evaluate the relevance of the retrieved neutral/background frames.



(a) User 3 performs No, not urgent.

(b) User 2 performs Is it urgent?.

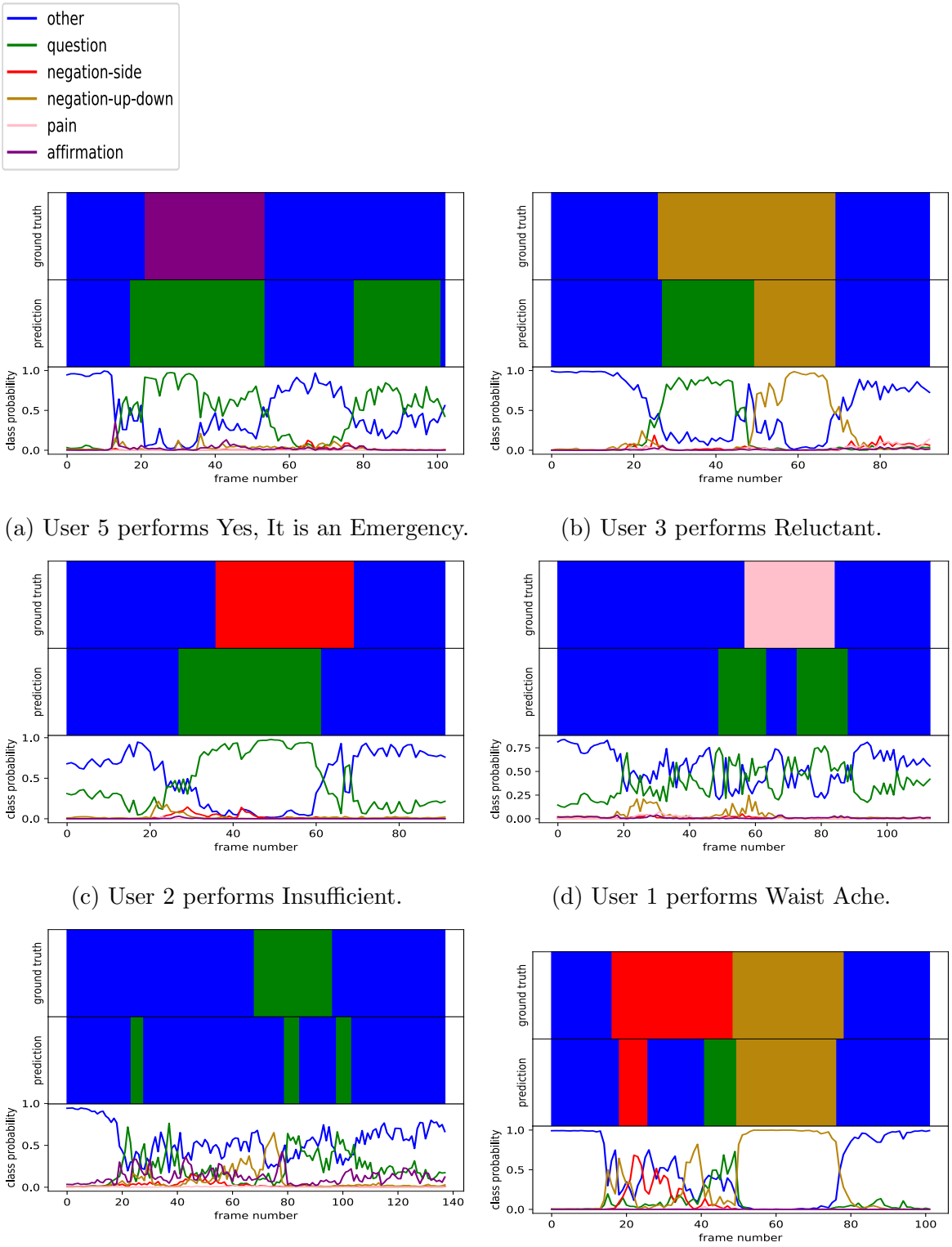
(c) User 3 performs Yes there is.

(d) User 4 performs Headache.

(e) User 4 performs Not available.

(f) User 5 performs No not available.

Figure 5.7: Prediction and ground-truth labels given with prediction probabilities of each class.



(a) User 5 performs Yes, It is an Emergency.

(b) User 3 performs Reluctant.

(c) User 2 performs Insufficient.

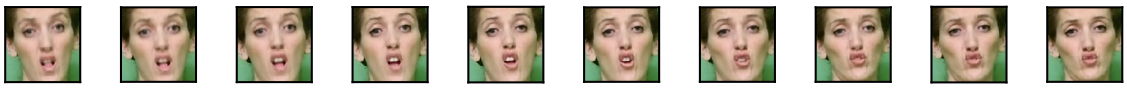
(d) User 1 performs Waist Ache.

(e) User 3 performs Do You Have an Appointment?.

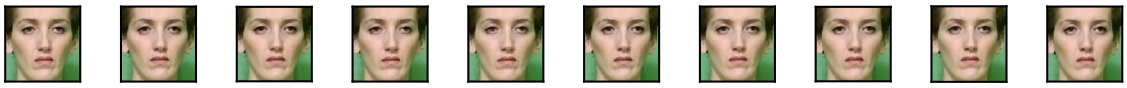
(f) User 2 performs No, not urgent.

