Ramaiah Institute of Technology

(An Autonomous Institute, Affiliated to VTU)

MSR Nagar, MSRIT post, Bangalore - 54

A Dissertation PBL Report on

Sign Language Recognition

Submitted by

M Sneha	1MS14CS058
Mipsa Patel	1MS14CS148
Tilak S Naik	1MS14CS134
Vibha Karanth	1MS14CS136

Bachelor of Engineering in Computer Science & Engineering

Under the guidance of

Pramod Sunagar
Assistant Professor
Department of Computer Science
Ramaiah Institute of Technology

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

M. S. RAMAIAH INSTITUTE OF TECHNOLOGY
(Autonomous Institute, Affiliated to VTU)
BANGALORE - 560054
www.msrit.edu, 2017

DECLARATION

I Student of seventh semester BE, Dept of Computer Science and Engineering, Ramaiah Institute of Technology, Bangalore, hereby declare that the project entitled "Sign Language Recognition", thesis completed and written by me under the guidance of Pramod Sunagar, Dept of CSE, Bangalore for the partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering.

Place: Bangalore
Date: 28th October, 2017

(1MS14CS058 M Sneha)
(1MS14CS148 Mipsa Patel)
(1MS14CS134 Tilak S Naik)
(1MS14CS136 Vibha Karanth)

ACKNOWLEDGEMENT

First and foremost, my utmost gratitude to Pramod Sunagar, Dept of CSE, MSRIT whose sincerity and encouragement we will never forget. He has been our inspiration as we overcame all the obstacles in the completion of this project work.

Dr. Anita Kanavalli, Head of the Department of Computer Science and Engineering, had kind concern and consideration regarding project work and we would like to thank her for continuous support.

We would like to thank our beloved principal Dr. N. V. R. Naidu for his support and encouragement.

This work would not have been possible without the guidance and help of several individuals who in one way or another contributed their valuable assistance in preparation and completion of this study.

We would like to express sincere thanks to all the teaching and non-teaching faculty of CSE Department and my dear friends who helped in all the ways while preparing the report.

Abstract

Sign language is the language used by the hearing and speech impaired to communicate among themselves and others. Unless the concerned people know the sign language properly, there is a communication barrier between them. In most cases, an interpreter is required to carry out such a conversation. To reduce the dependency of the hearing and speech impaired people on interpreters, we develop a sign language detection system that identifies the gestures from images and videos, which can then be converted into grammatically correct sentences in any language, using various deep learning techniques.

List of Figures

Contents

1 Introduction

1.1 General Introduction

Sign Language is a vision based language that uses the movements of the hands and facial expressions for communication. Subtle differences among different hand gestures can have a huge impact on the meaning that they convey. Along with this, it is important to prevent the background noise of the image from affecting the gestures being communicated. We consider such problems and challenges and develop an effective sign language recognition system.

1.2 Statement of the Problem

There exist models that convert American Sign Language to sentences. However, most of these techniques are not based on deep learning. These models also cannot handle the nuances of various sign languages. We use deep neural networks to develop a model that can identify gestures of sign languages and can later be converted into English text.

1.3 Objectives of the project

- To develop a generic sign language recognition system.
- To design a deep neural network which can give a better performance and robustness compared to the existing image processing and machine learning models.

1.4 Project deliverables

- An application that obtains frames from sign language communication videos, processes them, and obtains the meaning being conveyed by them by identifying the various gestures.
- Comparison of performance of our model with existing algorithms used for sign language recognition.

1.5 Current Scope

Image and video processing is one of the important fields in Machine Learning and Deep Learning at present. The huge number of images and videos on social media and otherwise has led to an increased interest in the field. The algorithms and techniques developed as a result of this can be used in more impactful fields like helping the speech and hearing impaired lead more convenient lives.

The hand signs along with the emotions displayed by the individual provide the content and tone of the sentence. A tool to convert their words into a widely used language enhances their ability to communicate.

1.6 Future Scope

Since such applications will always be necessary, research and development of new algorithms to improve the efficiency of interpretation of sign language will be endless. Deep learning has a lot more to offer than what has already been explored. With such possibilities, there will always be scope to develop newer models that can be used in sign language recognition. The models used in this can also be used in other applications such as gesture controlled systems, tracking unusual activity, etc. with slight tweaks. With slight modification, it can also be used for recognition of other sign languages.

