SIGN LANGUAGE RECOGNITION

A PROJECT REPORT

Submitted by

SUBHAM BANERJEE (20BAI10051)
RISHABH MATHUR (20BAI10060)
PRIYAM JAIN (20BAI10087)
YASHASWI PATEL (20BAI10327)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING WITH SPECIALIZATION IN

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

VIT BHOPAL UNIVERSITY

KOTHRIKALAN, SEHORE
MADHYA PRADESH - 466114

DECEMBER 2021

BONAFIDE CERTIFICATE

Certified that this project report titled **“ SIGN LANGUAGE RECOGNITION”**. is the bonafide work of “**SUBHAM BANERJEE** (20BAI10051), **RISHABH MATHUR** (20BAI10060), **PRIYAM JAIN** (20BAI10087), and **YASHASWI PATEL** (20BAI10327)” who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported at this time does not form part of any other project/research work based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROGRAM CHAIR
Dr S Sountharrajan, Associate Professor
School of Computer Science and Engineering
VIT BHOPAL UNIVERSITY

PROJECT GUIDE
Dr. G. Elangovan, Assistant Professor
School of Computer Science and Engineering
VIT BHOPAL UNIVERSITY

The Project Exhibition I Examination is held on **22nd December 2021.**

ACKNOWLEDGEMENT

First and foremost I would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

I wish to express my heartfelt gratitude to Dr S.Sountharrajan, Program Chair, School of Computer Science and Engineering for much of his valuable support and encouragement in carrying out this work.

I would like to thank my internal guide Dr. G. Elangovan, for continually guiding and actively participating in my project, giving valuable suggestions to complete the project work.

I would like to thank all the technical and teaching staff of the School of Computer Science and Engineering, who extended directly or indirectly all support.

Last, but not least, I am deeply indebted to my parents who have been the greatest support while I worked day and night for the project to make it a success.

LIST OF ABBREVIATIONS

SL	Sign Language
ISL	Indian Sign Language
CNN	Convolutional Neural Network
ROI	Region Of Interest
HMM	Hidden Markov Model
API	Application Programming Interface
SLR	Sign Language Recognition
MATLAB	MATrix LABoratory
ASL	American Sign Language
BSD	Berkeley Software Distribution
SVM	Support Vector Machine
HCI	Human – Computer Interaction
DNN	Deep Neural Networks
RGB	Red Green Blue
ONEIROS	Open-ended Neuro-Electronic Intelligent Robot Operating System
SDK	Software Development Kit
HDD	Hard Disk Drive
DIP	Digital Image Processing

LIST OF FIGURES AND GRAPHS

ABSTRACT

Speech impaired people use hand signs and gestures to communicate. Sign Language (SL) can be thought of as a tiny collection of motions or touches done with the fingers, hands, arms, eyes, head, and face, among other things. Each SL touch is associated with a distinct meaning. Understanding SL is nothing more than comprehending the significance of these symbols. When someone who is fully reliant on this SL of communication act attempts to communicate with someone who does not comprehend SL, there is a communication difficulty. Each country has its own sophisticated SL. In India, the language is known as Indian Sign Language (ISL). In this paper, an innovative sign language recognition algorithm is suggested to recognise the letters of the alphabet and to interact with sign language. We can identify signals and offer suitable text output using computer recognition and sensory networks. The technology transforms touch based on ISL into English. Among the translated symbols are a few numbers, words, and phrases. The algorithm first performs data acquisition, and then pre-touch analysis is performed to track hand movements using a combination. Algorithm, and recognition are performed using simulations. The project itself has created 10,000 total images, of which 700 images are used to build the program and to test the performance. The accuracy of the program is up to 98.73%.

TABLE OF CONTENTS

1.PROJECT DESCRIPTION AND OUTLINE

1.1. INTRODUCTION

Gesture as defined by the dictionary means ‘Motion of limbs or body made to express thought or to emphasize speech’. Gestures are meaningful maneuvers that involve physical motion of different body parts such as hands, fingers, arms, head, neck, eyes etc. Gestures convey meaningful information. In order to extract information from the performed gestures a gesture recognition system can be implemented. Gesture recognition is the process by which gestures made by the user are recognized by the receiver.

Sign language, as a comprehensive topic, contains different components that may be studied collectively or individually in combination with computer engineering, particularly within the framework of. There are different approaches and studies that deal with facial recognition and body motions, which is owing to the fact that computers and facial recognition can play a significant part in assisting deaf disabled people to connect with intelligent devices.

1.2 MOTIVATION FOR THE PROJECT

According to India's 2011 census, 1.3 million persons have a "hearing issue." In India, the National Association of the Deaf estimates that 18 million people, or about 1% of the population, are deaf. These figures sparked the idea for our project. Rather than a programme, these speech impaired and deaf persons require an appropriate path to communicate with ordinary people. Not everyone can comprehend sign language used by people with impairments. As a result, the goal of our project is to convert touch movements into understandable text for the broader audience.

