



School of Information Technology
and Engineering at the
ADA University



School of Engineering
and Applied Science at the
George Washington University

PROBABILISTIC IMAGE PROCESSING AND RECOGNITION MODEL FOR THE
STATIC LETTERS OF AZERBAIJANI SIGN LANGUAGE

A Thesis

Presented to the Graduate Program of Computer Science and Data Analytics
of the School of Information Technology and Engineering
ADA University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Computer Science and Data Analytics
ADA University

By
Aykhan Nazimzada
April, 2022




THESIS ACCEPTANCE

This Thesis by: Aykhan Nazimzada

Entitled: *Probabilistic Image Processing and Recognition Model for the Static Letters of Azerbaijani Sign Language*

has been approved as meeting the requirement for the Degree of Master of Science in Computer Science and Data Analytics of the School of Information Technology and Engineering, ADA University.

Approved:

Dr. Jamaladdin Hasanov (Adviser)		28.04.2022 (Date)
Dr. Abzatdin Adamov (Program Director)		28.04.2022 (Date)
Dr. Sencer Yeralan (Dean)		28.04.2022 (Date)

ABSTRACT

As technology is being improved day by day, accessibility to the different resources is being cleared up. Today, several technologies are aimed to solve problems for disabled people. One of the obstacles is that people from deaf/mute people communities have difficulties creating healthy communication with others, especially those who are not from that community. The technology that solves such problems is known as Sign Language Recognition (SLR) systems.

There are approximately 50,000 deaf/mute people in Azerbaijan. They have their very own sign language called Azerbaijani Sign Language (AzSL). It is inevitable fact that people not from deaf/mute people communities do not know the AzSL except sign language translators. There are 32 letters in Azerbaijani Sign Language. 24 of them are static letters which mean it is interpreted as forming hand parts to a specific orientation. These letters can be illustrated in just one frame. Rest 8 letters are dynamic which are just like static letters, but hands needed to move such as up and down, rotation, or anything else. Those letters cannot be illustrated in a single frame. They are a bunch of frames similar to videos. Other than letters, all words are also dynamic. Our goal in this paper is to come up with an SLR system that reads the video from the live camera and converts it into text in real-time.

AzSL is different than other well-known sign languages such as American, German, French, Russian, and others. Hereby, there is no such dataset that contains letters and words of AzSL. Therefore, the first task was to collect both qualitative and quantitative datasets. For that goal, we created a Telegram bot where volunteer users can capture pictures (for static letters) and videos (for dynamic letters) according to samples provided and can upload them to servers. Users were mostly students of ADA University. In total, approximately 14,000 pictures and 3,000 videos are collected. For further research and applications, data for words that are dynamic is being collected. In this paper, the scope of the aim is to develop a recognition system for static letters only.

For this research, a sufficient number of papers have been read. For our dataset, we detect that using “MediaPipe” for feature extraction is the best option. MediaPipe is an open-source framework that helps users to extract important landmarks from human body parts. In our project, we only use hand landmarks as there is not any effect of human pose or facial emotions in AzSL. It extracts 21 hand joints for a single hand, and each of them has 3 parameters. Hereby, the size of the input becomes 63 (21x3). If both hands are present that number becomes 126 (2x21x3). Another approach was to train raw images in Convolutional Neural Network (CNN) with different parameters. However, because of the few samples and computational power, all experiments with CNN could not reach the desired level of performance. Coming back to MediaPipe features, they are trained in different classifiers including Logistic Regression, Multilayer Perceptrons, Deep Neural Network, and others. There are similar letters that models cannot generalize well. For this reason, 2-level DNNs architecture was designed to train similar letters separately. It can be considered as also clusterization. This architecture gave the best result with 94% of test accuracy.

Keywords: *Sign Language Recognition, Azerbaijani Sign Language, Neural Network, MediaPipe*

TABLE OF CONTENTS

ABSTRACT.....	3
LIST OF FIGURES	5
LIST OF TABLES.....	6
LIST OF ABBREVIATIONS.....	7
1 INTRODUCTION	8
1.1 Definition of the Problem	10
1.2 Objective of the Study.....	11
1.3 Significance of the Problem.....	12
1.4 Review of Significant Research.....	12
1.5 Assumptions and Limitations.....	29
2 RESEARCH APPROACH AND METHODOLOGY	33
2.1 Convolutional Neural Network (CNN).....	33
2.2 MediaPipe + Simple Classification Algorithms.....	34
2.3 MediaPipe + Support Vector Machines (SVM).....	34
2.4 MediaPipe + Deep Neural Network (DNN).....	34
2.5 MediaPipe + 2 -Level Deep Neural Network (Clusterization + DNN).....	36
2.6 Beam Search + Lexicon Verification.....	37
3 RESEARCH RESULTS AND ANALYSIS OF RESULTS.....	40
4 SUMMARY AND CONCLUSION.....	41
REFERENCES	42
APPENDIX.....	44

LIST OF FIGURES

No	Figure Caption	Page
1	Static Letters ‘O’ and ‘U’	10
2	Example of How Sign Language Recognition System Works	11
3	Instance of Dataset “CLAP14 Gesture Spotting”	14
4	Preprocessing	14
5	5 Elements of Sign Language	16
6	A Sensor Based Glove	17
7	Glove with 11 Different Colors	17
8	Example Frames from AUTSL Dataset	18
9	Feature Pooling Module	19
10	Russian Sign Language Dactylology	20
11	MediaPipe Hand Landmarks	21
12	Feature Extraction using MediaPipe	21
13	Augmentation Results	23
14	Sign Language Transformer	25
15	Hand Segmentation	26
16	“Thank You” in Auslan Sign Language	27
17	Contour Detection by Snake Algorithm	28
18	Static Letters ‘L’ and ‘P’	32
19	Static Letters ‘C’ and ‘J’ from front view	32
20	Static Letters ‘C’ and ‘J’ from side view	32
21	Confusion Matrix	35
22	Architecture of Clusterization	37
23	Beam Search Example	38
24	Lexicon Verification Example	39

LIST OF TABLES

No	Figure Caption	Page
1	World's Most Valuable Companies by Market Capitalization	8
2	Percentage of People with and without Disabilities Use Tech Devices	9
3	Parameters of 2 CNN models	15
4	Parameters of ANN model	15
5	Neural Network Architecture	22
6	Results of Russian Sign Language Fingerspelling System	22
7	WLASL Dataset	23
8	Parameters of the Transformer Model	24
9	Effects of the Normalization and Augmentation	24
10	10 Most Frequent Words in Our Lexicon Dataset	30
11	10 Less Frequent Words in Our Lexicon Dataset	31
12	10 Most Probable Letter Given Particular Letter	31
13	Parameters of CNN model	33
14	Simple Classification Algorithms and Accuracies	34
15	Parameters of DNN model	35
16	Clusters and Letters	36

LIST OF ABBREVIATIONS

Abbreviation	Explanation
ANN	Artificial Neural Network
AzSL	Azerbaijani Sign Language
BLSTM	Bidirectional Long Short-Term Memory
CNN	Convolutional Neural Network
CSV	Comma-Separated Values
CV	Computer Vision
DNN	Deep Neural Network
FF	Feed Forward
FFT	Fast Fourier Transform
FPM	Feature Pooling Module
HMM	Hidden Markov Model
HSV	Hue, Saturation, Value
LSTM	Long Short-Term Memory
ML	Machine Learning
MLP	Multilayer Perceptron
NN	Neural Network
OCR	Optical Character Recognition
PE	Positional Encoding
RGB	Red, Green, Blue
SE	Spatial Embedding
SLR	Sign Language Recognition
SVC	Support Vector Classifier
SVM	Support Vector Machine
ViT	Vision Transformer
WE	Word Embedding

1 INTRODUCTION

We are living in the era of technology. It is hard to imagine any project or business that is not benefitting from the assets of the technology. Almost all companies or organizations have at least a website or social media. These assets are meant to increase productivity while decreasing cost. According to statistics of February 2022, 8 out of 11 most valuable companies are purely technology companies that develop software or manufacture hardware [1]. Remaining 3 of them are also strongly correlated with technology.

<i>Rank</i>	<i>Company</i>	<i>Market Capitalization</i>
1	Apple	\$2.8 trillion
2	Microsoft	\$2.2 trillion
3	Aramco	\$2.0 trillion
4	Alphabet	\$1.8 trillion
5	Amazon	\$1.6 trillion
6	Tesla	\$905.7 billion
7	Berkshire Hathaway	\$700.6 billion
8	Nvidia	\$613.0 billion
9	TSMC	\$600.3 billion
10	Tencent	\$589.8 billion
11	Meta	\$565.4 billion

Table 1. World’s Most Valuable Companies by Market Capitalization

Beyond the productivity, technology also makes resources accessible for everyone. Accessibility is a vital aspect in the technology which aims to spread the audience as much as possible. Accessibility has 2 main aspect. First one is about the accessing certain product regardless of used device, network, or location. This is important issue to be considered especially in software development. Latter one is making technological product accessible by people with disabilities such as vision impairment, memory loss and etc. In the rest of the paper, with saying accessibility, we mean making something accessible by people with disabilities. According to WHO (World Health Organization), about 15% of the world population (over one billion people) have some type of disability [2].

According to recent research held by “Pew Research Center”, Americans with disabilities are less likely to use technological devices [3]. In the Table 2, you can find that people without disabilities are more likely to use every kind of technological devices. This is because people with disabilities are not comfortable with using technology as not all of the contents are accessible by them.

<i>Use of Technology</i>	<i>Any Disability</i>	<i>No Disability</i>
Desktop or Laptop	62	81
Smartphone	72	88
Tablet	47	54
Home Broadband	72	78
All of the above	26	44

Table 2. Percentage of People with and without Disabilities Use Tech Devices

Machine Learning algorithms and techniques are in favor of accessibility. Machine Learning is a type of Artificial Intelligence which is designed in order to predict outcome according to given data without being explicitly programmed. Giant tech companies successfully exploit the power of Machine Learning and kick-off social projects that ease the life of disabled people. Now, we will cover several social apps and projects to form clear understanding of applications of Machine Learning in aspect of accessibility.

“Lookout” is an application developed by Google targeting people with vision impairments [4]. It is a mobile app providing video and image captioning to the media recorded with the camera. It also provides OCR (Optical Character Recognition) of the text which is hard to read for people with vision disabilities.

Another example is an app called “RogerVoice” [5]. It helps the people with hearing disabilities to have a phone call. Simply, it is speech to text system. Received signals from other end converted to the text in a real time that user can read it and answer.

Final example is an app developed by Samsung called “Wemogee” [6]. Aphasia is a disorder that causes the loss of language capabilities. The role of “Wemogee” is to transform text in English or Italian into non-linguistic form using images and emojis. Hereby, people with language disabilities can understand it.

Our paper consists of the research and implementations of Azerbaijani Sign Language Recognition system which aims to destroy the boundaries for mute/deaf people in communication with people who do not interpret the sign language.

1.1 Definition of the Problem

According to statistics of “Ministry of Labor and Social Protection of Azerbaijan”, there are 49,526 deaf/mute people in Azerbaijan. In their daily life, they use Azerbaijani Sign Language (AzSL) to express their feelings.

AzSL is unique sign language. It is different from worldwide used sign languages such as American, German, Russian and others. AzSL contains 32 letters. 24 of them are static signs which means they are interpreted as a static formation of hand position. There is no movement at all. And the remaining 8 letters are dynamic. For these letters, we make some moves with our hands. For better understanding, let’s look at the examples. In the below image, you can find the static letter ‘O’ and ‘U’ from AzSL correspondingly.