Figure 5.8: Prediction and ground-truth labels given with prediction probabilities of each class.

Problematic regions of selected videos represented in Figure 5.8 are given in Figure 5.9 and 5.10. Affirmation frames in Figure 5.9 and negation-up-down frames in Figure 5.10 are indeed ambigious when the facial expression is taken into account.



(a) Affirmation frames that are misclasified as question frames.



(b) Null class frames that are misclassified as question frames.

Figure 5.9: Misclasified frames of the video represented in Figure 5.8a.

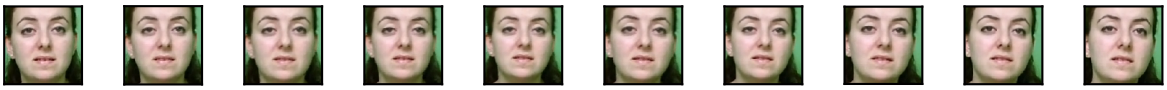
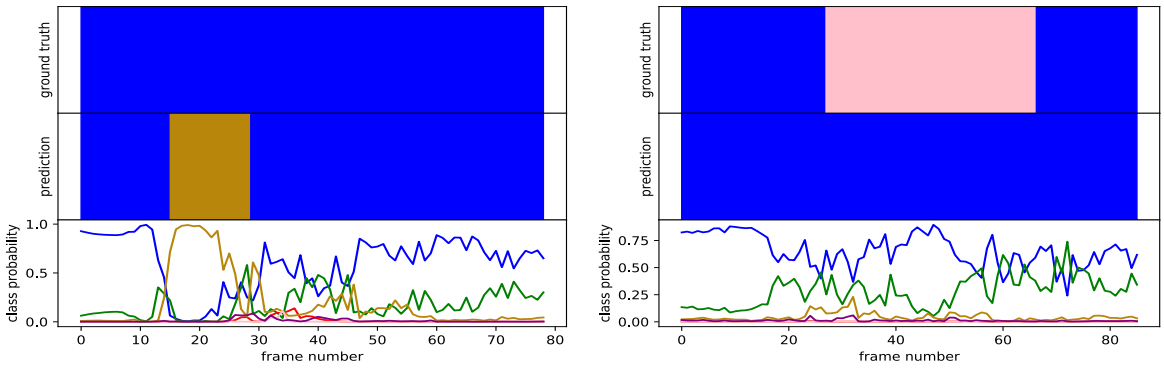


Figure 5.10: Negation-up-down frames that are misclassified as question frames, from the video represented in Figure 5.8b.



(a) User 5 performs Not available.

(b) User 2 performs Headache.

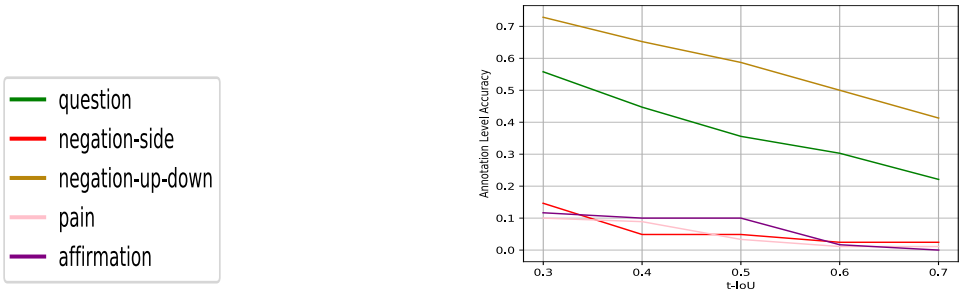
Figure 5.11: Selected test videos without positive ground-truth (left) and without positive prediction (right).

Annotation accuracy of each Test User fold and overall accuracy calculated with thresholds $t\text{-}IoU = [0.3, 0.7]$ are given in Figure 5.12. Accuracy is calculated for each class label, as the rate of correctly classified annotations to the total number of annotations. Statistics of ground truth annotations are given in Table 5.3.

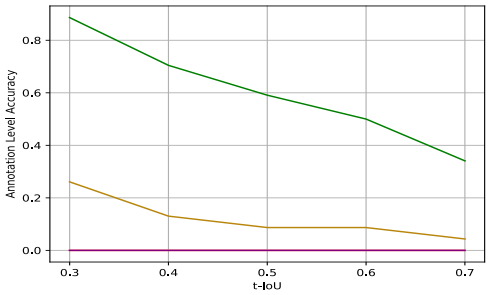
Table 5.3: Count of ground truth annotations per class label in each test user fold.

	question	negation-side	negation-up-down	pain	affirmation
Test User 1	44	12	23	23	12
Test User 2	48	12	15	22	12
Test User 3	35	7	22	4	12
Test User 4	48	1	23	15	12
Test User 5	33	9	9	26	12
Total	208	41	92	90	60

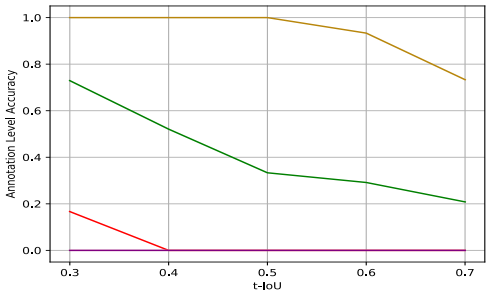
IoU plots given in Figure 5.12 verify that the system is more successful at recognizing negation-up-down and question classes. Negation-side is the least populated class in terms of ground truth frame count. This is reflected in its accuracy levels given in Figure 5.12a. Although having a relatively great number of training samples, pain class annotations have the lowest level of accuracy.