1

1.3. PROBLEM STATEMENT

People who have hearing impairment face a lot of challenges. Public address systems notify us of what's going on, but fail to do so when it comes to informing individuals with hearing impairment. They also find it difficult to convey their messages to people in most of the scenarios.

Even today there is unavailability of translators in many areas. There is a lack of innovation in technology for people with hearing impairment.

A software which can translate sign language with high accuracy can do wonders in the lives of such people.

1.4. OBJECTIVE OF THE WORK

I. The aim of the project is to create a Sign Language Recognition Application which helpful in every field for the deaf people eliminating the need of a human translator which can be used in any location and in any situation.

II. Application should improve itself with the help of Machine Learning and can increase its efficiency and accuracy.

III. To generate a large amount of appropriate dataset and apply appropriate image pre-processing techniques in order to remove the noise and obtain the ROI.

IV. To design the model and architecture for CNN to train the pre-processed images and achieve the maximum possible accuracy.

V. To develop an algorithm to predict the gesture in real time.

1.5 SUMMARY

People with hearing impairment have been facing difficulties for a very long time. It has been estimated that almost 18 million people or roughly 1 percent of the population of India are deaf. The technological advancements are not able to keep up with their requirements. There has been negligible development in ISL Dataset. The program will eliminate the need of a physical translator and can be integrated with various software for easy communication of people with hearing impairments. The software uses various extended data sets and is aided by machine learning.

2. OVERVIEW OF LITERARY WORKS

2.1. INTRODUCTION

The domain study that we performed for the project was primarily concerned with comprehending neural networks.

2.2. EXISTING SYSTEM

We went through additional similar attempts that are implemented in the realm of sign language recognition in the Literature survey. The following are descriptions of each of the project works.

2.2.1. A Survey of Hand Gesture Recognition Methods in Sign Language Recognition

For many years, researchers have been studying the Sign Language Recognition (SLR) system, which is required to recognise sign languages. The investigations are based on various input sensors, gesture segmentation, feature extraction, and classification algorithms. The purpose of this work is to analyse and compare the methods used in SLR systems, classification methods that have been used, and to suggest the most promising way for future research. Because of recent advancements in classification methods, many of the recently presented works primarily contribute to classification methods, such as the hybrid approach and Deep Learning. This research focuses on the categorization approaches utilised in previous Sign Language Recognition systems. According to our review, HMM-based techniques, including modifications, have been thoroughly researched in previous research.

This research is based on multiple input sensors, gesture segmentation, feature extraction, and classification approaches. The purpose of this work is to analyse and compare the methods used in SLR systems, classification methods that have been used, and to recommend the most trustworthy way for future research. Because of recent advancements in classification methods, many of the recently presented works, such as the hybrid approach and Deep Learning, primarily contribute to classification methods. According to our assessment, HMM-based techniques, including modifications, have been extensively researched in the past. Hybrid CNN-HMM and completely Deep Learning techniques have yielded encouraging results and provide avenues for future research.

2.2.2. Interaction between Deaf-Dumb and Hearing People

Chat programmes have evolved into a powerful medium for individuals to converse in different languages with one another. There are several chat software that are used by different individuals in different languages, but there isn't one that allows people to interact using sign languages. The technique developed is based on the Sinhala Sign language. Text messages are transformed to sign messages, voice messages are converted to sign messages, sign messages are converted to text messages, and voice messages are converted to sign messages. Speech character recognition for voice messages has been developed using Google's voice recognition API. The system was trained for voice and text patterns by employing some text parameters and Sinhala Sign language signs, which are shown using emoji. The emoji and signs incorporated in this system will bring ordinary people closer to disabled persons. This is a two-way communication system, however it relies on gesture recognition patterns, which are not always accurate.

2.2.3. A Strategy for Identifying Indian Sign Language for Deaf Individuals Utilizing Otsu's Algorithm

In this work, we offered various strategies for making it easier for people to recognise signs when communicating. And the outcome of those symbols signs will be transformed into text. In this project, we capture hand motions using a camera and transform the image to grayscale. The Otsu algorithm is used to segment a grayscale image of a hand gesture. The total image level is separated into two classes: hand and background. The optimal threshold value is calculated by dividing the class variance by the total class variance. The Canny edge detection technique is used to determine the boundary of a hand motion in a picture. We employed edge-based segmentation and threshold-based segmentation in Canny edge detection. Then, due to its simplicity and stability, Otsu's algorithm is utilized. When the worldwide distribution of the target and background change greatly, this algorithm fails.

2.2.4. Hand Gesture Recognition based on Digital Image Processing using MATLAB

This study shows a prototype system that assists normal people in recognising hand gestures in order to interact more successfully with special persons. The aforementioned research study focuses on the topic of real-time gesture identification in sign language utilised by the deaf community. The solution is based on D.I.P. techniques such as Color Segmentation, Skin Detection, Image Segmentation, Image Filtering, and Template Matching. This system recognises ASL motions, as well as the alphabet and a subset of its words.