Moreover, the current work can be further extended to output speech instead of text, in various languages, from various sign languages that are in use across the world. A similar generative model can be used to map speech or text to actions by stitching together pre-recorded videos or actually generate an interface with a skeletal structure. This model can be seamlessly integrated with various applications on different platforms to support high productivity.

2 Project Organization

2.1 Software Process Models

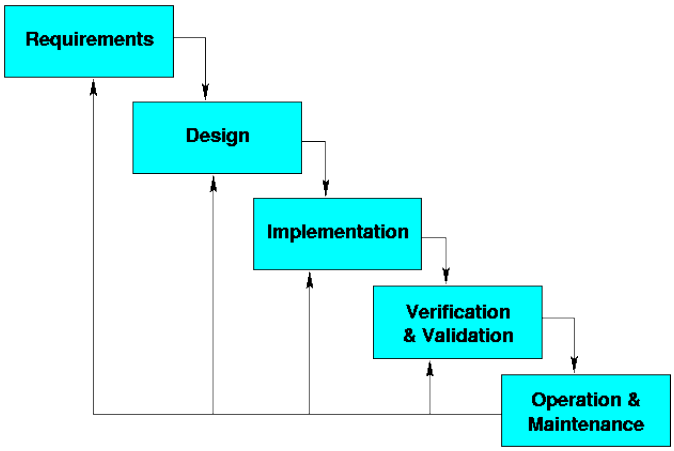

Figure 1: Iterative waterfall model

Figure 1 shows the process model being used for the project.

2.2 Roles and Responsibilities

Analysis of various requirements Sneha, Vibha, Mipsa and Tilak

Designing the architecture of the deep learning model Mipsa and Tilak

Assessing the data to be used Vibha

Implementation of CNN Tilak

Implementation of RNN Mipsa

Testing the implementation Sneha and Mipsa

Predicting outputs for new data Vibha and Tilak

3 Literature Survey

3.1 Introduction

There have been various approaches used to solve the problem of sign language recognition. However, most of these use image processing in MATLAB, upon which Machine Learning algorithms are applied. There has also been some work done using artificial neural networks.

3.2 Related work with the citation of references

[1] deals with the double handed Indian Sign Language. It is captured as a series of images, processed with the help of MATLAB and then converted to speech and text. However this approach uses image processing techniques which are highly sensitive to lighting conditions.

[2] proposes an image processing, computer vision and neural network based approach to identify the characteristics of the hand in images taken from a video through webcam. Identification of hand shapes from continuous frames is done by using series of image processing operations. Interpretation of signs and corresponding meaning is identified by using Haar Cascade Classifier. Finally displayed text is converted into speech using speech synthesizer.

[3] proposes a system that consists of three phases, a training phase, a testing phase and a recognition phase. In the training phase, each class is trained with a multiclass support vector machine (MSVM). Hu invariant moment and structural shape descriptors are combined to make a combinational feature vector that are to be extracted from the input image in the testing phase after applying preprocessing. In the recognition phase, different classes are used for testing an input gesture.

3.3 Conclusion of Survey

Deep neural networks have revolutionized how different Machine Learning problems are approached. Using them to solve this problem can give significant results, and that is precisely what this project explores.

4 Project Management Plan

4.1 Schedule of the project

Figure 2 shows the project schedule.

4.2 Risk Identification

Risks faced can be classified as:

- Wrong recognition of gestures as the program returns before waiting for further gestures that might occur after a time gap. (No explicit way to signal the end of a sentence).
- The same sentence might mean different things when said in different tones so the program also needs to take into account the expressions of the subject.
- Missing frames in the video due to improper data transfer or improper recording.
- Insufficient training.
- Incorrect training data.
- Insufficient lighting in the videos.
- Videos recorded from different angles.

5 Software Requirement Specification

5.1 Project Overview

Images of numbers and letters, and videos of words and sentences are fed as input to the model and interpreted text is received as output using various deep learning techniques.

5.2 Hardware

We use a system with NVIDIA® GeForce® 940MX (4 GB DDR3) GPU and 8 GB RAM to perform our experiment. A camera is required to record hand gestures and facial expressions.