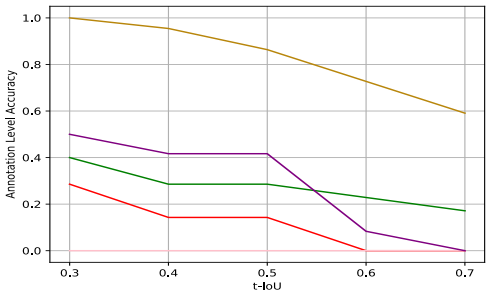
(a) Overall annotation level accuracy.



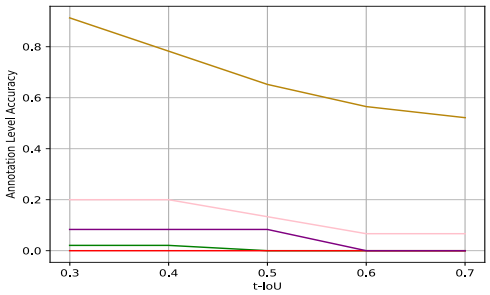
(b) Test User 1.



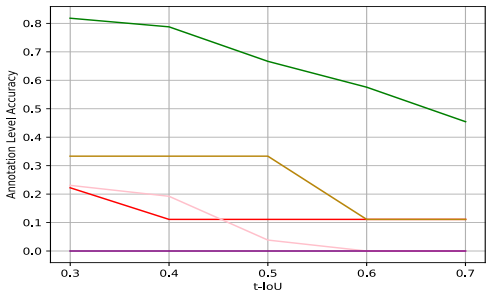
(c) Test User 2.



(d) Test User 3.



(e) Test User 4.



(f) Test User 5.

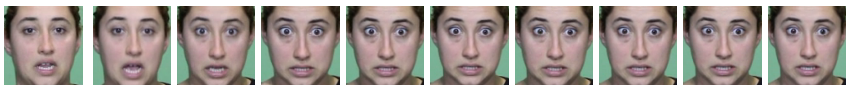
Figure 5.12: Accuracy of ground truth annotations versus different t-IoU values.

Generally, we find that the signer specific movements visibly affect the IoU scores for affirmation and pain classes.

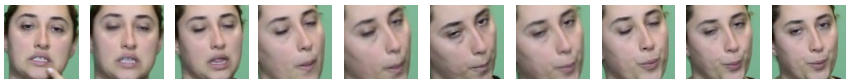
As mentioned earlier in Chapter 4, these two classes do not have as standard movements and expressions as the question and negation classes. The negation-side class has the least amount of training and test samples in our experiments, which explains its low recognition performance in general. Despite that, we find that the negation-side signs displayed by User 3 and User 5 are recognized better from Figures 5.12d and 5.12f. These users are more articulate with the side-to-side head shake.

5.2.2. Cross-database test with the SEBEDER dataset videos

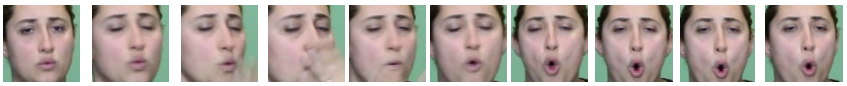
Affirmation, negation-side, negation-up-down and question frames are obtained from sign language translation clips of a selected film as mentioned earlier in Chapter 4.1.4. Representative sequences of each class are given in Figure 5.13. The models from the previous experiment which are trained using the leave-one-subject-out method are employed in this experiment. We report the classification results of each model and the overall performance, as well as the IoU based annotation level accuracy in this section.



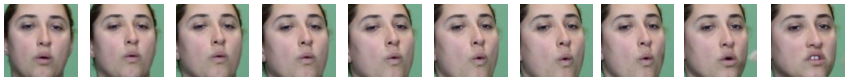
(a) Question key frames.



(b) Affirmation key frames.



(c) Negation-side key frames.



(d) Negation-up-down key frames.

Figure 5.13: Selected key frames of each class label in SEBEDER.

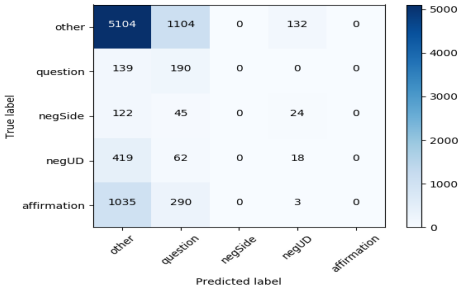
Table 5.4: Training and test splits for each partition fold in SEBEDER experiments.