2.3. PROPOSED SYSTEM

Our suggested system is a convolution neural network-based sign language recognition system that recognises diverse hand movements by recording video and processing it into frames. The hand pixels are then fragmented, and the resulting image is forwarded for comparison to the trained model. As a corollary, our system seems to be more capable of obtaining precise text labels for letters.

2.3.1 Sign Language Recognition Training Model Flowchart

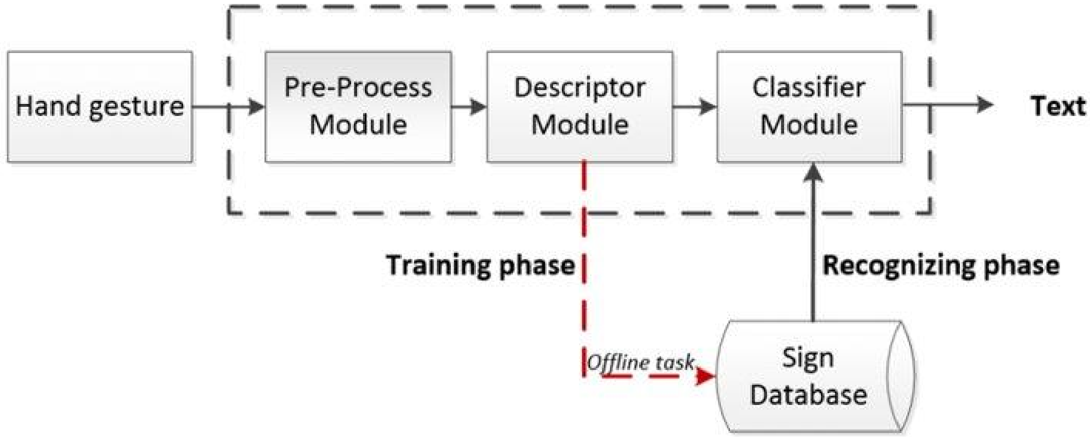


Fig 1: Sign Language Recognition Training Model Flowchart

2.4. SUMMARY

For many years, researchers have been studying the Sign Language Recognition (SLR) system, which is required to recognise sign languages.

The first approach's investigations are based on various input sensors, gesture segmentation, feature extraction, and classification algorithms. The purpose of this work is to analyse and compare the methods used in SLR systems, classification methods that have been used, and to recommend the most trustworthy way for future research.

The second approach's technique developed is based on the Sinhala Sign language. The emoji and signs incorporated in this system will bring ordinary people closer to disabled persons. This is a two-way communication system, however it relies on gesture recognition patterns, which are not always accurate.

In the third approach, hand motions were captured using a camera and transformed the image to grayscale. The Canny edge detection technique is used to determine the boundary of a hand motion in a picture.

Our suggested project is CNN based SLR system which seems to be more capable of obtaining precise text labels for letters.

3. REQUIREMENT ARTIFACTS

3.1. INTRODUCTION

The subject investigation that we implemented for the research was essentially concerned with fully comprehending neural network models.

3.2. SYSTEM CONFIGURATION

3.2.1. Software Requirements:-

OPERATING SYSTEM: Windows, Mac, Linux

SDK: OpenCV, TensorFlow, Keras ,Numpy

3.2.2. Hardware Requirements:-

CAMERA: Good quality, 3MP

RAM: Minimum 8GB or higher

GPU: 4GB dedicated

PROCESSOR: Intel Pentium 5 or higher

HDD: 10GB or higher

MONITOR: 15” or 17” colour monitor

MOUSE: Scroll or Optical Mouse or Touchpad

KEYBOARD: Standard 110 keys keyboard

3.3. MODULE SPLIT UP

3.3.1. TensorFlow

TensorFlow is a platform - independent software library for dataflow and differentiable programming that may be used for a wide range of purposes. It's a symbolic math library that's also used in applications of machine learning like neural networks. It is utilised by Google for both development and technology.

FEATURES: TensorFlow supports stable Python (for version 3.7 across all platforms) and C APIs, as well as C++, Go, Java, JavaScript, and Swift APIs with no promise of API backwards compatibility (early release). There are third-party packages for C#, Haskell Julia, MATLAB, R, Scala, Rust, OCaml, and Crystal. On top of the C API, new language support should be created. However, not all functionality is presently accessible in C. The Python API adds some extra capabilities.

APPLICATIONS: Automated image captioning software, such as DeepDream, is one of the applications for which TensorFlow serves as the foundation.

3.3.2. OpenCV

OpenCV (Open Source Computer Vision Library) is a programming function library geared mostly at real-time computer vision. It was created by Intel and was later sponsored by Willow Garage and Itseez (which was later acquired by Intel). The library is cross-platform and available for free under the BSD open-source licence.