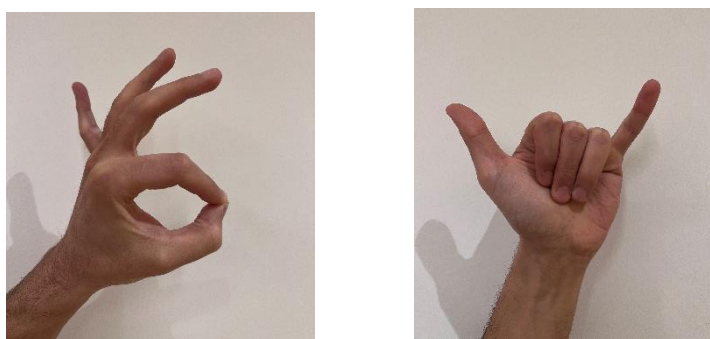


Fig 1. Static Letters ‘O’ and ‘U’

Dynamic letters ‘Ö’ and ‘Ü’ are interpreted as 2 above static letters ‘O’ and ‘U’ with only difference. Hand is moved from top to bottom for a second while keeping the fingers’ positions the same. Hereby, similar letters such as ‘O’ and ‘Ö’ are differentiated.

Beyond the letters, each word is interpreted with special dynamic movements of the one or two hands. In total, there are 24 dynamic, 8 static letters, and lots of dynamic words.

Problem is that when deaf/mute individual wants to speak with person without disability or someone who does not know the AzSL, it creates obstacle. This case occurs in social life, in work life, or when individual wants to get service from governmental organization. Current solutions to these kinds of problems are whether holding communication using pen and paper or via the help of sign language translators. In the former case, there are too much labor stuff. If the conversation is long enough, writing it will be tiring for both side. In the latter case, finding sufficient number of sign language translator will be impossible and the cost will be high.

1.2 Objective of the Study

As technology and Machine Learning improved enough, why not to come up with a solution that uses it which can be accessible by anyone in anywhere. The objective of the study is to use Computer Vision techniques. Computer Vision is a subfield of Machine Learning which lets computers predict via deriving useful information from digital images and videos.

Machine Learning is developed so much that it makes accomplishment of such a goal to be realistic. With the power of Neural Networks, similar systems for different sign languages such as American, German, and others are already developed. Some of them are still in development. Objective is to build such system in an efficient way without using any special hardware which will make recognition system accessible by everyone.

Today, powerful Computer Vision libraries such as OpenCV, MediaPipe, SimpleCV, TensorFlow, Keres, PyTorch and much more makes the job easier.



Fig 2. Example of How Sign Language Recognition System Works

Aim is to have a system that converts interpretations of deaf/mute people into written language, in other words, a system that converts from AzSL to text in Azerbaijani. In this paper, focus is the recognition and translation of the static letters only. In the upcoming sections, I will talk about previous works, research I have made, methodology, results, and more.

1.3 Significance of the Problem

As mentioned earlier, there are approximately 50,000 deaf/mute people in Azerbaijan. In communication with people who do not know Azerbaijani Sign Language, they are facing with problem. Recognition system that records motion using camera and translating to written text in real time will break down that barrier.

Hundreds of deaf/mute people go to governmental organization to acquire service. In order to have healthy communication, translators are employed. Nevertheless, there is shortage in the number of Azerbaijani Sign Language translators.

Our stakeholder in this project is “Ministry of Labor and Social Protection of Azerbaijan”. They acknowledge the importance of such system in every aspect. Furthermore, such system will result in better service to the citizens. It can also be integrated to the other ministries and governmental organization.

Moreover, it can be expanded to the businesses such as restaurants and hotels for increasing the customer satisfaction. Mobile applications at the end even can influence deaf/mute community’s daily life in every aspect.

1.4 Review of Significant Research

In this section, we will review some written papers and research about SLR systems.

Helen Cooper, Brian Holt & Richard Bowden give brief introduction about sign language recognition and its motives in one of the chapters of their book “Visual Analysis of Humans” [7]. Chapter begins with motivation behind the building Sign Language Recognition system. Currently almost all translation services held by human translators. This results in increased cost and limited services as they are much more deaf/mute people than human sign language translators. They also claim that SLR systems shouldn’t be supposed as gesture recognition system. SLR is much more complex which requires suitable data and optimal model. It is also mentioned that all publicly available datasets are limited and not good in both quantity and quality level. Another decisions need to be made are feature extraction techniques and classification models.

Signs in the sign languages are divided in up to 3 parts. First one is ‘Manual Features’. Manual Features are interpreted with positioning and movements of the hands. Second type is ‘Non-Manual Features’. Those are the signs which are based on body and face gestures (Non-Manual Features are not included in the Azerbaijani Sign Language). Last one is ‘Finger-Spelling’. Some words or special names including person name, city or others may not be included in the sign language dictionaries. In this case, those words are interpreted with illustrating each letter of word separately which is a challenge. Another challenge mentioned is co-articulation. It generally occurs when meaning of the sign is affected by the preceding or following sign. Authors resembles SLR systems with speech recognition system because of co-articulation problems.

In the next section, authors talk about the conventional feature extraction methods. Mostly used one is using sensor-based gloves. These gloves extract most important features that can be used in training of the classification algorithms. Although these sensor-based gloves give sufficient features, they also bring some mobility problems. In case of complex signs, it can fail to extract optimal features. Biggest issue is accessibility. These gloves are not only needed in the process of data collection for training. For further usage and testing, individual needs to wear gloves and perform the signs. Features will be extracted and the Machine Learning algorithm will make predictions according to those features. Hereby, for every application of SLR system, these sensor-based gloves needed to be used which becomes costly. Not everyone is able to afford it. Another such device used for the purpose is Microsoft Kinect. It is a tracking device developed by Microsoft which was mostly used in gaming industry. Datasets and applications of the outputs of the Kinect devices are limited. After a while, Microsoft stopped production of Kinect devices and its support. Later on, these features are trained in different models, such as Neural Network, Hidden Markov Model, and others.

In the paper “Sign Language Recognition Using Convolutional Neural Networks”, authors **Lionel Pigou, Sander Dieleman, Pieter-Jan Kindermans, and Benjamin Schrauwen** talk about the building Sign Language Recognition System using Convolutional Neural Networks [8]. In general, they divide the task into 2 parts; Feature Extraction and Classification. For them, from each frame sequence, features need to be extracted. These will result in one or more feature vectors. For this purpose, they use Convolutional Neural Networks. Then, each feature vector is classified to the corresponding sign or gesture. This is where the classification occurs. Here, they use Artificial Neural Networks.

For the data, “Track 3: Gesture Spotting” data from the subset of dataset “ChaLearn Looking at People 2014” (CLAP14) are used [9]. This dataset contains 20 different gestures from Italian Sign Languages recorded by 27 people. They have used 6600 samples during training (4600 for training, 200 for validation). And 3543 samples are used as a holdout set. It’s worth to mention that videos are recorded with Microsoft Kinect which means depth map, user index and another filters of the frames are accessible. In the figures below, you can find depth map, user index and joint positions for a particular frame.

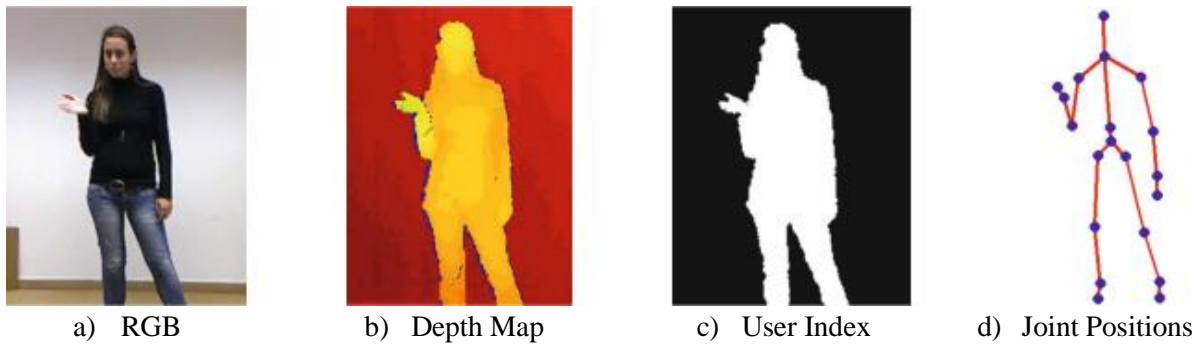


Fig 3. Instance of Dataset “CLAP14 Gesture Spotting”

As a preprocessing, they have cropped the hand parts from the frames, did the grayscaleing to the depth maps, noise reduction and applied Median Filters.

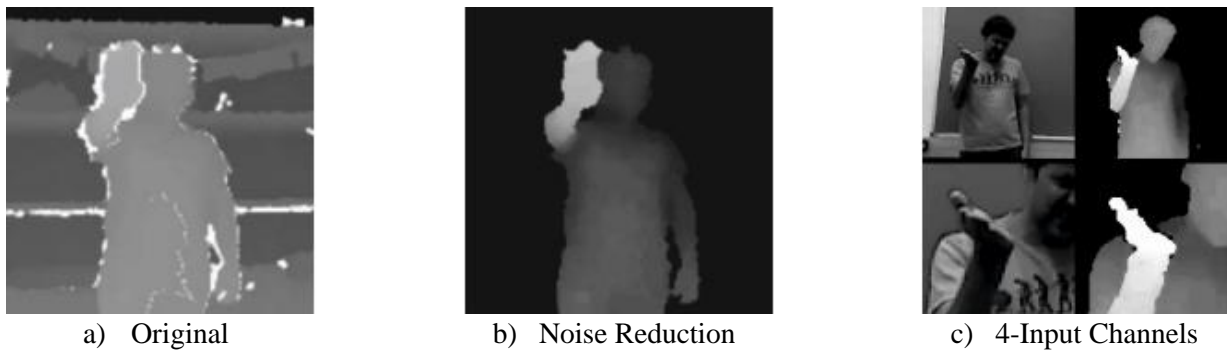


Fig 4. Preprocessing

The model they proposed consists of 2 CNNs and 1 ANN. First CNN is responsible for extracting features for hand and another is for upper parts of the body. Both CNNs are 3 layers with same parameters. Those are simple CNNs with Convolutions and Max Pooling layers. In the last part of the CNN, output is flattened into 1D array. For the ANN part, outcomes of both CNNs are concatenated. Finally, 2 Dense layers are followed with units of 512 and 20 which is number of targets to be classified. In the tables below, you can find the information about CNN models and ANN model.

<i>Parameters</i>	<i>CNN 1 & CNN 2</i>
<i>Input Layer</i>	input_shape = (64, 64, 32)
<i>Layer 1a) Convolution Layer</i>	filters = 16 kernel_size = (5, 5)
<i>Layer 1b) Max Pooling</i>	pool_size = (2, 2, 2)
<i>Layer 2a) Convolution Layer</i>	filters = 32 kernel_size = (5, 5)
<i>Layer 2b) Max Pooling</i>	pool_size = (2, 2, 2)
<i>Layer 3a) Convolution Layer</i>	filters = 48 kernel_size = (4, 4)
<i>Layer 3b) Max Pooling</i>	pool_size = (2, 2, 2)
<i>Layer 3c) Flatten</i>	

Table 3. Parameters of 2 CNN models

<i>Parameters</i>	<i>ANN</i>
<i>Input Layer</i>	Concatenation of outputs of 2 CNN models
<i>Layer 1b) Max Pooling</i>	units = 512
<i>Layer 2b) Max Pooling</i>	units = 20

Table 4. Parameters of ANN model

Overall accuracy of the best model on the validation set was 91.7% with 8.3% error rate where accuracy on the holdout set was 95.7% with error rate of 4.3%.