	Class Labels						
Partitions	Affirmative	Negation-Side	Negation-UpDown	Other	Pain	Question	Total
Train	1614	784	2099	26427	3094	6975	40993
Test	1328	191	499	6340	-	329	8687
Train	1571	749	2365	27038	3155	6683	41561
Test	1328	191	499	6340	-	329	8687
Train	1666	931	2124	25433	4115	7480	41749
Test	1328	191	499	6340	-	329	8687
Train	1803	1054	2183	23688	3711	7280	39719
Test	1328	191	499	6340	-	329	8687
Train	1626	790	2589	25942	2865	6910	40722
Test	1328	191	499	6340	-	329	8687
Train Total	204744						
Test Total	43435						

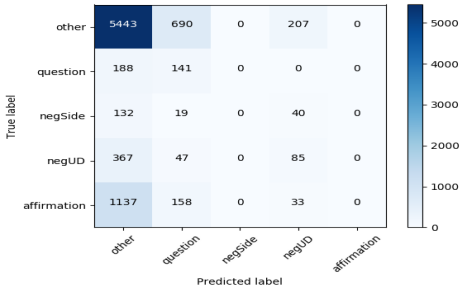
All of the aforementioned methods are applied to the SEBEDER video frames during annotation, preprocessing, face image cropping as in the previous experiment. Similarly, median filtering and minimum frame count filtering are applied to the results reported. We find that the median filter of size $k = 3$ performs the best and report the results respectively.

After analysing the histogram of annotation tuples for SEBEDER clips, we set the minimum frame count thresholds for as follows: $t_{p_{question}} = 6$, $t_{p_{negation-side}} = 6$, $t_{p_{negation-up-down}} = 4$, $t_{p_{affirmation}} = 6$.

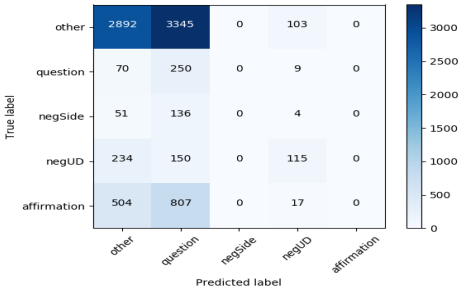
Accuracy values for each user fold that are represented with confusion matrices in Figure 5.14 are calculated as 55 %, 61 %, 31 %, 68 % and 37 %. For the imbalanced test set of SEBEDER video frames, again the macro average of recall values represent the balanced accuracy score of the system. Precision, recall, and f1-scores are calculated only for the predicted class labels to prevent ill-defined values.



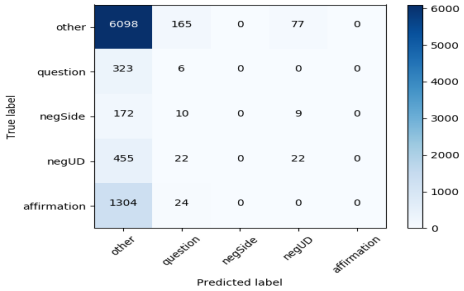
(a) Model 1.



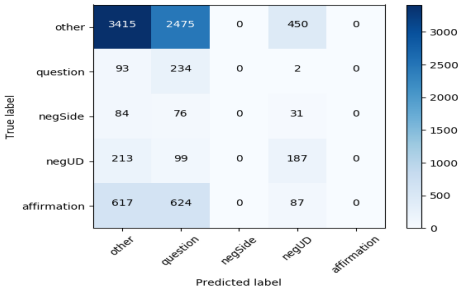
(b) Model 2.



(c) Model 3.



(d) Model 4.



(e) Model 5.

Figure 5.14: Confusion matrices of each test fold in SEBEDER experiment.

As the test set of each partition consists of the same set of video frames in this experiment, we report the average frame-level performance measurements in Table 5.5. The pain class does not exist in SEBEDER videos.