OpenCV's application area include: -

- ➔ 2D and 3D feature toolkits
- ➔ Egomotion estimation
- ➔ Facial recognition system
- ➔ Gesture recognition
- ➔ Human–computer interaction (HCI)
- ➔ Mobile robotics
- ➔ Motion understanding
- ➔ Object identification
- ➔ Segmentation and recognition

OpenCV has a statistical machine learning library that contains functions to help with some of the aforementioned topics.:-

- ➔ Boosting
- ➔ Decision tree learning
- ➔ Gradient boosting trees
- ➔ Expectation-maximization algorithm
- ➔ k-nearest neighbor algorithm
- ➔ Naive Bayes classifier
- ➔ Artificial neural networks
- ➔ Random forest
- ➔ Support vector machine (SVM)
- ➔ Deep neural networks (DNN)

3.3.3. Keras

Keras is a Python-based open-source neural network library. It can run on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML. It is user-friendly, modular, and expandable, and was designed to allow rapid experimentation with deep neural networks. It was created as part of the project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System) research effort, and its principal inventor and maintainer is François Chollet, a Google engineer. Chollet is also the creator of the XCeption deep neural network model.

FEATURES: Keras contains numerous implementations of commonly used neural network building blocks such as layers, objectives, activation functions, optimizers, anda host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code. Keras supports convolutional and recurrent neural networks in addition to regular neural networks. Other popular utility layers supported include dropout, batch normalisation, and pooling.

2.3.4. Numpy

NumPy is a Python library that adds support for massive, multidimensional arrays and matrices, as well as a vast number of high-level mathematical functions to operate on these arrays.

FEATURES: NumPy is designed to work with Python's CPython reference implementation, which is a non-optimizing bytecode interpreter. Algorithms written for this version of Python are frequently much slower than compiled equivalents. NumPy tackles the slowness issue in part by offering multidimensional arrays as well as functions and operators that operate effectively on arrays, which necessitates rewriting some code, primarily inner loops, in NumPy. Because they are both interpreted, NumPy in Python provides functionality comparable to MATLAB, and they both allow the user to construct fast programs as long as most operations work on arrays or matrices rather than scalars. In comparison, MATLAB has a plethora of extra toolboxes, most notably Simulink, but NumPy is inextricably linked with Python, a more current and comprehensive programming language. Additionally, there are additional Python programmes available; SciPy is a library that adds more MATLAB-like capability, and Matplotlib is a plotting tool that gives MATLAB-like plotting functionality. Both MATLAB and NumPy internally rely on BLAS and LAPACK for efficient linear algebra processing.

LIMITATIONS: Implanting or adding entries to an array is more difficult than it is with Python's lists. To expand arrays, the np.pad(...) procedure produces new arrays with the specified form and padding values, copies the given array into the new one, and returns it. The np.concatenate([a1,a2]) action in NumPy does not truly link the two arrays, but instead returns a new one that contains the elements from both given arrays in order. Reshaping the dimensionality of an array using np.reshape(...) is only possible if the array's element count does not change. The fact that NumPy arrays must be views on contiguous memory buffers causes these conditions.

3.3.5 Neural Networks

A neural network is a set of algorithms that attempts to recognise underlying relationships in a batch of data using a technique similar to how the human brain works. In this context, neural networks are systems of neurons that might be organic or artificial in nature. Because neural networks can adapt to changing input, they can produce the best possible results without having to rethink the output criteria. The neural network concept, which has its roots in artificial intelligence, is quickly gaining traction in the creation of trading systems.

A neural network functions in the same way as the neural network in the human brain does. A "neuron" in a neural network is a mathematical function that collects and categorises data based on a predefined architecture. The network is quite similar to statistical approaches like curve fitting and regression analysis.

4. DESIGN METHODOLOGY AND ITS NOVELTY

4.1. METHODOLOGY

4.1.1. Data Collection

The first step is the data-set collection. We collect the full frame (RGB) image of the corresponding depth map from the camera.

4.1.2 Training module

SUPERVISED MACHINE LEARNING : Supervised machine learning is a type of machine learning in which the model is trained using input data and predicted output data. To develop such a model, the following stages must be completed:

1. Model Creation
2. Model Development
3. Model Validation

4. Assessment of the Model

4.1.2.1. MODEL CONSTRUCTION:

It is determined by machine learning algorithms. It was neural networks in this project's situation.

We start with its object, which is made up of layers of different sorts.

The model is compiled after a sufficient number of layers have been added.

Keras is now communicating with TensorFlow in order to build the model. It is critical to write a loss function and an optimizer algorithm during model compilation. The loss function displays the model's forecast accuracy for each prediction.

It is necessary to scale data prior to model training in order to use it later.

4.1.2.2. MODEL TRAINING:

It's time to train the model after it's been built. The model is trained in this phase utilising training data and expected output for this data. When the script runs, the progress is displayed on the console. Finally, it will give the model's final accuracy.