Paper “A Review on Systems-Based Sensory Gloves for Sign Language Recognition State of the Art between 2007 and 2017” written by **Mohamed Aktham Ahmed, Bilal Bahaa Zaidan, Aws Alaa Zaidan, Mahmood Maher Salih, and Muhammad Modi bin Lakulu** talks about the methods for building Sign Language Recognition Systems using sensor-based gloves for extracting important features during the period of 2007 and 2017 [10]. For authors, Sign Language is positional and visual component which has special meaning in it. They claim Sign Language is about 5 components which defines it. Those 5 block represent important values which should not be ignored while developing such recognition system. In the figure below, you can find those 5 elements of Sign Language.

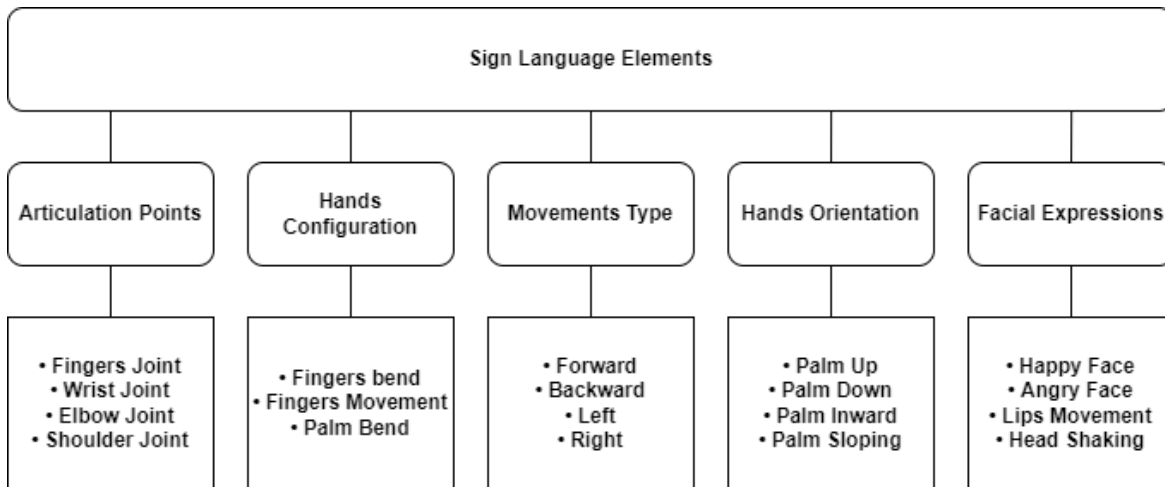


Fig 5. 5 Elements of Sign Language

There are 3 main types of methodology of Sign Language Recognition System which are “Sensor Based”, “Vision Based”, and “Hybrid”.

In Vision-based systems, primary tool is camera. Input is retrieved using cameras which is quite affordable. Almost all laptops and smartphones have cameras nowadays. In comparison to sensor-based systems, costs are low. However, it has also several downsides. Each cameras may have different specializations such as image depth and field of view which is one of challenges. Main problem is extracting features from frames in vision-based systems are difficult and computationally costly. In this paper, different feature extraction methods such as Combined Orientation Histogram, Statistical (COHST) Features, and Wavelet Features on different Sign Languages are tested. Best results retrieved by using Wavelet Features in recognition of static signs of numbers from 0-9 in American Sign Language. Accuracy is 98.17%. With same feature extraction method and Neural Networks, accuracy of recognizing 32 Persian alphabets is 94.06%.

In Sensor-based systems, the challenge of feature extraction is done by the specific hardwares. Most widely spread approach is use of gloves using sensors such as flexion sensors, accelerometers, proximity sensors, and abduction sensors. Such sensors measures angles between fingers, orientation of wrist, abduction between fingers, and more.

Hybrid systems use both Vision-based and Sensor-based systems. In other words, both gloves and cameraas are used in order to build consistent recognition system. This approach is not widely used in researchs and developments due to its high cost.

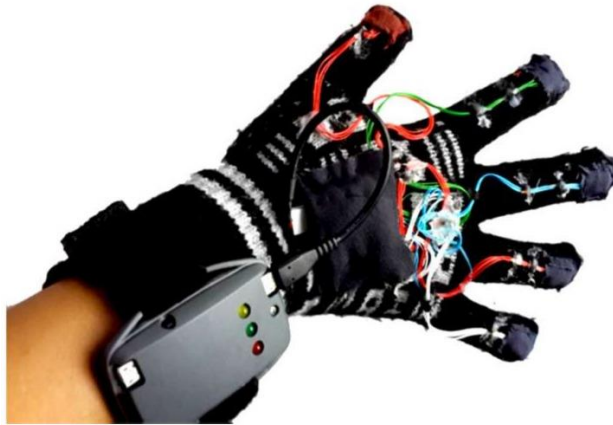


Fig 6. A Sensor Based Glove

In the rest of the paper “A Review on Systems-Based Sensory Gloves for Sign Language Recognition State of the Art between 2007 and 2017”, authors deep dive in sensors in an engineering aspect which is not in the scope of our research.

The paper “American Sign Language Alphabet Recognition Using Microsoft Kinect” written by **Cao Dong, Ming C. Leu, and Zhaozheng Yin** is about building SLR system using another sensor-based device called Microsoft Kinect [11]. In training, they use the depth images retrieved from publicly available dataset from Surrey University taken by Microsoft Kinect [12]. Before capturing the images with Kinect device, users wear specially designed gloves which is colored with 11 different color. Reason of using such glove is that they aim to classify hand regions. Then, depth images taken with Kinect while wearing those colored gloves are transformed from RGB (red, green, blue) space into HSV (hue, saturation, value) space.











Color glove	Raw color image	Segment ground truth	Raw depth image
			
			
			

Fig 7. Glove with 11 Different Colors

Next step is feature extraction. For each depth image's (I) each pixel (x), the value of pixel is subtracted from the value of offset pixel ($x + v_n$).

$$f_n(I, x) = I(x + v_n) - I(x)$$

Those pixel features are used to classify hand regions into 11 classes. Authors call this classification step as “Per-pixel Classifier”. For this purpose, they have used simple Random Forest Classifier.

Using only joint coordinates in gesture recognition was not successful. Therefore, authors propose 2 new techniques that will increase the robustness of the recognition system. There are Joint Localization and Kinematic Constraints. What returned from per-pixel classifier is probability distribution just like softmax layer. We have 11 probabilities which each of them refers to single hand region. For representing joint coordinates, global mass center of the probability distribution map is not what they seek. They used new mean algorithm called mean-shift local mode-seeking which gave better results

Above mentioned algorithm sometimes fails in localizing joint positions as pixels in neighborhood hand regions can be misclassified. Authors propose using kinematic constraining method to handle such errors in system. If the pixels cannot fit the kinematic structure of the hand, weight of that pixel is penalized.

Several experiments are held to evaluate the system with different parameters. Best result was 90% accuracy.

Ozge Mercanoglu Sincan and **Hacer Yalim Keles** from Ankara university wrote a paper called “AUTSL: A Large Scale Multi-modal Turkish Sign Language Dataset and Baseline Methods” which covers building Turkish Sign Language recognition system using deep learning techniques such as CNN, LSTM, BLSTM, feature pooling, and attention model [13]. They present Turkish Sign Language dataset consists of RGB, depth, and skeleton versions which captured by Kinect. The dataset has 226 signs with 38,000 samples. 43 signers are involved in data collection where videos are taken in 20 different backgrounds.



Fig 8. Example Frames from AUTSL Dataset

The CNN model they use is VGG16 from ImageNet. As you may know, it is well-known feature extraction model which is pre-trained on over 14 million images belonging to more than 1,000 classes. For sure, they do some modifications to the layers of VGG16 model. Except last max pooling layer, all the layers are used. Large networks have several convolutional layers. First layers try to analyze and extract edges, shapes, and etc. In other words, it focuses on low-level. Last convolutional layers are specialized in objects, colors, and more. For this reason, only last 2 convolutional layers are modified. Also, frames are resized to 256 x 256. Hereby, input of the CNN model is 256 x 256 x 3 where output of last convolutional layer is 16 x 16 x 512.

FPM (Feature Pooling Module) is an effective technique while extracting features from multi scale data. It is also used in isolated sign language recognition systems. In a simple manner, FPM is bunch of convolution layers running in parallel with different dilation rates. In this paper, authors mention 4 convolutional layers running in parallel. Their results are concatenated at the end of Feature Pooling Module. Each of 4 CNN models output 128 features which result in 512 features at the end.

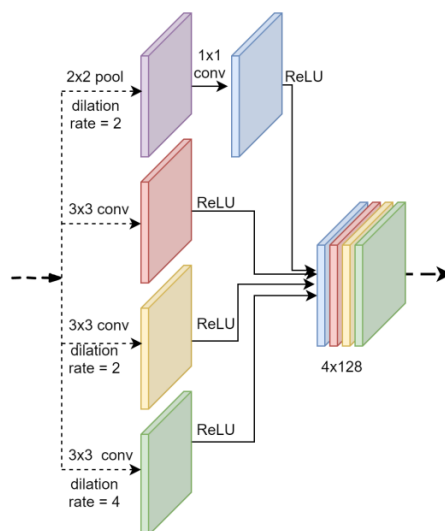


Fig 9. Feature Pooling Module

LSTM (Long Short-Term Memory) is a well-known technique in deep learning which is used to capture temporal relationships. As Sign Language is a dynamic language meaning signs are interpreted using hand movements. Bidirectional LSTM is special case of LSTM. LSTM focuses on the sequences from beginning to the end. It is also called forward-pass. On the other hand, Bidirectional LSTM focuses on backward. Simply, it is 2 LSTM model running in parallel in opposite directions. In this paper, bidirectional LSTM is used. Both LSTM models have hidden unit sizes of 512. Hence, bidirectional LSTM has hidden unit size of 1024. For each hidden unit, result of forward LSTM and backward LSTM are summed up. Attention models are also integrated to the LSTM and bidirectional LSTM models in order to choose most important features.

Experiments are held in different combinations of models containing CNN, LSTM, BLSTM, FPM, and Attention model. Best model is CNN + FPM + BLSTM + Attention with 95.46% accuracy.

Speaking of LSTM models, I will review a paper “Development of a software module for recognizing the fingerspelling of the Russian Sign Language based on LSTM” written by **M. G. Grif** and **Y. K. Kondratenko** [14]. Fingerspelling or Dactylogy is used when there is no special sign for a particular word in the sign language. Fingerspelling is mostly used for interpreting the names, places, and etc. In this paper, authors talk about the methodologies for building recognition system for fingerspelling in Russian Sign Language. In Russian Language, there are 33 letters. Data they have used collected researchers from Novosibirsk State Technical University. Dataset contains approximately 15,000 photos and videos for 33 letters.