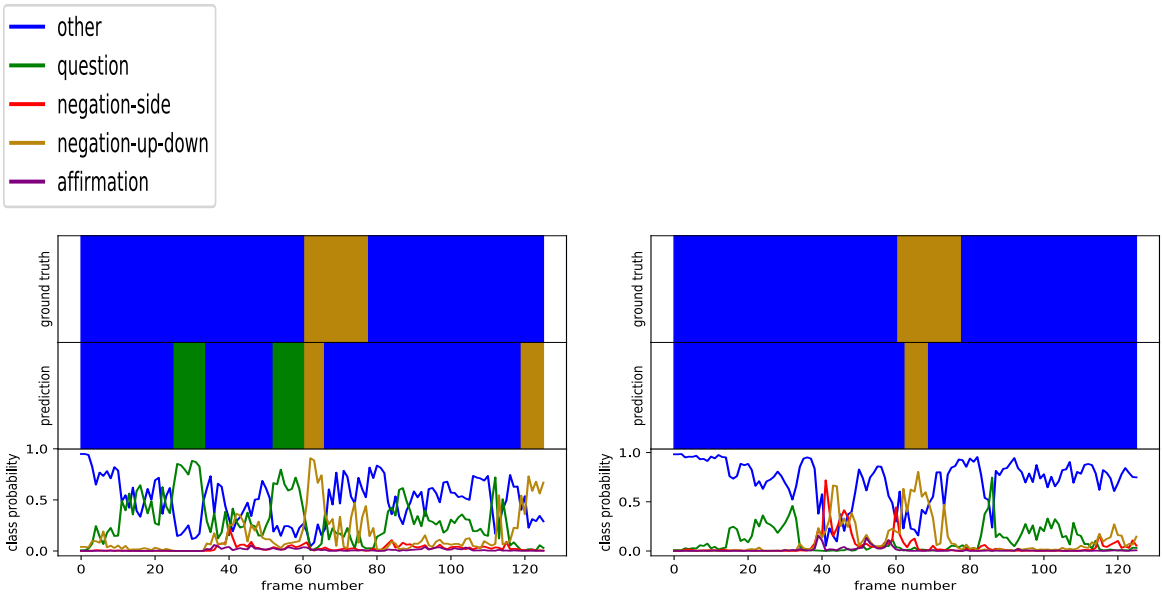
Table 5.5: Frame level average performance measurement for non-manual sign recognition in SEBEDER. Balanced accuracy is denoted with *.

	Class Labels			Averages Without Null Class		
	Other	Negation-up-down	Question	Micro Avg.	Macro Avg.	Weighted Avg.
f1-score	0.72	0.19	0.13	0.14	0.16	0.16
precision	0.75	0.25	0.08	0.11	0.16	0.18
recall	0.72	0.17	0.50	0.30	0.34*	0.30

We observe particularly low performances in this experiment. As is the case in the BosphorusSign Facial Sign dataset experiment, the question class frames have the highest recall value. The system can correctly identify half of the face images with inquiring expressions, which is a promising rate when used as an inquiry retrieval system.Precision value of the question class, however, show that, as the second most populated class (in training phase of the earlier experiment), its probability value dominates the classifier decision and causes the system to misclassify other positive class frames as the question class frame. The patterns of head movements while asking a question and while approving something are somewhat similar indeed. In both cases, the subject slightly tips their head to one side and tilt their head down. These classes can be differentiated better with facial expressions.

As already observed from Table 5.5, we find that the negation-side and affirmation classes are not classified correctly by any of the classifiers. Model 3 and Model 5 recognize the question and negation-up-down class frames slightly better than the other models.

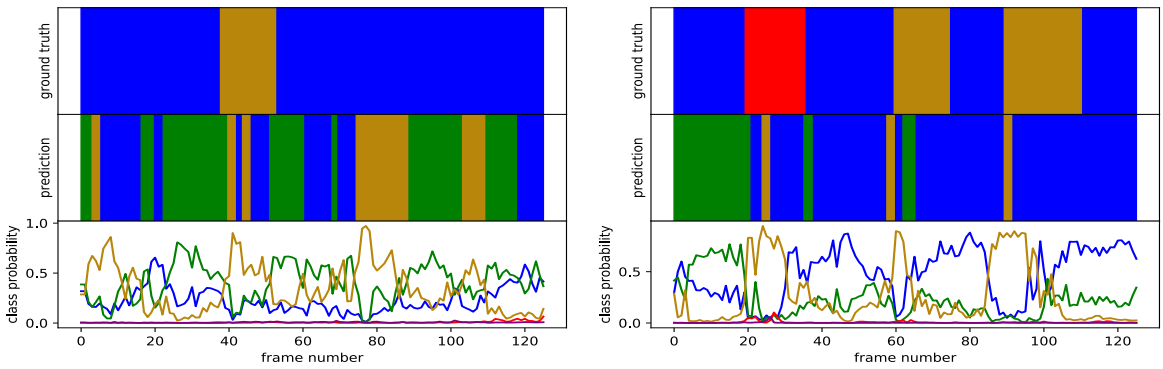
While overall frame-level recognition accuracy is low, we observe that some models are better at avoiding noisy, isolated predictions as seen in Figure 5.15. However, from Figure 5.16, we see that not all of the noisy predictions could be avoided.



(a) Predictions of Model 3.

(b) Predictions of Model 1.

Figure 5.15: Signer performs 'This does not exist, that does not exist', labeled as negation-up-down.

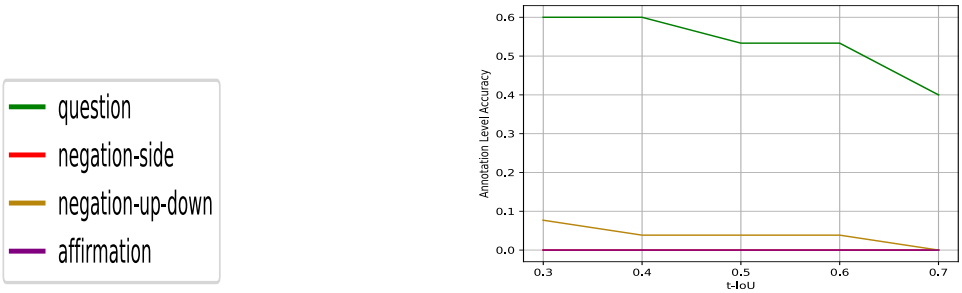


(a) Predictions of Model 5. Signer performs "No, Zafer. Not a leaf is stirring right now.".