4.1.2.3. MODEL TESTING:

A second set of data is loaded during this phase. Because the model has never seen this data set before, its genuine correctness will be validated. After the model training is finished and it is determined that the model produces the desired results, it can be saved. Finally, the model that has been saved can be used in the actual world. This step is known as model evaluation. This means that the model can be applied to new data.

4.2. NOVELTY

- ➔ This project will help Deaf People to Communicate with People in India
- ➔ It incorporate various kind of Indian Hand Signs
- ➔ Can add various kind of personalized sign common in a particular area
- ➔ This project has a diverse kind of hand sign which are not available in a typical ISL structure making it more innovative that previous models

4.3. SYSTEM ARCHITECTURE DIAGRAM

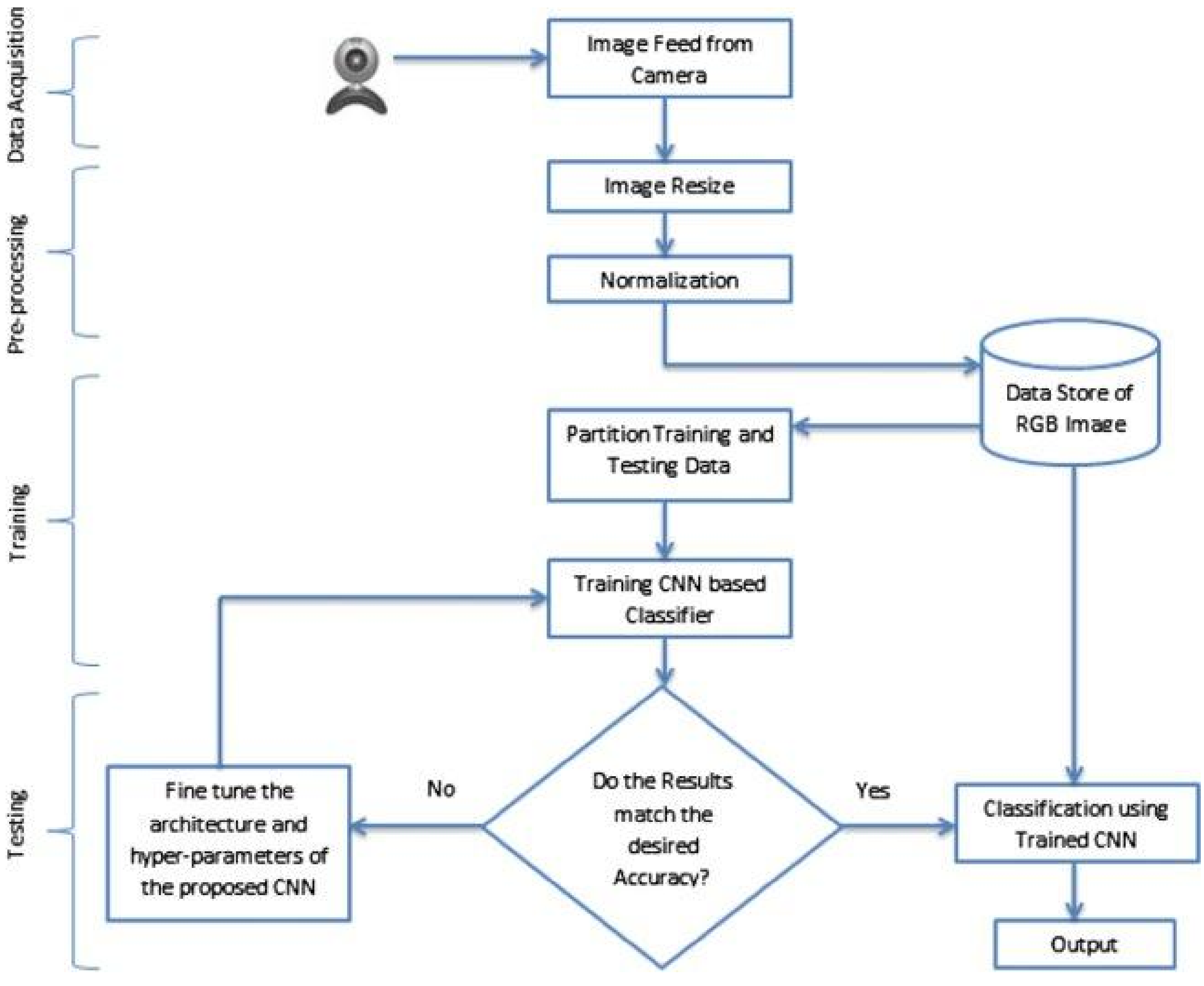


Fig 2: System Architecture Diagram

Chapter 5.