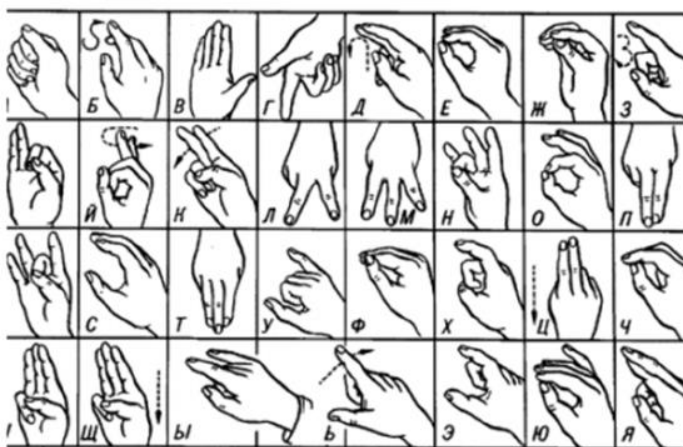


Fig 10. Russian Sign Language Dactylogy

For feature extraction, they use MediaPipe framework which is available in Python. It is mostly used for extracting important features of the human body. For sign language, most important features would be surely hands. For SLR systems, only hand landmarks of the MediaPipe are being used. For hand landmarks, MediaPipe detects 21 most important features for each hand. Simply, they are joint parts of the hand. For each feature, there are 3 coordinates; x, y, and z. In short. For each hand, there will be 63 features (21 x 3). If there are 2 hands, number of features will be 126 (63 x 2). Those features can be used in different classifiers later on. Power of MediaPipe is that it does not require any device or any kind of hardware. It is an open-source framework developed by Google.

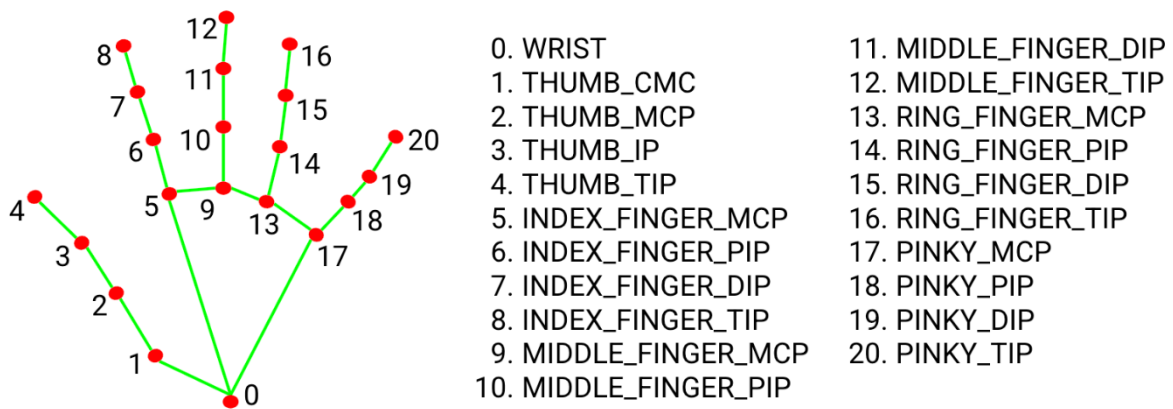


Fig 11. MediaPipe Hand Landmarks

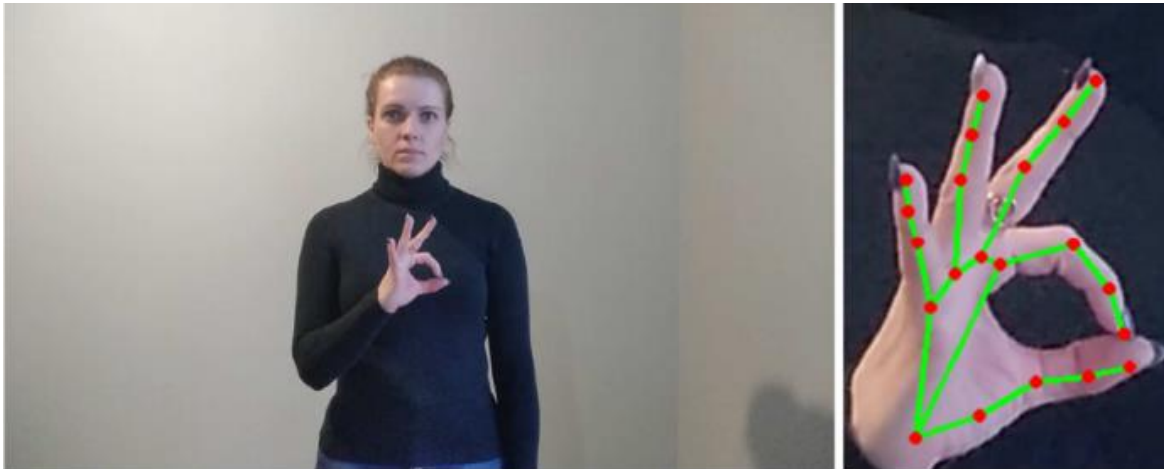


Fig 12. Feature Extraction using MediaPipe

The model authors proposed is multi-class neural network. It is consist of 6 layers. First 3 layers are Bidirectional LSTM. Those 3 layers detects dynamic gestures via exploiting relationships between frames. 3 Bidirectional LSTM layers have 1024, 512, and 256 neurons correspondingly. The last 3 layers are simple DNN layers which are fully connected. The first 2 layers have “ReLu” activation function with 128 and 64 neurons. Last fully connected layer is softmax layer with 33 neurons which corresponds to number of letters in Russian sign language.

<i>Layers</i>	<i>Input Size</i>	<i>Output Size</i>
<i>1st BLSTM Layer</i>	(None, 100, 63)	(None, 100, 1024)
<i>2nd BLSTM Layer</i>	(None, 100, 1024)	(None, 100, 512)
<i>3rd BLSTM Layer</i>	(None, 100, 512)	(None, 100, 256)
<i>1st Dense Layer</i>	(None, 256)	(None, 128)
<i>2nd Dense Layer</i>	(None, 128)	(None, 64)
<i>3rd Dense Layer</i>	(None, 64)	(None, 33)

Table 5. Neural Network Architecture

The network is trained 20 epochs. Overall accuracy of the system was 91%. Value of the loss function was 0.21. In the table below, you can find the detailed measures such as precision, recall, and F1-measure for 16 letters.

<i>Sign</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Measure</i>
а	0.72	0.82	0.77
б	0.82	0.80	0.81
в	1.00	0.98	0.99
г	1.00	0.97	0.99
д	0.50	0.29	0.36
е	0.88	0.83	0.85
ё	0.19	0.25	0.21
ж	0.89	0.98	0.93
з	0.29	0.71	0.42
и	0.83	0.91	0.87
й	0.33	0.43	0.38
к	0.61	0.69	0.65
л	0.82	0.89	0.86
м	0.94	0.91	0.92
н	0.95	0.95	0.95
о	0.96	0.95	0.96
Overall	0.91	0.90	0.91

Table 6. Results of Russian Sign Language Fingerspelling System

Results shows that dynamic gestures such as “д”, “ё”, “з”, and “й” have accuracy less than 50% accuracy. As they are trained in LSTM layers, more data are needed for better performance.

In the paper “Sign Pose-based Transformer for Word-level Sign Language Recognition” written by **Matyas Bohacek** and **Marek Hruz** is about a system for word-level sign language recognition based on the Transformer model [15]. They have used 2 datasets to test their model. First one is Word Level American Sign Language (WLASL) [16]. The dataset is collected by native American Sign Language interpreter which is used for teaching purposes. WLASL contains 4 subsets which are WLASL100, WLASL300, WLASL1000, and WLASL2000. The detailed information about datasets can be found at the table below. 2nd dataset is LSA64 (Argentinian Sign Language dataset) which has 64 classes and 3200 videos [17].

<i>Subset</i>	<i>Classes</i>	<i>Videos</i>	<i>Mean</i>	<i>Signers</i>
WLASL100	100	2,038	20.4	97
WLASL300	300	5,117	17.1	109
WLASL1000	1000	13,168	13.2	116
WLASL2000	2000	21,083	10.5	119

Table 7. WLASL Dataset

For feature extraction, they have used Vision API which is used for extracting head, body, and hand landmarks where authors claim that any other pose landmark extraction tools can work in this project. In total, they retrieve 54 body landmark. 5 of them are head landmarks and 21 of them are hand landmarks.

Additionally, in order to increase number of samples, augmentation techniques are also applied. Some of them are in-plane rotation, squeezing, perspective transformation, and sequential joint rotation. In Fig. 13, 4 different augmentation results of a single image are illustrated.

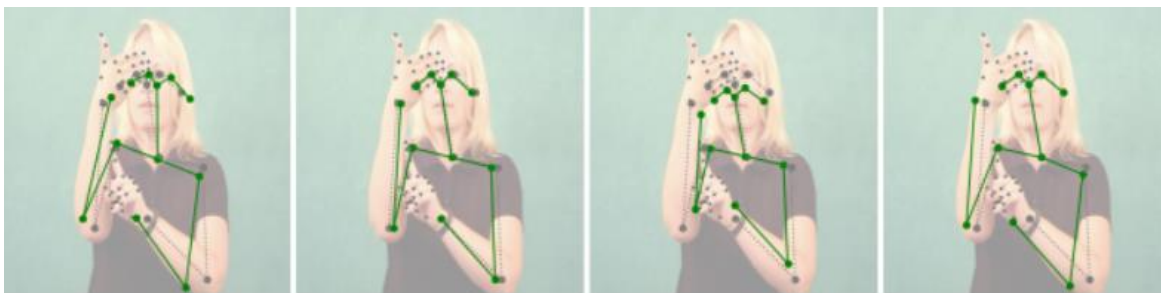


Fig 13. Augmentation Results (In-Plane Rotation, Squeezing, Perspective Transformation, Sequential Joint Rotation)

Moreover, as videos and images are captured from different perspectives and field of views, several normalization techniques are applied. Hereby, model would converge faster as the features are in close scale. After normalization, performance and accuracy would increase.

The model is modified Transformer. As discussed above, normalized 54 landmark features are used as vectors. For each landmark, there are 2 features. In total, input size becomes 108. For each vectors, there are positional encoders. Self-attention model contains 6 encoder layers and 9 heads. Input of the transformer's decoder is a single query. After query is decoded, it passes through multi head projection module. Last layer of the attention model which is softmax layer gives the result.

Encoder Layer	Decoder Layer	Heads	Hidden Dim.	Feed Forward Dim.	Input Dim.
6	6	9	108	2048	108

Table 8. Parameters of the Transformer Model

Accuracy of the model on the WLASL100 dataset is 63.18%. After increasing the number of classes, accuracy on the WLASL300 dataset becomes 43.78%. If it was tested on the datasets WLASL1000 and WLASL2000, accuracy would decrease more.

Authors also did different experiments in order to test effects of the normalization and augmentation. 8 different models are implemented and evaluated on the WLASL100 dataset. Those 8 models and their accuracies are shown in the Table 9.

<i>Model</i>	<i>Normalization</i>	<i>Augmentation</i>	<i>Accuracy</i>
<i>A</i>	X	X	45%
<i>B</i>	✓	X	59%
<i>C</i>	✓	In-Plane Rotation	61%
<i>D</i>	✓	Squeezing	61%
<i>E</i>	✓	Perspective Transformation	61%
<i>F</i>	✓	Sequential Joint Rotation	61%
<i>G</i>	✓	All	62%
<i>H</i>	✓	All + Gaussian Noise	63%

Table 9. Effects of the Normalization and Augmentation

According to the table above, most important asset is normalization. It increases the accuracy of the model significantly where augmentation techniques have less impact on the performance of the model. The reason of this difference is that regardless of augmentation techniques, results of feature extraction methods are almost identical. If the model was Convolutional Neural Network, most probably augmentation techniques applied here would be beneficial.