5.3 Software Requirements

- Python
- OpenCV

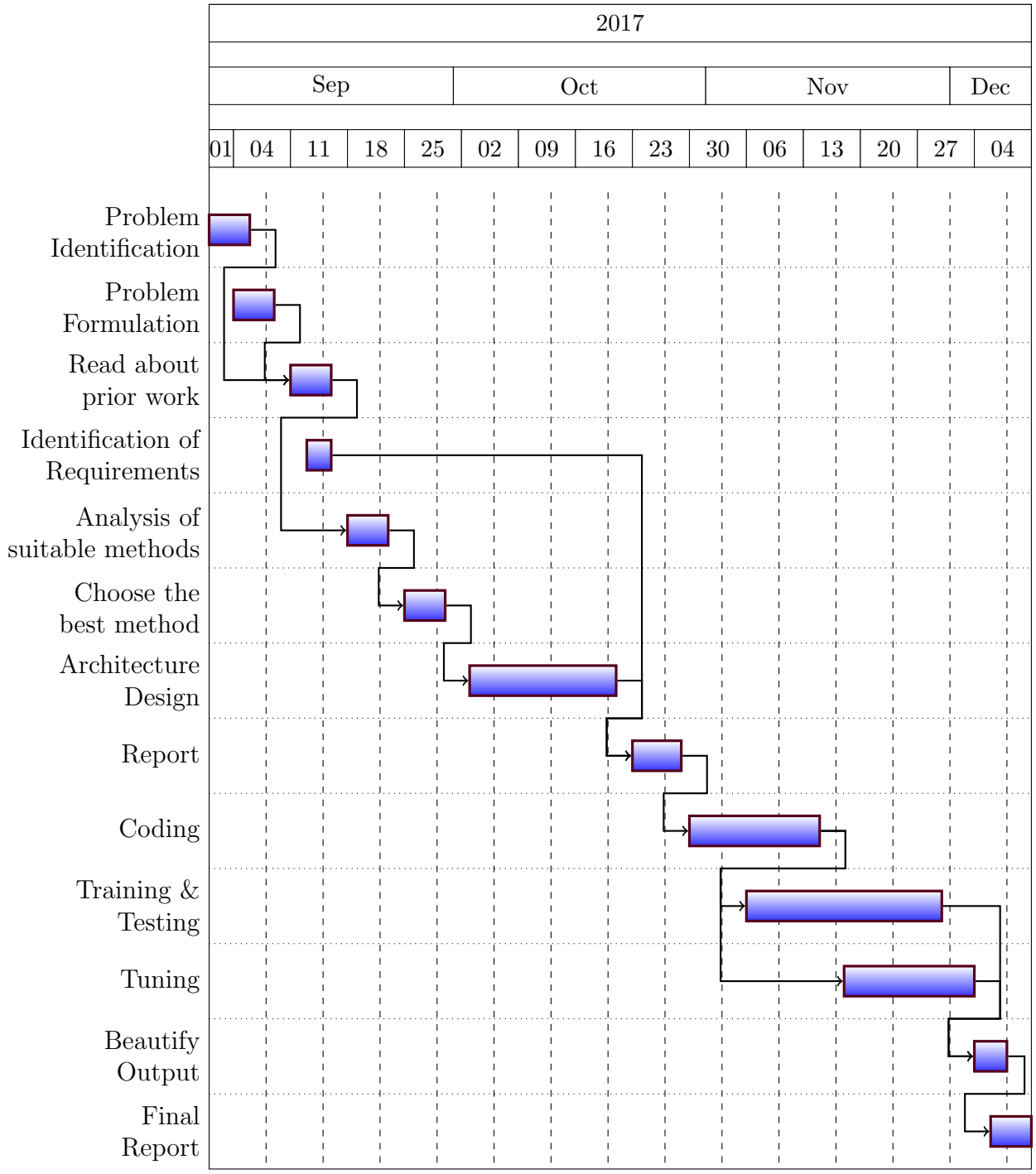

Figure 2: Timeline

- CUDA
- PyTorch for training and testing the deep neural network.

5.4 Functional Requirements

- Take image and video inputs of sign language conversation.
- Extract frames from videos for training.
- A training algorithm to reduce train-test error and generalize the model to make it invariant to lighting conditions.
- Convert the received output classes to letters, words or sentences.