TECHNICAL IMPLEMENTATION & ANALYSIS

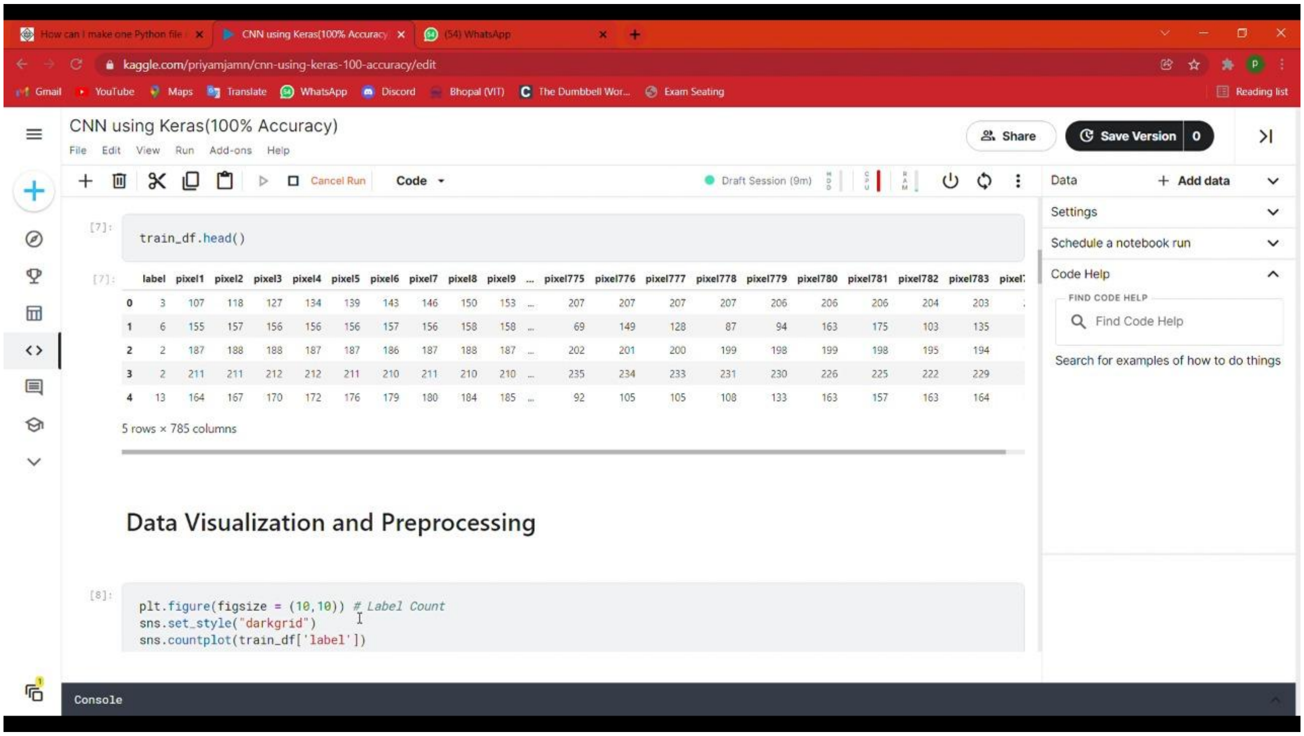
5.1. Outline

Our eye movement tracking experiment was designed to test the responses of deaf viewers to a wide range of sign language movements and gestures and to investigate viewing patterns that might be exploited in the design of optimised video communication systems. Our results demonstrate that the most important region of the sign language video image is the face of the signer. This is particularly evident in the results obtained for video clip one where the signer is closer to the camera than in the other video clips. Fixations are mainly on the upper face region with no visual excursions to the distractor objects in the background. Gaze is more on the lower face region for video clip two where the signer is further from the camera and the face region is therefore smaller. Participants were found not to follow the movements of