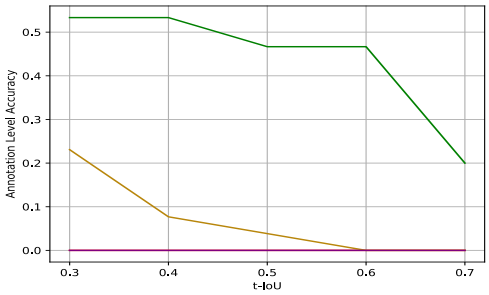
(b) Signer performs "Do not tell anyone, okay?".

Figure 5.16: Prediction probability plots of selected videos from SEBEDER.

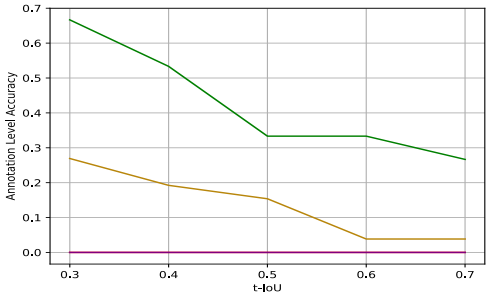
Ground truth annotation block counts of non-manual signs in SEBEDER video clips are given in Table 5.6. Using the same threshold values $t\text{-}IoU = [0.3, 0.7]$ for IoU scores, accuracy values for these 108 annotation blocks are calculated seperately for all five models. The results are given in Figure 5.17.



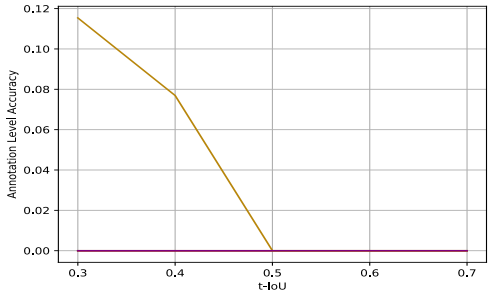
(a) Model 1.



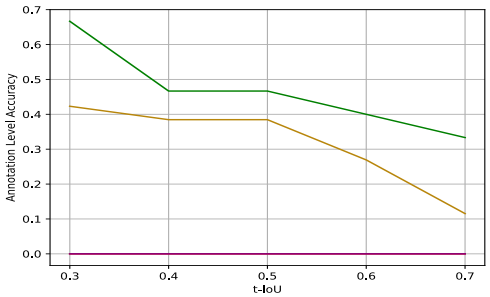
(b) Model 2.



(c) Model 3.



(d) Model 4.



(e) Model 5.

Figure 5.17: Accuracy of ground truth annotations versus different t-IoU values.

Table 5.6: Total number of ground truth annotation tuples per class label in SEBEDER video clips.

	question	negation-side	negation-up-down	affirmation
Test User	15	9	26	58

Similar to the results in Chapter 5.2.1, Model 4 fails to recognize the question annotations. Model 5 gives the most stable accuracy rates with 46% for question annotations and 38 % for negation-up-down annotations at $t\text{-}IoU = 0.5$.

Despite not being exactly comparable, we observe from Figure 5.12f and Figure 5.17e that the Model 5 learns to classify the two out of five class of facial signs: Question and negation-up-down. We find that Model 5 generalizes to unseen data, considering the promising annotation recognition rate on the challenging set of video clips from the SEBEDER dataset.

6. CONCLUSION

In this study, we have prepared a dataset of non-manual signs from Turkish Sign Language and reported recognition results using a baseline classification method. Our non-manual sign dataset has five classes that have significance: Question, negation side to side and negation up-down, affirmation, and pain. We have provided frame-level annotations and developed a baseline system to classify these signs and the null class.

Our classifier relies on a Resnet model that has been pre-trained on a large number of face images. We fine-tune the model and add a final fully connected layer. We also apply post-processing to remove noisy frames. We report frame-level classification results using a leave-one-subject-out protocol. The frame-level precision for the five non-manual sign classes are 58%, 62%, 40%, 68%, and 52%. Since the dataset is highly unbalanced in favor of the "other" class, the balanced accuracy score is calculated as the average recall values of each class, which is 28 %.

The annotation-level classification performance is reported using IoU threshold values. We calculate the annotation accuracy as the rate of correctly classified positive annotation blocks overall 491 positive annotation blocks. We consider an annotation to be correctly classified if its IoU score is above the pre-determined threshold value. For the threshold value $t\text{-}IoU = 0.3$, annotation-level accuracy values are 55.77 %, 14.63 %, 72.83 %, 10 % and 11.67 % for question, negation-side, negation-up-down, pain and affirmation classes respectively. Considering the short duration of actions, we consider setting $t\text{-}IoU = 0.3$ as acceptable. We have also reported the sign recognition performance for each user. It is observed that performance varies from user to user. Some users have better articulations of non-manual signs and the classification accuracy of their signs is higher. Others have non-standard articulations and the misclassification rate is higher. As future work, user adaptation techniques can be applied to remedy this problem.