6 Design

6.1 Introduction

This project has been designed such that it incorporates some of the major applications of machine learning and deep learning. Various concepts such as image and video processing, different neural network architectures, etc. are used. Simplicity and effectiveness are the key factors considered while designing and developing this application.

6.2 Architecture Design

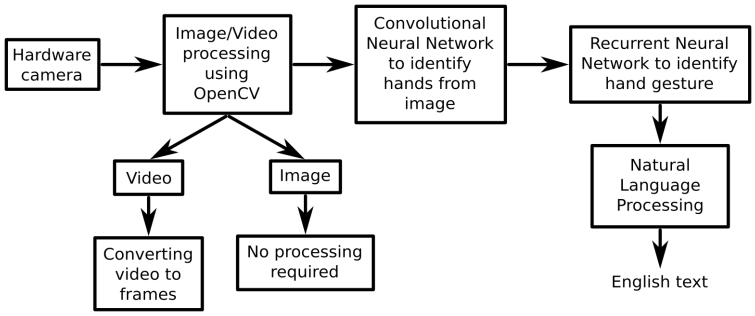

Figure 3: Architecture

Figure 3 gives a brief overview of the architecture that the Sign Language Recognition application follows.

Input module

A camera is required to record the entire conversation. Letters and digits can be considered as single frames, whereas words and sentences are recorded as videos.

Image and Video Processing

Image and video processing is done using Python and OpenCV. Videos are converted into a series of frames to simplify the process of recognizing gestures.

Convolutional Neural Network

These networks are a class of deep feedforward neural networks that are based on weight sharing. They are commonly used in image and video processing to identify manifolds. In this project, they are used to identify hands from the given frames.

Recurrent Neural Network

Recurrent Neural Networks are used whenever memory of previous computations is required. It is used to identify the hand gesture by considering the gestures denoted by the current and the previous frames.

Natural Language Processing

Natural Language Processing is used to convert the output of the Recurrent Neural Network into proper English text.

6.3 Data Flow Diagram

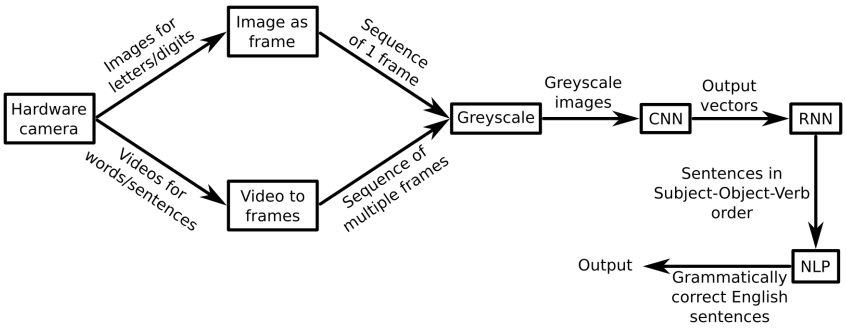

Figure 4: Data Flow Diagram

Figure 4 gives the interaction between the different modules of the application and how data flows between them.

Input

The input to the application is taken in the form of either images or videos. Images are considered as sequences of a single frame, whereas videos are converted into a series of frames using OpenCV.

These frames are converted into grayscale images to ensure uniformity in the training and test data in terms of the background, etc.

Convolutional Neural Network
This neural network is given grayscale images as input. It processes them and converts them to vectors that denote the positions of the hands in the image.

Recurrent Neural Network
This network takes in the input vectors and outputs letters/digits/words in the order of $subject - object - verb$.

Natural Language Processing module
NLP is required to convert this sentence in the form of $subject - object - verb$ into a proper English sentence.

7 Implementation

7.1 Technology Introduction

7.1.1 Python

A high level programming language for general purpose programming. It is an interpreted language with a syntax that allows programmers to express concepts in fewer lines of code.

7.1.2 OpenCV

Open Source Computer Vision is a library of programming functions mainly aimed at real-time computer vision. Here were using OpenCV for gesture recognition.

7.1.3 CUDA

A parallel computing platform and application programming interface (API) model created by Nvidia. It allows software developers and software engineers to use a CUDA-enabled graphics processing unit (GPU) for general purpose processing

7.1.4 PyTorch

An open source machine learning library for python, which is used for training and testing deep neural networks.

7.2 Overall view of the project in terms of implementation

Overall implementation can be divided into three parts, training, testing and saving the models.