Authors **Necati Cihan Camgoz**, **Oscar Koller**, **Simon Hadfield** and **Richard Bowden** introduce new video transformer-based architecture for Sign Language Recognition in their book “Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation” [18]. Video Transformers read the sign language videos and convert it into sign language glosses. It is a sequence-to-sequence model hereby, most important assets are conditional probabilities. They divide the task of sign language translation into 2 groups. The first approach is to consider Continuous Sign Language Recognition as text-to-text translation. Hereby, model maps the sign glosses with its corresponding text. The second set of approaches focuses on translation from sign video representations to spoken language with no prior knowledge of the sign language. With providing a sufficient number of data and building strong model solves such problem.

The authors propose a unified model which they call Sign Language Transformer. This model is trained in order to generate spoken languages from sign language representations.

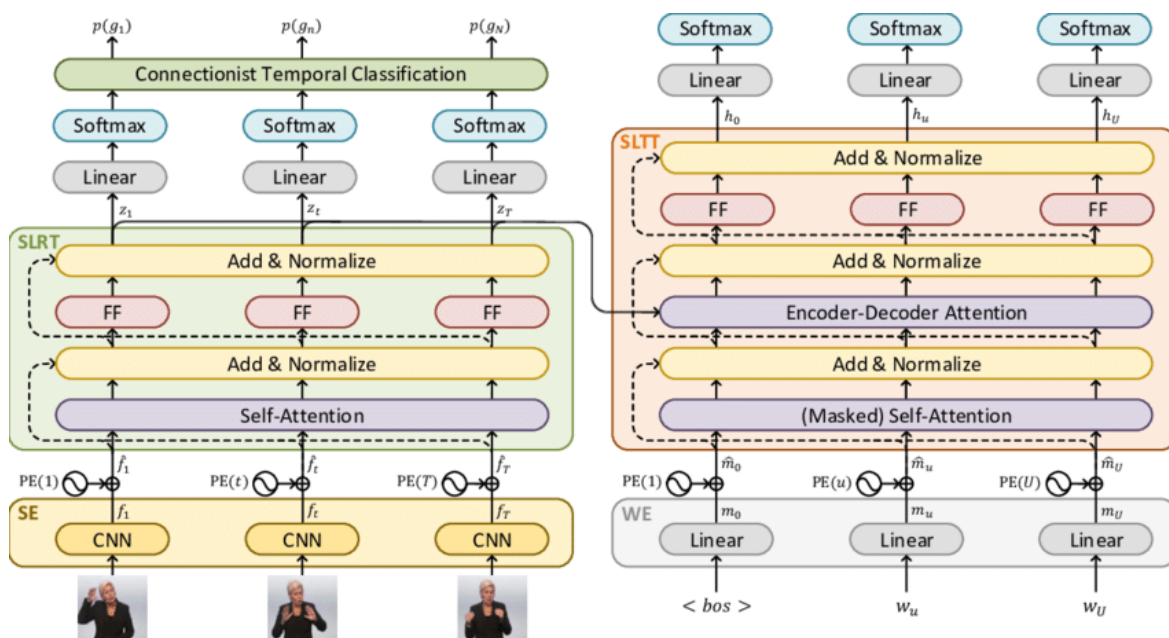


Fig 14. Sign Language Transformer (SE: Spatial Embedding, WE: Word Embedding , PE: Positional Encoding, FF: Feed Forward)

The model trained and tested in RWTH-PHOENIX-Weather-2014T dataset where annotations are in German [19]. BLEU Score for their best model is 47.26

The paper “Dataset of Pakistan Sign Language and Automatic Recognition of Hand Configuration of Urdu Alphabet through Machine Learning” written by **Ali Imrana, Abdul Razzaqa, Irfan Ahmad Baig, Aamir Hussaina, Sharaiz Shahida, and Tausif-ur Rehmana** is about how the Pakistan Sign Language dataset is collected and how it is recognized by Support Vector Machine [20]. Dataset contains 1480 images for 37 letters of alphabet. Proposed recognition system is consist of 3 parts; Segmentation, Detection, and Sign Recognition.

Segmentation is simply keeping the hand in the image while removing the background. After successful segmentation, hand parts in the image becomes white and background becomes black. Task here is to convert colorful image into black and white image. However, finding best threshold is challenging. For this purpose, they decided to work on HSV space images rather than RGB space. HSV space is consist of 3 components; Hue, Saturation, and Value. Hue represents the dominant color where saturation is the purity of color. From the hue and saturation components, hand parts can be easily isolated using fixed threshold.

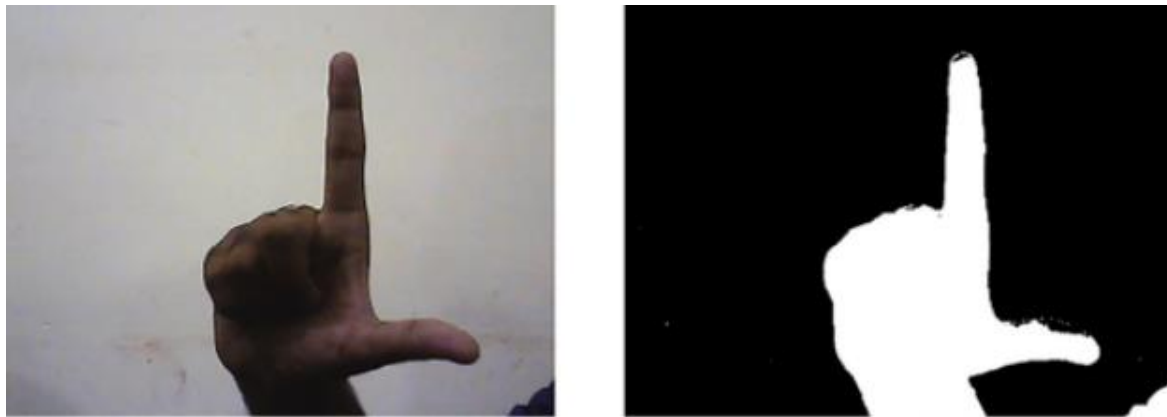


Fig 15. Hand Segmentation

In detection phase, there are 2 tasks. 1st task is the shape classification of the segmented images. To make the system robust, they use Fast Fourier Transform (FFT). Hereby, regardless of object’s orientation, good results are achieved. For the second task, the hand form is localized by using homography which enables the system to place the hand configuration in the reference position.

In Sign Recognition, inputs are the outputs retrieved from detection phase. Those vectors are normalized, and trained in Support Vector Machines. As there are 37 classes, methodology is one against all (One VS All). Kernel of the SVM is radial basis function which gives better results than polynomial or sigmoid functions.

Accuracy of the system fluctuates between 80%-90%. Reason for those instabilities is the lack of sufficient data and simplicity of the used algorithm (SVM).

In the paper “Australian sign language recognition”, authors **Eun-Jung Holden, Gareth Lee, and Robyn Owens** discuss about building SLR system for Australian Sign language [21]. Auslan is Australian Sign Language which is different than other sign languages. In addition to hand orientation and movement, facial expressions are also important for Auslan sign language. Therefore, strong tracking system is needed for detecting 3 objects; face and 2 hands. However, building such tracking system is also difficult because in some signs, hands and face overlap. In other words, hands cover some part of the face. This problem in sign language recognition systems is called occlusion. Occluded objects need to be identified and segmented. In the figure below, “thank you” in Auslan is illustrated.



Fig 16. “Thank You” in Auslan Sign Language

The system authors proposed has 3 components. First component is tracking module which detects hands and face and segments the occlusions. Second component is feature extraction module which extract features from face and hands. The final component is recognition module which is based on Hidden Markov Model (HMM).

Tracking module first detects the hands and face. If there is not any occlusion, 2nd component is ready to extract features from them. In another case or if there is any occlusion, several processes have been applied. In occlusion, hands cover the face. In other words, hands become on the foreground where face remains in the background. When occlusion is detected, module uses snake algorithm to detect the contours of the hand using snake algorithm. After full detection of hand, it is segmented and removed. For the lacking parts of the face, previous and next frames are taken as reference.

For the feature extraction, several geometrical functions are used such as angles among hands and head, moving direction, roundedness, change of position with respect to previous frame and more. So the features can be categorized into 2 groups; intraframe and interframe. Intraframe is the features that can be retrieved only from current frame. Interframes are the features we get after comparing the current frame with previous frame.

For the model selection, they propose the use of HMM. HMM is a probabilistic model that generalize features which associated with events over a period of time.



Fig 17. Contour Detection by Snake Algorithm

The dataset has 379 samples of 14 distinct sentences. The sentences are simple phrases used by everyday such as “what is your name?”, “where do you live”, and etc. 216 samples out of 379 have been used for testing, and remaining 163 samples have been used in testing.

The system has accuracy of 97% on sentence level and 99% on word level recognition. The false predictions are assumed to be the cause of coarticulations.

1.5 Assumptions and Limitations

In this section, I will talk about some challenges that are directly related to the Computer Vision side. When collecting data or real-time video capturing, one of the main challenges are environmental concerns. Lighting sensitivity, background noise, camera position and so on. Powerful models are able to deal with such problems but they are harder to implement comparatively. In another scenario, suppose that some of the fingers or entire hand moved out of the field of view of the camera. For a short period of time, a probabilistic model can handle such a situation. On the other hand, if it occurs continuously, that can create some overhead.

Furthermore, it is important to draw a strict line when a particular sign ends and next begins. This problem is called “Sign Boundary Detection”. Sometimes, a sign can be affected by preceding or succeeding signs (Co-articulation). Several languages also have static words or letters which brings another task to deal with which is integration. NLP tasks become complex depending on the converted output language, however today’s machine learning techniques are powerful enough to deal with it.

Final challenge we want to mention in this paper is computational cost. As sign language recognition systems require processing frames of videos in multilayer perceptrons, it requires high end GPUs or TPUs. Having sufficient memory access is also essential.

Today, several sign language recognition systems are developed such as American, German, French, and Russian sign language, and some are still in development. Mainly, there are 2 approaches; hardware-based approach and pure ML techniques. In the former one, several hardware are used to extract features of hands. One of them is the use of sensor gloves. It is just like usual gloves we use in our lives with cables connected to computers. It consists of multiple sensors that capture position of joints and their correlation while making moves. In other words, these gloves are extracting important features of hands which then will be used for training. Another example is the use of Microsoft Kinect to estimate human body poses. Using a depth sensor it extracts important features to generate a 3D virtual version of hand. Hardware based approaches are kind of old ones and especially for data collection, it became costly, and also Microsoft stopped support for Kinect devices.

With the development of ML tools and frameworks, hardware-based techniques started to disappear. CNN (Convolutional Neural Network) models are the widespread technique for sign language recognition systems. It takes images as input and classifies it to the corresponding letter or word. The Turkish Sign Language recognition system built in CNN + FPM (Feature pooling Model) + BLSTM (Bi-directional Long-Short Term Memory) + Attention model gave the best results. Moreover, “MediaPipe” framework is a recent trend for feature extraction of both pose and hand landmarks. Then, those features are trained in CNN, LSTM, HMM and in other models for sign language recognition.

The static and dynamic letters that are similar such as ‘O’ and ‘Ö’ introduction in section 1 create challenges in implementation. Whether to use 2 separate models that recognize each type or having a unified model. If the answer is 2 separate models, how to determine when to use which model? In other words, how to determine a letter is static or dynamic from a web camera where actually every sign seems dynamic. For a unified model, things got even more complicated.