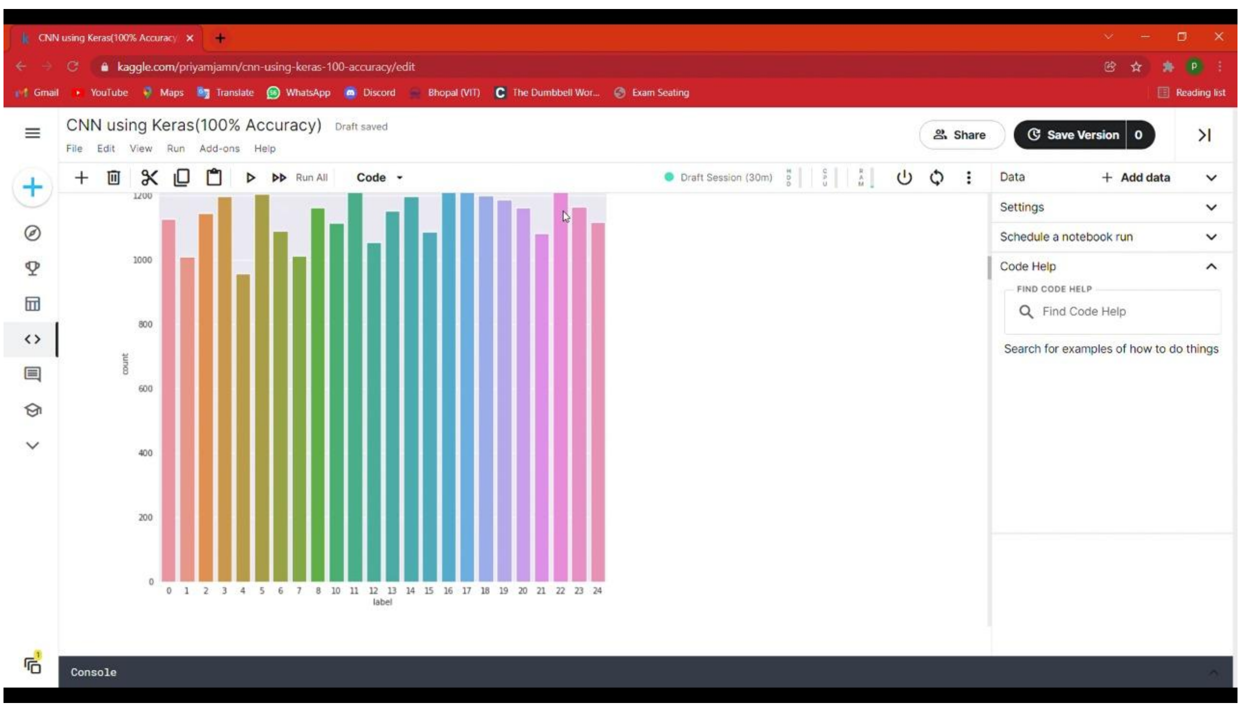
Perception of Sign Language 23 the hands or detailed movements of the fingers during periods of finger spelling, suggesting that sign information was observed in peripheral (lower resolution) vision. Short excursions to the hands were noted only when the hands of the signer were close to the face. The hands were close enough for the face to remain in foveal (high resolution) vision. The wider, more rapid gestures and movements of the signer in video clip three seemed to cause gaze to fall more on the upper body region of the signer for some viewers. There was no statistically significant difference in the patterns of viewing behaviour across the three videos tested, as determined by the Friedman Test (in section 3.1.1) of this paper. This leads us to conclude that the same viewing strategies are applied by viewers to different aspects of sign language video regardless of the background, distance of the signer from the camera and movement of the signer around the scene