Training
It is done in two phases. First, the CNN is trained to identify various features in each frame and then the RNN is trained on the outputs of CNN on continuous set of frames, to capture various features spread across time.

Testing

30% of the input data is set aside for testing. Both the training phases use the same output with the help of 2 linear fully connected layers that classify the output.

7.3 Explanation of Algorithm and how it is been implemented

Convolution neural network

Use a filter that runs across the input images along with maxpooling to identify specific features in images that help identify gestures.

Recurrent neural network

The CNN can only be used on one frame but gestures are spread across time. It is often better to use RNN that retains the previous inputs in memory.

Optimizer

Adam optimizer combines the features of AdaGrad and RMSProp to learn adaptively, and is currently known to be the best learning algorithm.

7.4 Information about the implementation of Modules

Input

CNN is trained on individual frames while RNN is trained on videos. A common wrapper is used to load frames as tensors in order for RNN and the same are randomly accessed for CNN.

Image and Video Processing

The frames are extracted from the video using OpenCV and are converted to grayscale, cropped, scaled and converted to tensors.

Neural network

A common model is developed and RNN can be optionally disabled while training CNN.

7.5 Conclusion

Use of python and inbuilt libraries has greatly simplified the implementation and training the neural networks for the purpose of this project was made faster with the help of CUDA enabled GPU.

8 Testing

8.1 Introduction

The testing code is written in python, using CUDA GPU as and when necessary for faster processing. OpenCV is used for gesture recognition. The machine

learning library Pytorch is used for training and testing the deep neural network. Notably we use torch.optim for implementing various optimization algorithms and torch.autograd to automatically differentiate native Torch code.

8.2 Testing Tools and Environment

8.2.1 Python

A high level programming language for general purpose programming. It is an interpreted language with a syntax that allows programmers to express concepts in fewer lines of code.

8.2.2 OpenCV

Open Source Computer Vision is a library of programming functions mainly aimed at real-time computer vision. Here were using OpenCV for gesture recognition.

8.2.3 CUDA

A parallel computing platform and application programming interface (API) model created by Nvidia. It allows software developers and software engineers to use a CUDA-enabled graphics processing unit (GPU) for general purpose processing

8.2.4 PyTorch

An open source machine learning library for python, which is used for training and testing deep neural networks.

8.3 Test cases

The videos are converted into a series of frames using OpenCV. A wrapper is created to load each frame from a given video. The frames from all the videos are loaded as tensors. A data-loader instance is created for each video so that torch's built-in features are used to load frames as batch. To improve training for the CNN we allow random access to any frame in any video.

9 Conclusion

Deep neural networks have revolutionized how different Machine Learning problems are approached. Using them to solve this problem can give significant results.

Image and video processing is one of the important fields in Machine Learning and Deep Learning at present.

The huge number of images and videos on social media and otherwise has led to

an increased interest in the field. The algorithms and techniques developed as a result of this can be used in more impactful fields like helping the speech and hearing impaired lead more convenient lives.

The hand signs along with the emotions displayed by the individual provide the content and tone of the sentence. A tool to convert their words into a widely used language enhances their ability to communicate.

Since such applications will always be necessary, research and development of new algorithms to improve the efficiency of interpretation of sign language will be endless.

Deep learning has a lot more to offer than what has already been explored. With such possibilities, there will always be scope to develop newer models that can be used in sign language recognition.

The models used in this can also be used in other applications such as gesture controlled systems, tracking unusual activity, etc. with slight tweaks. With slight modification, it can also be used for recognition of other sign languages.

Moreover, the current work can be further extended to output speech instead of text. A similar generative model can be used to map speech or text to actions by stitching together pre-recorded videos or actually generate an interface with a skeletal structure. This model can be seamlessly integrated with various applications on different platforms to support high productivity.

References

[1] Kusumika Krori Dutta, Satheesh Kumar Raju K, Anil Kumar G S, and Sunny Arokia Swamy B. Double handed indian sign language to speech and text. In *Proceedings of the 2015 Third International Conference on Image Information Processing (ICIIP)*, ICIIP '15, pages 374–377, Washington, DC, USA, 2015. IEEE Computer Society.

[2] K. Dabre and S. Dholay. Machine learning model for sign language interpretation using webcam images. In *2014 International Conference on Circuits, Systems, Communication and Information Technology Applications (CSCITA)*, pages 317–321, April 2014.