A cross-database experiment is also conducted to test the generalization capacity of our non-manual sign recognition system on a highly different dataset. TSL translation video of a Turkish movie, in which a professional signs each cue in each movie scene is trimmed into several short clips using a list of negation and affirmation keywords and the movie subtitles. The obtained short clips are then temporally annotated and processed, to extract face images with the non-manual sign labels. Four out of five of the non-manual signs from the first experiment occurred in this test set: Question, affirmation, negation-side, and negation-up-down. Without re-training the model, new video frames are given to the five different models of pre-trained ResNet and the results are reported. The models cannot recognize the affirmation and negation-side frames. However, the most stable results are obtained from one of the models, which classifies 66.67 % of question annotations and 42.31 % of negation-up-down annotations correctly.

Although annotation-level accuracies for selected class labels are promising, we find that the spatial modality of the ResNet is not adequate for modeling the sequential information that lies in sign language videos. As future work, 3D CNNs [12] can be employed for non-manual sign recognition. Alternatively, instead of employing an end-to-end CNN architecture, we may consider feeding the learned features of a CNN to an LSTM as in [13].

From class-specific performance measurements, we find that the pain is one of the most challenging classes to recognize, while the question and negation-up-down classes are more distinguishable. We argue that the facial expressions might be the cause of this issue. An alternative modality for encoding the head movements, such as feeding selected facial landmark coordinates together with the RGB images could increase the classification performance. For ambiguous movements like an approving head shake and a declinatory head nod, eyebrow coordinates would carry crucial information. Similarly, derivative information such as the rotation of landmarks or distance of landmarks from a fixed reference point could be tracked for modeling the facial expression and head movement.

Specifically, in less controlled setups as in the SEBEDER dataset, movements and expressions that occur instantly cannot be easily annotated with the human annotators. Considering the promising recognition rate of two out of five facial sign classes in cross-database experiments, employing our classifier before annotating question and negation-up-down videos may ease the job of the annotator.

To the best of our knowledge, a temporally annotated Turkish Sign Language dataset with non-manual tier labels does not exist. A possible future work would be to annotate other publicly available TSL datasets and use the learned weights on the BosphorusSign-HospiSign subset for a cross-database comparison.

REFERENCES

1. *OpenPose Demo Output-Face Keypoints*, 2019, `https://github.com/CMU-Perceptual-Computing-Lab/openpose`, accessed at April 2019.

2. He, K., X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition", *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

3. *Deafness and Hearing Loss*, 2019, `https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss`, accessed at June 2019.

4. Camgöz, N. C., A. A. Kındıroğlu, S. Karabüklü, M. Kelepir, A. S. Özsoy and L. Akarun, "BosphorusSign: a Turkish sign language recognition corpus in health and finance domains", *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pp. 1383–1388, 2016.

5. Süzgün, M., H. Özdemir, N. Camgöz, A. Kındıroğlu, D. Başaran, C. Togay and L. Akarun, "Hospisign: an interactive sign language platform for hearing impaired", *Journal of Naval Sciences and Engineering*, Vol. 11, No. 3, pp. 75–92, 2015.

6. Cao, Z., G. Hidalgo, T. Simon, S.-E. Wei and Y. Sheikh, "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields", *arXiv preprint arXiv:1812.08008*, 2018.

7. Bobick, A. F. and J. W. Davis, "The recognition of human movement using temporal templates", *IEEE Transactions on Pattern Analysis & Machine Intelligence*, Vol. 23, No. 3, pp. 257–267, 2001.

8. Akyol, S. and P. Alvarado, "Finding relevant image content for mobile sign language recognition", *IASTED International Conference-Signal Processing, Pattern Recognition and Applications (SPPRA), Rhodes*, pp. 48–52, 2001.

9. Cheok, M. J., Z. Omar and M. H. Jaward, "A review of hand gesture and sign language recognition techniques", *International Journal of Machine Learning and Cybernetics*, Vol. 10, No. 1, pp. 131–153, 2019.

10. Krizhevsky, A., I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", *Advances in neural information processing systems*, pp. 1097–1105, 2012.

11. Karpathy, A., G. Toderici, S. Shetty, T. Leung, R. Sukthankar and L. Fei-Fei, "Large-scale video classification with convolutional neural networks", *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1725–1732, 2014.

12. Tran, D., L. Bourdev, R. Fergus, L. Torresani and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks", *Proceedings of the IEEE international conference on computer vision*, pp. 4489–4497, 2015.

13. Yue-Hei Ng, J., M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga and G. Toderici, "Beyond short snippets: Deep networks for video classification", *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4694–4702, 2015.

14. Huang, J., W. Zhou, Q. Zhang, H. Li and W. Li, "Video-based sign language recognition without temporal segmentation", *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

15. Ye, Y., Y. Tian, M. Huenerfauth and J. Liu, "Recognizing American Sign Language Gestures from within Continuous Videos", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 2064–2073, 2018.