What about treating all letters as static or dynamic? Second option does not sound logical because we cannot add moves to the static letters. For the first option, as in the previous example, the letter ‘O’ and ‘Ö’ would be the same.

Another proposal can be just ignoring dynamic letters as they are small in numbers in comparison to static letters. In this case, letters ‘Ö’ and ‘Ü’ would be recognized as ‘O’ and ‘U’ correspondingly. Dynamic letters with no opposition in static letters such as ‘K’ and ‘Z’ would be recognized as nothing. After recognition, those letters that form words can be modified using lexicon-based verification. For this purpose, letter frequencies from sentences which deaf-mute people mostly use are extracted. This goal is achieved with the help of the statistical HMM (Hidden Markov Model) model.

Data used in the statistical model is retrieved from the book “Əsərlər, hekayələr, povestlər” written by Anar Rzayev. In the upcoming tables, we provide some statistics about the used data such as most and less frequent words and letters.

<i>10 Most Frequent Words</i>	<i>Frequencies</i>
bir	2890
bu	2102
və	1776
mən	1175
amma	674
heç	674
elə	666
onun	563
sonra	542
belə	479

Table 10. 10 Most Frequent Words in Our Lexicon Dataset

<i>10 Less Frequent Words</i>	<i>Frequencies</i>
barışa	1
şüurumuzasığışmaz	1
vərdeşli	1
analogiyalar	1
açıklığında	1
zəiflik	1
qırıldadan	1
oğulları	1
yatağına	1
kitabxanadan	1

Table 11. 10 Less Frequent Words in Our Lexicon Dataset

As we have 32 letters in AzSL there may be up to 1024 (32x32) letter combinations. Some of them are morphologically impossible and some of them are not present in our dataset. Probabilities of 628 combinations out of 1024 are equal to 0. Rest of them have distributions between 0 and 1. In the below table, you can find 10 most probable letters given the previous letters.

<i>Letter Given Particular Letter</i>	<i>Frequency</i>	<i>Probability</i>
ı ğ	755	0.514
n ı	2411	0.372
i j	32	0.314
ə c	456	0.301
a y	1519	0.288
n i	3223	0.286
ə l	2863	0.279
ə h	432	0.277
a v	418	0.275
a d	2208	0.272

Table 12. 10 Most Probable Letter Given Particular Letter in Our Lexicon Dataset

Beyond all, there are some challenges of building a strong model due to some similarity between static letters. In the upcoming sentences, we will talk about those similarities and provide examples. First case is the similarity between ‘L’ and ‘P’.



Fig 18. Static Letters 'L' and 'P'

As you may notice, the same fingers are used in order to interpret the letters. Only difference is for 'L' fingers are parted away where for 'P' they are joint. Think about a scenario where a user of this system shows a letter something similar to 'L' or 'P' where finders are separated in a manner that they are almost joint. Model would have hard times to recognize the letter. Another example is letters 'M' and 'T' where 3 fingers are used.

Another challenge in recognition of letters is that some letters are almost identical from side-view. Letters 'C' and letter 'J' are examples of that type. From the front-view they are clearly identifiable. However, when it comes to the side-view, it becomes complicated.



Fig 19. Static Letters 'C' and 'J' from front view



Fig 20. Static Letters 'C' and 'J' from side view

2 RESEARCH APPROACH AND METHODOLOGY

We have experimented with a variety of models and with a variety of parameters for recognition of static letters. In this section, we will talk about them.

2.1 Convolutional Neural Network (CNN)

CNN is a strong class of Artificial Neural Network (ANN) generally designed for analyzing visual content. It is a widely spread exercise to use CNN for image classification. Here, rather than using MediaPipe results as an input to the CNN, we use raw images. However, our dataset contains images with different sizes. Beyond that, to reduce the computational complexity, images are resized to 128x128. Output of the NN will be 24 as we have 24 static letters in AzSL. Total parameters of the model are more than 800,000. Layers of the sequential model are as follows:

<i>Layers</i>	<i>Parameters</i>			
<i>Input Layer</i>	input_shape = (128, 128, 3)			
<i>Conv2D</i>	filters = 32	kernel_size = (3, 3)	padding = 'same'	activation = 'ReLU'
<i>MaxPooling2D</i>	pool_size = (2, 2)			
<i>Conv2D</i>	filters = 64	kernel_size = (3, 3)	padding = 'same'	activation = 'ReLU'
<i>MaxPooling2D</i>	pool_size = (2, 2)			
<i>Conv2D</i>	filters = 64	kernel_size = (3, 3)	padding = 'same'	activation = 'ReLU'
<i>Flatten</i>				
<i>Dense</i>	units = 64		activation = 'ReLU'	
<i>Dense</i>	units = 24		activation = 'Softmax'	

Table 13. Parameters of CNN model

Overall accuracy of the CNN model is 60%. While training, after some point, validation accuracy stops increasing while training accuracy hits 100%. This gap between training and validation accuracy is the indicator of overfitting. It is obvious that NNs require more data than simple classification algorithms such as Logistic Regression. In conclusion, it is not a good idea to feed the NN model with raw images at this point. We need a stronger algorithm to exploit the most important features of the image which we can train later on. What we need is Mediapipe.

2.2 MediaPipe + Simple Classification Algorithms

Results we got from MediaPipe are fit to the simple classification algorithms such as Logistic Regression, Ridge Classifier, Random Forest Classifier, and Gradient Boosting Classifier. It is worth mentioning that the data we retrieved from MediaPipe are in the shape of (21, 3). Those data are scaled using the Standard Scaler. For training, approximately 15,000 samples are used. After successful training, models are tested on a dataset containing around 4,000 samples. In the table below, testing accuracies are illustrated for each model.

<i>Classification Model</i>	<i>Accuracy</i>
Logistic Regression	85
Random Forest Classifier	85
Gradient Boosting Classifier	83
Ridge Classifier	79

Table 14. Simple Classification Algorithms and Accuracies

2.3 MediaPipe + Support Vector Machines (SVM)

Results of the Logistic Regression was around 85% which is insufficient. Next is training the data using Support Vector Machines (SVM). SVM supports both classification and regression. We have used SVC (Support Vector Classifier) in this task. As the number of targets is 24, the kernel of the SVC is set to be 'poly' which means vectors that separate data are polynomial. And the degree of the polynomial kernel function is 5.0. Overall accuracy is 84%.

2.4 MediaPipe + Deep Neural Network (DNN)

SVM and other classifications such as Logistic Regression do not seem generalizing well. For this purpose, we tried Deep Neural Network (DNN) for better results. Network is trained with 15,000 samples. Input size is 63 (21x3). Model is a simple network consisting of 5 Dense layers. Dense layer except the last one followed by Dropout layer. Accuracy is 90%.

Layers	Parameters
Input Layer	input_shape = 63
Dense	units = 128 activation = "ReLU"
Dropout	rate = 0.1
Dense	units = 256 activation = "ReLU"
Dropout	rate = 0.1
Dense	units = 256 activation = "ReLU"
Dropout	rate = 0.1
Dense	units = 128 activation = "ReLU"
Dropout	rate = 0.1
Dense	units = 24 activation = "Softmax"

Table 15. Parameters of DNN model

All models we have tried so far failed to go over 90% accuracy. In order to detect what is going on, it is always a good idea to check the confusion matrix.

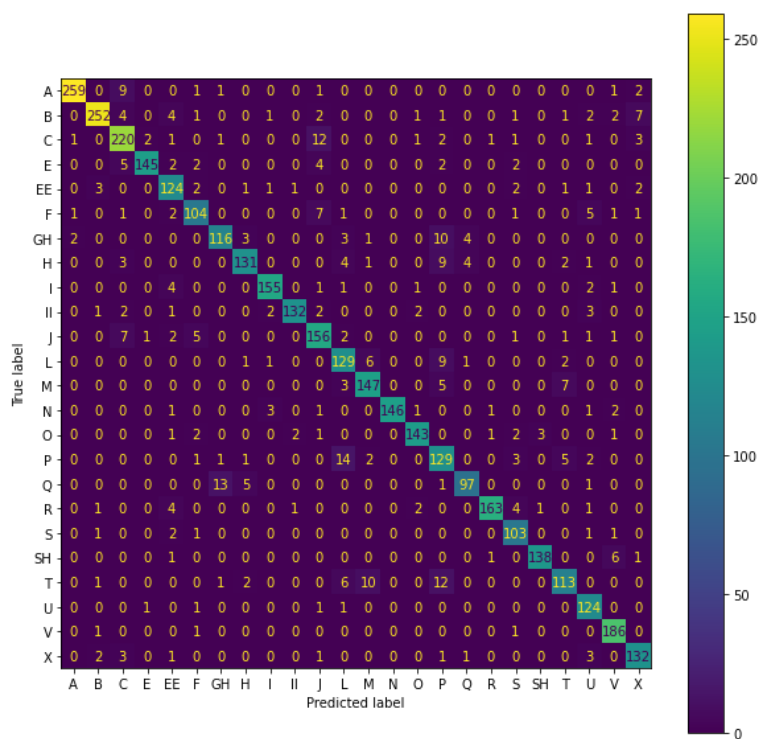


Fig 21. Confusion Matrix

2.5 MediaPipe + 2-Level Deep Neural Network (Clusterization + DNN)

In the previous section, we have mentioned that several letters are very similar. It is also reflected in the performance of the model according to the confusion matrix. Single NN tries to differentiate 24 classes. What if we design multi NNs rather than single NN, and every NN classifies similar letters? But how to find the letters that form clusters?

Firstly, we tried to do it by unsupervised learning. All data used in the clusterization process using the K-means algorithm. However, the results were not satisfactory. Whether all samples of a particular class were not in the same cluster or classes in the particular cluster were not what we want.

Therefore, we made clusterization manually according to the confusion matrix. Clusters are as follows:

<i>Clusters</i>	<i>Letters</i>
Cluster 1	B, X
Cluster 2	C, F, J
Cluster 3	Ĝ, Q
Cluster 4	L, P
Cluster 5	M, T
REST	A, E, Θ, H, I, Ì, N, O, R, S, Ş, U, V

Table 16. Clusters and Letters

Now, we will talk about how the system works. In the first level, there is a NN that finds which cluster a sample belongs to. In other words, it has 6 targets; ‘Cluster 1’, ‘Cluster 2’, ‘Cluster 3’, ‘Cluster 4’, ‘Cluster 5’, and ‘REST’. In the second level, there are 6 different NNs each for one cluster. Input comes to the 1st level and is directed to the corresponding 2nd level NN. Targets of the second level NNs are their included letters. 2nd level NNs give the final decision. All NNs are trained with different parameters so that best performance is achieved.

About the evaluation of the model, 1st level NN predicts the cluster of the samples in the test dataset. If it mispredicts the cluster, it is counted as wrong prediction, else sample moves to the corresponding 2nd level NNs. Now, a sample is tested on a NN which is specialized to differentiate its targets. Test accuracy is around 94%. For overall architecture, you can refer to the graph below.