[3] K. Dixit and A. S. Jalal. Automatic indian sign language recognition system. In *2013 3rd IEEE International Advance Computing Conference (IACC)*, pages 883–887, Feb 2013.

Appendix

Screen snapshots

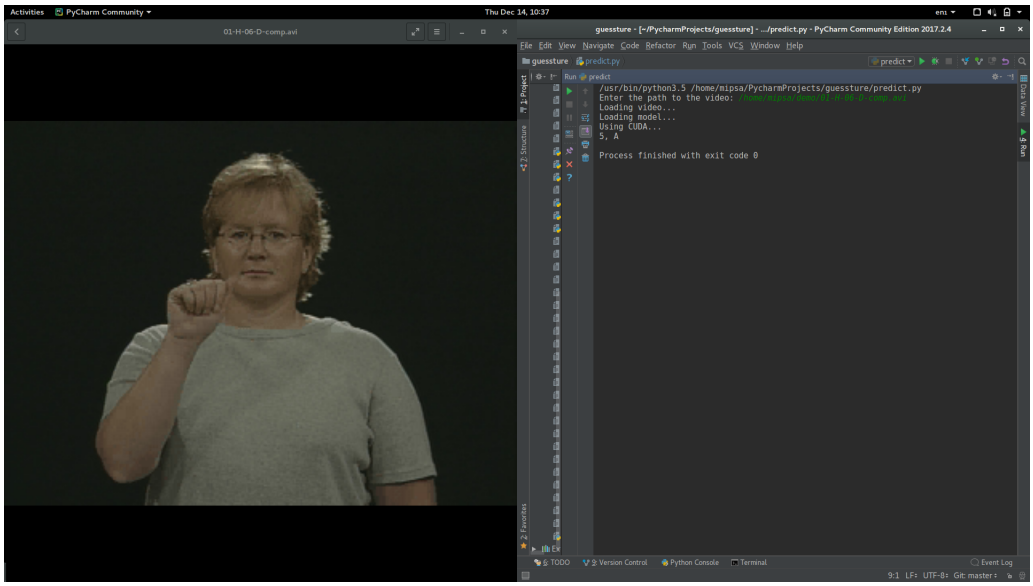

Figure 5: The symbol A

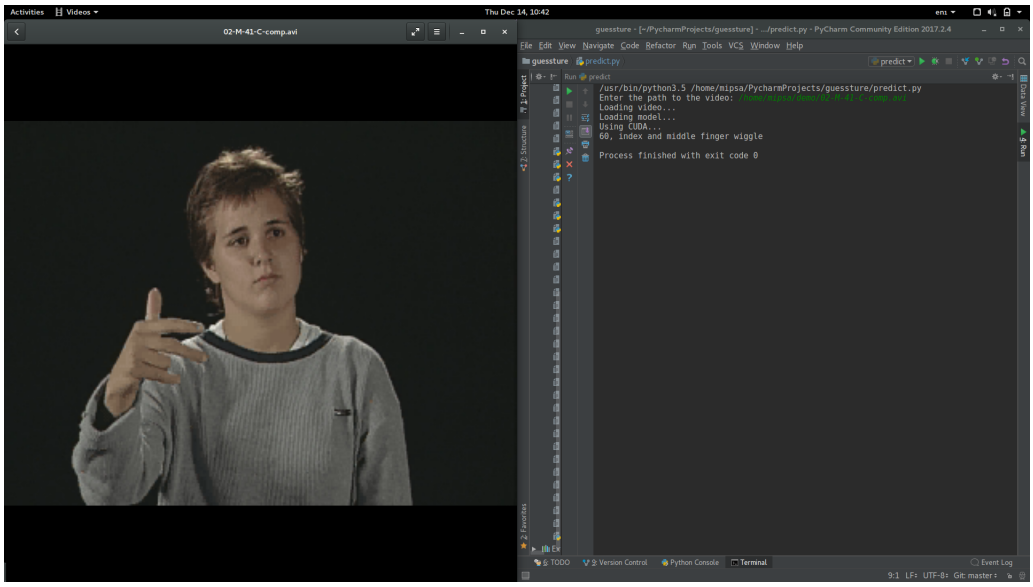

Figure 6: The movement of index and middle fingers