16. Zafrulla, Z., H. Brashear, T. Starner, H. Hamilton and P. Presti, "American sign language recognition with the kinect", *Proceedings of the 13th international conference on multimodal interfaces*, pp. 279–286, ACM, 2011.

17. Chuan, C.-H., E. Regina and C. Guardino, "American sign language recognition using leap motion sensor", *2014 13th International Conference on Machine Learning and Applications*, pp. 541–544, IEEE, 2014.

18. Shi, B., A. M. Del Rio, J. Keane, J. Michaux, D. Brentari, G. Shakhnarovich and K. Livescu, "American Sign Language fingerspelling recognition in the wild", *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 145–152, IEEE, 2018.

19. Yang, H.-D. and S.-W. Lee, "Robust sign language recognition by combining manual and non-manual features based on conditional random field and support vector machine", *Pattern Recognition Letters*, Vol. 34, No. 16, pp. 2051–2056, 2013.

20. Aran, O., C. Keskin and L. Akarun, "Sign language tutoring tool", *2005 13th European Signal Processing Conference*, pp. 1–4, IEEE, 2005.

21. Antonakos, E., A. Roussos and S. Zafeiriou, "A survey on mouth modeling and analysis for sign language recognition", *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, Vol. 1, pp. 1–7, IEEE, 2015.

22. Darwin, C. and P. Prodger, *The expression of the emotions in man and animals*, Oxford University Press, USA, 1998.

23. Friesen, E. and P. Ekman, "Facial action coding system: a technique for the measurement of facial movement", *Palo Alto*, Vol. 3, 1978.

24. Ojala, T., M. Pietikäinen and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns", *IEEE Transactions on Pattern Analysis & Machine Intelligence*, Vol. 24, No. 7, pp. 971–987, 2002.

25. Ahonen, T., A. Hadid and M. Pietikainen, "Face description with local binary patterns: Application to face recognition", *IEEE Transactions on Pattern Analysis & Machine Intelligence*, Vol. 28, No. 12, pp. 2037–2041, 2006.

26. Zhao, G. and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions", *IEEE Transactions on Pattern Analysis & Machine Intelligence*, Vol. 29, No. 6, pp. 915–928, 2007.

27. Von Agris, U., M. Knorr and K.-F. Kraiss, "The significance of facial features for automatic sign language recognition", *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*, pp. 1–6, IEEE, 2008.

28. Aran, O., I. Ari, A. Guvensan, H. Haberdar, Z. Kurt, I. Turkmen, A. Uyar and L. Akarun, "A database of non-manual signs in turkish sign language", *2007 IEEE 15th Signal Processing and Communications Applications*, pp. 1–4, IEEE, 2007.

29. Liu, J., B. Liu, S. Zhang, F. Yang, P. Yang, D. N. Metaxas and C. Neidle, "Recognizing eyebrow and periodic head gestures using CRFs for non-manual grammatical marker detection in ASL", *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pp. 1–6, IEEE, 2013.

30. Aran, O., T. Burger, A. Caplier and L. Akarun, "A belief-based sequential fusion approach for fusing manual signs and non-manual signals", *Pattern Recognition*, Vol. 42, No. 5, pp. 812–822, 2009.

31. Crasborn, O. A., J. Mesch, D. Waters, A. Nonhebel, E. Van der Kooij, B. Woll and B. Bergman, "Sharing sign language data online: Experiences from the ECHO project", *International journal of corpus linguistics*, Vol. 12, No. 4, pp. 535–562, 2007.

32. Nguyen, T. D. and S. Ranganath, "Tracking facial features under occlusions and recognizing facial expressions in sign language", *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*, pp. 1–7, IEEE, 2008.

33. Freitas, F. A., S. M. Peres, C. A. Lima and F. V. Barbosa, "Grammatical facial expression recognition in sign language discourse: a study at the syntax level", *Information Systems Frontiers*, Vol. 19, No. 6, pp. 1243–1259, 2017.

34. Walawalkar, D., "Grammatical facial expression recognition using customized deep neural network architecture", *arXiv preprint arXiv:1711.06303*, 2017.

35. *ELAN Video Annotation Tool 5.2*, 2018, `https://tla.mpi.nl/tools/tla-tools/elan/`, accessed at September 2018.

36. Ren, S., K. He, R. Girshick and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks", *Advances in neural information processing systems*, pp. 91–99, 2015.

37. Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge", *International journal of computer vision*, Vol. 115, No. 3, pp. 211–252, 2015.

38. *Torch-NN Loss Functions*, 2019, `https://pytorch.org/docs/stable/index.html`, accessed at June 2019.

39. Masters, D. and C. Luschi, "Revisiting small batch training for deep neural networks", *arXiv preprint arXiv:1804.07612*, 2018.

40. *Assistive Technology and Education Laboratory for Individuals with Visual Disabilities (GETEM)*, 2006.

41. *Audio Description Association (SEBEDER) Film Archive*, 2018, `http://sebeder.org/film-arsivi.php`, accessed at July 2019.

42. Mart Lubbers, F. T., *pympi-ling: a Python module for processing ELANs EAF and Praats TextGrid annotation files.*, 2013.