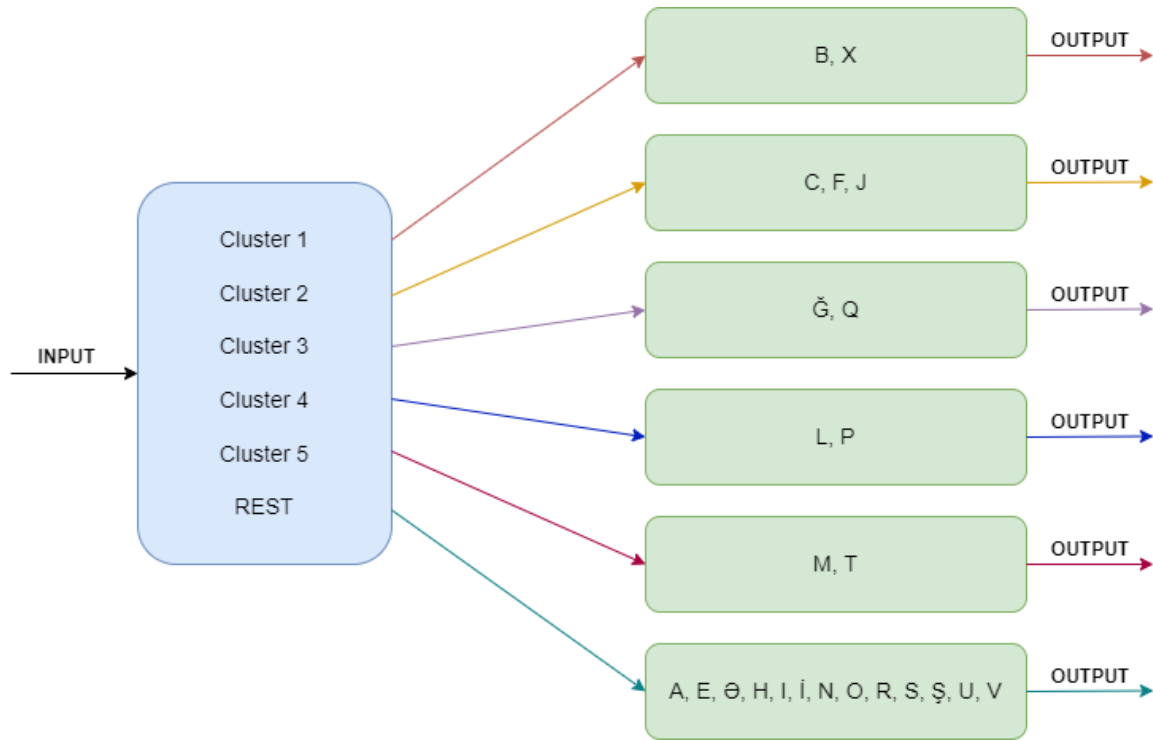


Fig 22. Architecture of Clusterization

2.6 Beam Search + Lexicon Verification

Now, we have a model that predicts sign language letters. It is quite normal that it can make false predictions as accuracy is not 100%. For this purpose, we propose using beam search and lexicon verification.

Beam Search is a heuristic search algorithm. Only parameter it takes is 'width'. Let's think about a scenario where width is equal to 3. Our model predicts the first letter interpreted in front of the camera. In the softmax layer, probabilities of all 24 classes are reflected. Beam Search with width 3 takes 3 outputs with maximum probability. For the second letter, it does the same while taking 3 highest results. Here, we fix the position of the first 3 results of the 1st position and concatenate the next 3 results. Hereby, 9 (3x3) outputs are generated. 3 highest of them are again picked up where others are dropped. This process continues till the last token. As our width is 3, we will have 3 outputs at the end where we will verify them using our probabilistic data. In the chart below, step by step Beam Search for the word 'SALAM' which means 'HELLO' in Azerbaijani is illustrated. Width is set to 3.

S	0.99	A	0.99	L	0.99	A	0.99	M	0.99
ə	0.0000431	U	0.00028	P	0.00566	U	0.000266	L	0.0000124
J	0.0000167	C	0.0000496	H	0.00115	B	0.000002478	T	0.0000000045

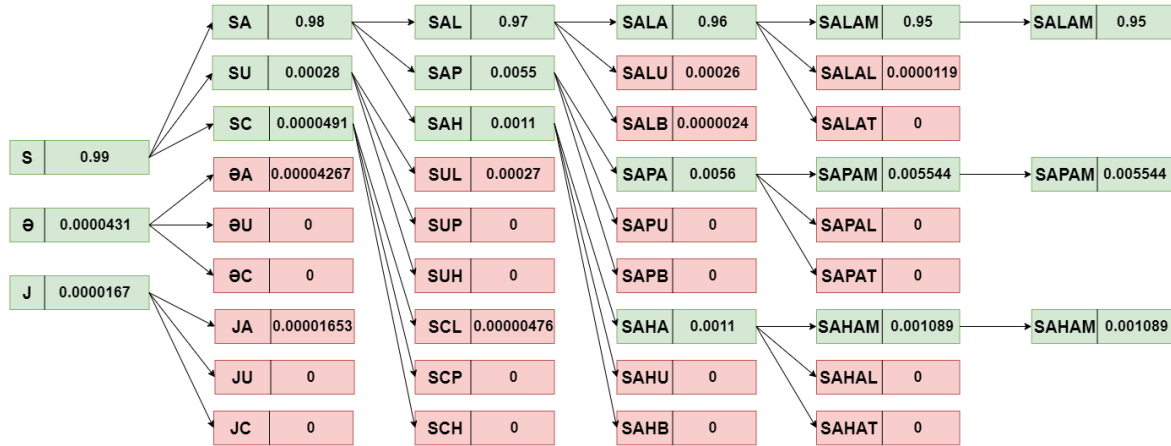


Fig 23. Beam Search Example

In the table above, 3 highest probabilities of each recognition of letters retrieved from the softmax layer are shown. Then, for each letter, outer product has been applied followed by taking 3 highest and dropping others. In the end, we got 3 highest possible results. As you notice, the model predicted ‘SALAM’ successfully with 95% accuracy. Other 2 outputs have very low probabilities but in some cases difference can be little or they can be useful for lexicon verification.

After doing a Beam Search, there are a number of results which are equal to width. We have provided earlier a table of letter conversion with probabilities such as probability of ‘b’ given ‘a’. In order to calculate lexicographical probability of words, we will multiply all probability of letters. For instance, $P(S) * P(A|S) * P(L|A) * P(A|L) * P(M|A)$ is the lexicographical probability of the word ‘SALAM’.

For Lexicon Verification, we will calculate lexicographical probability of all outputs of the Beam Search. Later, lexicographical probabilities will be multiplied with the probabilities that Beam Search generates. The resulting probabilities that number of them is equal to width will be final results. The word with highest probability is the result.

The goal of lexicon verification is to drop words generated by Beam Search which is not possible by language rules. For instance, the letter ‘i’ cannot follow the letter ‘i’. If the model predicts in that way with high probability, probability after lexicon verification will be 0 as $P(i|i) = 0$.

	Beam Search probabilities		Lexicographical probabilities		Result
bir	0.60	X	$P(\text{bir}) = P(b) * P(i b) * P(r i) = 0.067 * 0.176 * 0 = 0$	=	0
bir	0.10	X	$P(\text{bir}) = P(b) * P(i b) * P(r i) = 0.067 * 0.176 * 0.094 = 0.0011$	=	0.00011
bin	0.02	X	$P(\text{bin}) = P(b) * P(i b) * P(n i) = 0.067 * 0.176 * 0.286 = 0.0034$	=	0.00007

Fig 24. Lexicon Verification Example

According to the graph above, although the model predicted the word as “bir” with more probability, the correct word is “bir” due to the fact that letter ‘i’ cannot come after letter ‘i’.

In upcoming sentences, we will sum up everything. Inputs of the system are the frames retrieved from the live camera. When a hand is detected, MediaPipe extracts the features, and the model predicts the letter. If the probability of the most probable letter is more than 80%, the system pretends it as an intentional sign language gesture. When the user finishes the word, the system calculates the Beam search probabilities using the softmax values of each letter. Depending on the Beam search width, the system keeps the number of candidate words. Lexicographical probabilities for each candidate word are calculated and multiplied by Beam search probabilities. Final result is the word with the highest product. That word is displayed on the screen with its Beam search probability.

3 RESEARCH RESULTS AND ANALYSIS OF RESULTS

As written in Section 2, different experiments are held. Aim of all research and experiments was to find the best model with highest performance. In this section, I will briefly cover the models that have been implemented and their results.

First model was CNN model. As you know, CNN is a widely used NN model to extract features. However, the results were not satisfactory. There are several reasons of it. First of all, all the signs are interpreted with single hand. Only difference is orientation of fingers. CNN model could not be able to exploit those differences. Different models with different number of layers and neurons have been trained. But the overall accuracy did not pass to accuracy of 60%. The second reason is the data. Data we have has not enough number of samples for training in CNN. It is known that for a Neural Network model, qualitative and quantitative data is needed especially when training images in CNN model. Although, several data augmentation techniques are applied, it was also not helpful.

After the failure of CNN model, I have looked for other feature extraction techniques. The answer was MediaPipe. Using MediaPipe hand landmarks, 21 joints of hands are extracted for each image. Each feature is described using 3 features which are the coordinates. In total, there are 63 (21x3) features (floating numbers) for each hand. Those numbers are saved as “NumPy” files so that feature extraction algorithm runs once for all samples of the data. Then, all features are collected in single “CSV” file (.csv). Now, we have a single file that can be used in training processes.

Those features are trained in various models and algorithms. First, I started with simple algorithms such as Logistic Regression, Random Forest Classifier, Gradient Boosting Classifier, and Ridge Classifier. Accuracies are 85%, 85%, 83%, and 79% correspondingly. Although performance of the system increased in comparison to CNN model, those simple classification algorithms could not generalize well for our features.

Next experiment was training MediaPipe features in Deep Neural Network. Multi-Layer Perceptron (MLP) from Scikit-learn and Deep Neural Network (DNN) from TensorFlow have been used. The difference between them is that MLP is fully connected network in comparison to DNN. Additionally, in MLP, we only set hidden unit sizes where DNN is more customizable. The final decision was to use DNN from TensorFlow. Different DNN models with different parameters are tested. The best result was 90% of test accuracy.

After careful analysis on the different metrics, I detect that the only problem with the model was that some letters are similar. Therefore, I manually divided letters into clusters. Similar letters are trained in the same cluster. In other words, first NN classifies the letters according to clusters. Then, corresponding cluster’s NN finds which letter is it in that cluster. Accuracy of this 2-level NN model is 94%. This result is the best result we got for our dataset.

4 SUMMARY AND CONCLUSIONS

Aim of this paper is to research existing solutions to the Sing Language Recognition systems. I have read plenty amount of papers and books for this goal. Some of them are reviewed in Section 1.4 Review of Significant Research.

In general, there are 2 types of implementations of Sign Language Recognition systems. 1st one is traditional and outdated one. This type of SLR systems require special hardware for implementation and usage. According to reviewed papers, using sensor-based gloves is one of them. Those gloves extract important features from the hand parts. For this reason, those gloves are involved in whole data collection task. Those features can be trained in various model. However, for further usage, those gloves are also needed which creates another problem of accessibility. They are expensive that not everyone can afford. Another example can be Microsoft Kinect devices. Depth images are captured with it which are the features. Accessibility problem also present in such devices.

2nd type is the use of pure Machine Learning techniques. Deep Neural Networks, Convolutional Neural Networks, Vision Transformers (ViT) and others are used for this purpose. In comparison to the 1st type, they do not require any specific hardware. Therefore, they have less cost but require more computational power.

Our aim in this paper is to come up with a solution that uses the pure ML techniques. As AzSL has 32 letters (24 static letters and 8 dynamic letters), our dataset contains both images and videos. Totally, we have 17,000 images and videos. Scope of this paper is to propose solution for system that recognizes static letters only. So, we have used images only.

First experiment was the CNN model. Models with different neurons, filters, and layers including convolutional, max pooling, batch normalization, dropout, and dense layers have been used. The best CNN model had holdout accuracy of 60% which is not satisfactory. As CNN failed to extract features, we switched to the MediaPipe which is open-source framework that extract important features of human pose and body parts.

For Azerbaijani Sign Language Recognition system, what we need is only hand landmarks of MediaPipe. Features for all of images in the dataset have been extracted. Firstly, they are trained in simple classification methods including Logistic Regression, Random Forest Classifier, Gradient Boosting Classifier, and Ridge Classifier. The highest performance is achieved by Logistic Regression with the accuracy of 85%.

Last experiment was the use of DNN and MediaPipe features together. With fine tuning of the parameters of network, accuracy reached to the 94%. We integrated our best model with Beam Search and lexicon verification. Beam Search finds ‘n’ (width) best results with their probabilities. Later, those ‘n’ results are verified by lexicon verification. For lexicon verification, letter probabilities extracted from one book which is in Azerbaijani language. Lexicon probabilities of those ‘n’ results are calculated and multiplied by probabilities of Beam Search. Highest multiplication is the output of the system.

REFERENCES

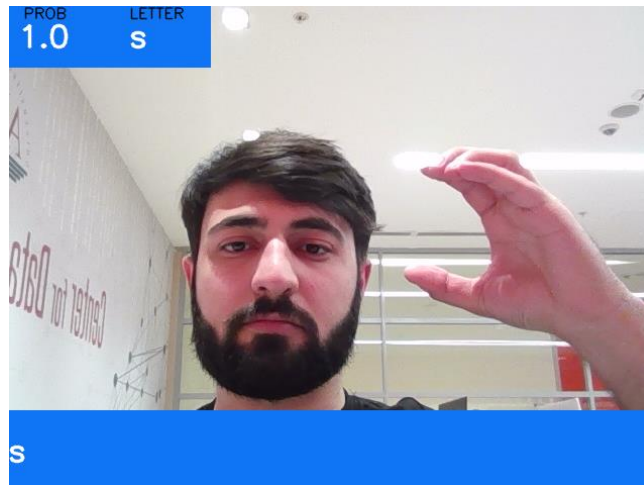
- [1] Huileng Tan. 2022. Meta, formerly Facebook, is no longer one of the world's top 10 most-valuable companies. Business Insider. <https://www.businessinsider.com/meta-no-longer-one-of-the-worlds-top-10-most-valuable-companies-2022-2>
- [2] World Health Organization. 2011. World Report on Disability 2011. <https://www.who.int/teams/noncommunicable-diseases/sensory-functions-disability-and-rehabilitation/world-report-on-disability>
- [3] Andrew Perrin, Sara Atske. 2021. Americans with disabilities less likely than those without to own some digital devices. Pew Research Center. <https://www.pewresearch.org/fact-tank/2021/09/10/americans-with-disabilities-less-likely-than-those-without-to-own-some-digital-devices/>
- [4] Lookout - Assisted vision. Google LLC. Google Play Store. <https://play.google.com/store/apps/details?id=com.google.android.apps.accessibility.reveal>
- [5] RogerVoice. <https://rogervoice.com/en/>
- [6] Wemogee. 2017. Samsung Wemogee: A New Communication Tool for People with Language Disorders <https://news.samsung.com/global/samsung-wemogee-a-new-communication-tool-for-people-with-language-disorders>
- [7] Helen Cooper, Brian Holt, Richard Bowden. 2011. Sign Language Recognition. In: Visual Analysis of Humans. Springer, London. https://doi.org/10.1007/978-0-85729-997-0_27
- [8] Lionel Pigou, Sander Dieleman, Pieter-Jan Kindermans, Benjamin Schrauwen. 2015. Sign Language Recognition Using Convolutional Neural Networks. In: Computer Vision - ECCV 2014 Workshops. ECCV 2014. Lecture Notes in Computer Science(), vol 8925. Springer, Cham. https://doi.org/10.1007/978-3-319-16178-5_40
- [9] Sergio Escalera, Xavier Baró, Jordi González, Miguel A. Bautista, Meysam Madadi, Miguel Reyes, Víctor Ponce-López, Hugo J. Escalante, Jamie Shotton, Isabelle Guyon. 2015. ChaLearn Looking at People Challenge 2014: Dataset and Results. In: Agapito, L., Bronstein, M., Rother, C. (eds) Computer Vision - ECCV 2014 Workshops. ECCV 2014. Lecture Notes in Computer Science(), vol 8925. Springer, Cham. https://doi.org/10.1007/978-3-319-16178-5_32
- [10] Mohamed Aktham Ahmed, Bilal Bahaa Zaidan, Aws Alaa Zaidan, Mahmood Maher Salih, Muhammad Modi bin Lakulu. 2018. A Review on Systems-Based Sensory Gloves for Sign Language Recognition State of the Art between 2007 and 2017. Sensors (Basel, Switzerland), 18(7), 2208. <https://doi.org/10.3390/s18072208>
- [11] Cao Dong, Ming C. Leu, Zhaozheng Yin. 2015. American Sign Language alphabet recognition using Microsoft Kinect. Conference: 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). <https://doi.org/10.1109/CVPRW.2015.7301347>
- [12] Nicolas Pugeault, Richard Bowden. Spelling it out: Real-time ASL fingerspelling recognition, 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2011, pp. 1114-1119, <https://doi.org/10.1109/ICCVW.2011.6130290>
- [13] Ozge Mercanoglu, Hacer Keles. 2020. AUTSL: A Large Scale Multi-Modal Turkish Sign Language Dataset and Baseline Methods. 8:181340-181355. <https://doi.org/10.1109/ACCESS.2020.3028072>
- [14] M. G. Grif, Y. K. Kondratenko. 2021. Development of a software module for recognizing the fingerspelling of the Russian Sign Language based on LSTM. Journal of Physics Conference Series 2032(1):012024. <https://doi.org/10.1088/1742-6596/2032/1/012024>

- [15] Matyas Bohacek, Marek Hruz. 2022. Sign Pose-based Transformer for Word-level Sign Language Recognition. University of West Bohemia, Faculty of Applied Sciences, Department of Cybernetics and New Technologies for the Information Society Technicka 8, 301 00 Plzen, Czech Republic
- [16] Dongxu Li, Cristian Rodriguez, Xin Yu, Hongdong Li. 2020. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In Proceedings of the IEEE/CVF winter conference on applications of computer vision, pages 1459–1469.
- [17] Facundo Quiroga, Franco Ronchetti, Cesar Armando Estrebow, Laura Cristina Lanzarini, Alejandro Rosete. 2016. Lsa64: An Argentinian sign language dataset. In XXII Congreso Argentino de Ciencias de la Computacion (CACIC 2016), pages 794–803.
- [18] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, Richard Bowden. 2020. Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020. <https://doi.org/10.1109/CVPR42600.2020.01004>
- [19] Oscar Koller. 2014. RWTH-PHOENIX-Weather 2014: Continuous Sign Language Recognition Dataset. Human Language Technology & Pattern Recognition Group. RWTH Aachen University, Germany. <https://www-i6.informatik.rwth-aachen.de/~koller/RWTH-PHOENIX/>
- [20] Ali Imran, Abdul Razzaq, Irfan Baig, Aamir Hussain. 2021. Dataset of Pakistan Sign Language and Automatic Recognition of Hand Configuration of Urdu Alphabet through Machine Learning. 36(2):107021. <https://doi.org/10.1016/j.dib.2021.107021>
- [21] Eun-Jung Holden, Gareth Lee, Robyn Owens. 2005. Australian sign language recognition. Machine Vision and Applications 16, 312 (2005). <https://doi.org/10.1007/s00138-005-0003-1>

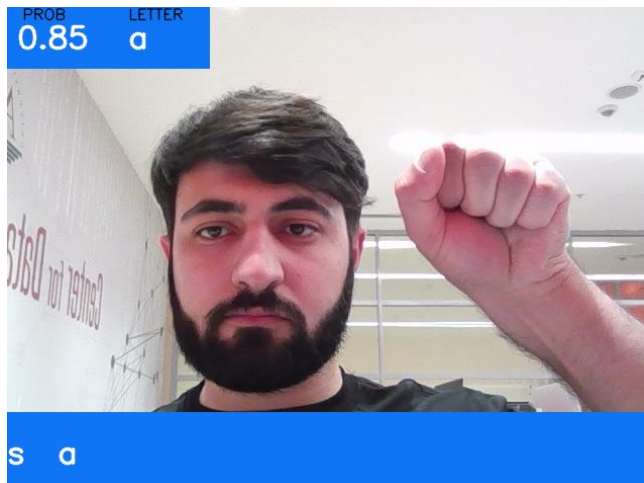
APPENDIX

In this section, I will share screenshots from live camera. For demo purpose, I am fingerspelling the word “salam” step-by-step which means “hello” in Azerbaijani.

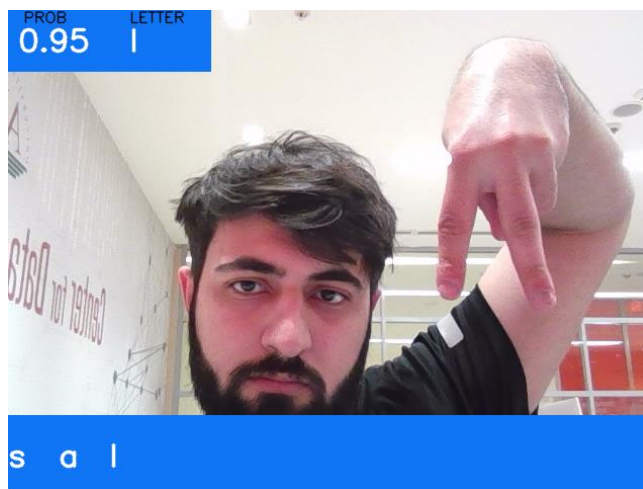
- 1) ‘s’ – 100% Recognition Rate



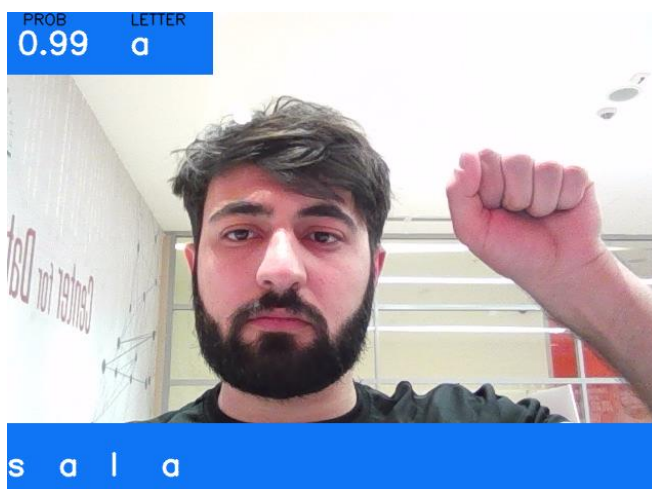
- 2) ‘a’ – 85% Recognition Rate



3) 'l' – 95% Recognition Rate



4) 'a' – 99% Recognition Rate



5) 'm' – 90% Recognition Rate



After fingerspelling the letters, Beam Search and Lexicon Verification occurs. Calculations are made as in the Figures 23 and 24 in Section 3. The highest probable word is illustrated together with its Beam Search probability. The image below illustrates the output.