5.6. Performance Analysis (Graphs / Charts)

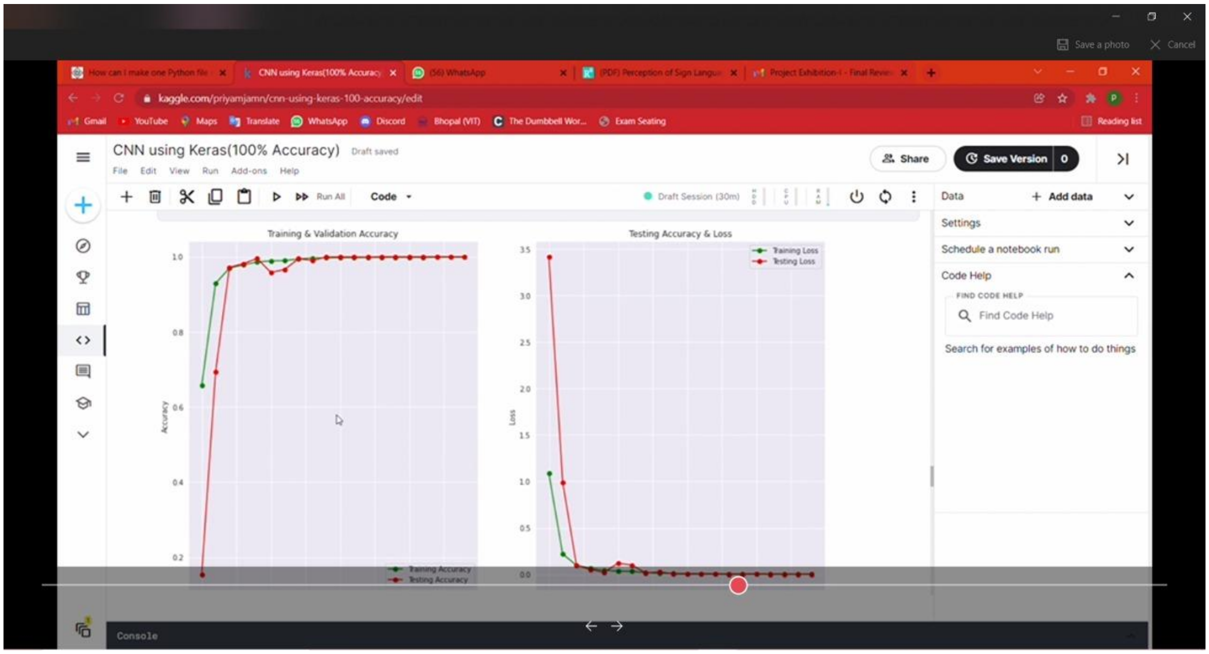
A. Fig 3: Data Verification



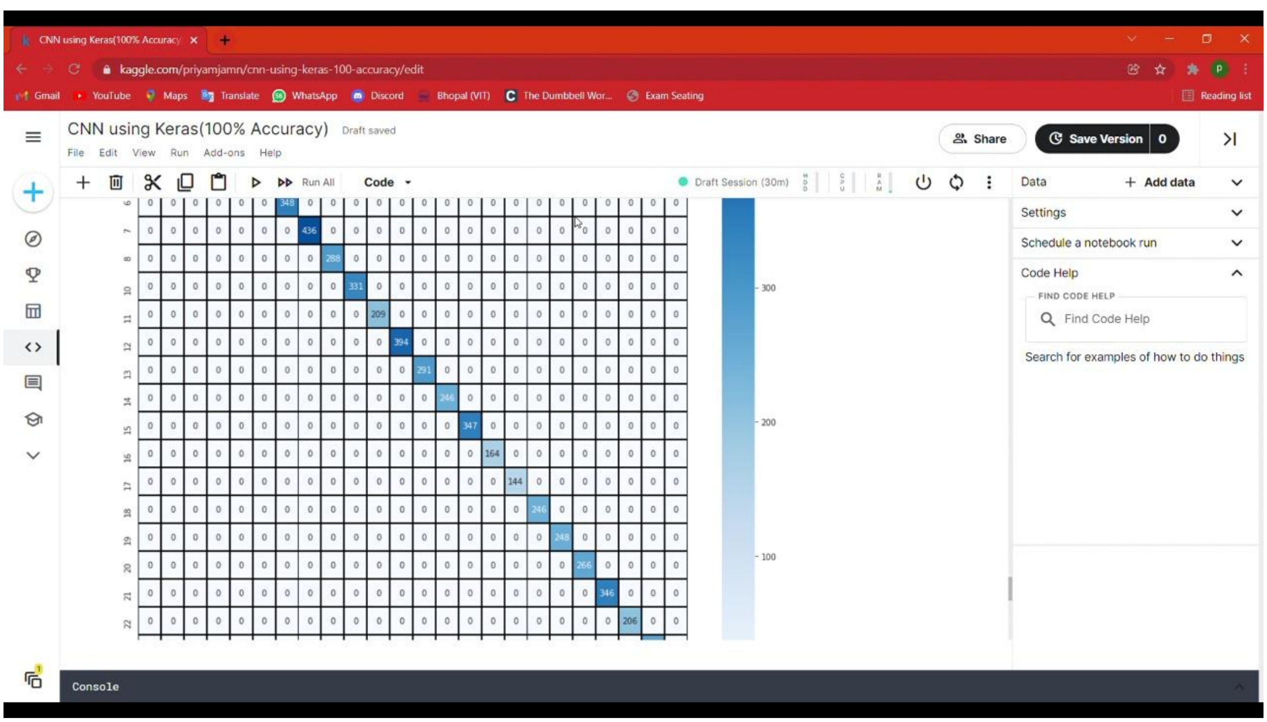
B. Fig4: Pre-Processing Data Visualization



C. Fig 5: Training Accuracy



D. Fig 6: Graph Accuracy



CHAPTER - 6:
PROJECT OUTCOME AND APPLICABILITY

6.1 Outline

The training accuracy attained using the picture dataset without any augmentation was quite good (about 99 percent), however the real-time performance was not up to par. Because hand-gestures were not positioned exactly in the middle and aligned vertically in real time, it was predicting inaccurately most of the time. We trained our model by supplementing our dataset to solve this limitation. Although the training accuracy was dropped to 89%, the real-time predictions were mostly accurate.

Also, whilst we grow our schooling photo length the accuracy betters itself each at some stage in schooling and actual time implementation (because the variety of parameters to gain knowledge increases). However, schooling snap shots with massive length is without delay proportional to the computational energy of the system.

6.1 Key Implementations Outlines of the System

a. The training accuracy attained using the picture dataset without any augmentation was quite good (about 99 percent), however the real-time performance was not up to par

b.We trained our model by supplementing our dataset to solve this limitation. Although the training accuracy was dropped to 89%, the real-time predictions were mostly accurate.

6.2 Project Applicability on Real-World Applications

People who communicate using sign language are deprived of modern day tech like video conferencing. Since most of these applications transition window focus to those who speak aloud, it makes it difficult for signers to "get the flow " so they can communicate easily and effectively. Enhanced real time sign language detection is challenging since application need to perform classification using high volume field as the input which makes the task computationally heavy . Therefore optimization in such program is necessary. We tried to reduce the dimensionality and to create a light weight model that consume fewer CPU cycles by isolating the information the model needs from the videos in order to perform classification of every frame. We present a real time sign language detection model and demonstrate how it can be used to provide a mechanism to identify the person signing as the active speaker. We are proud of what we have achieved but it's only just the beginning . With a bigger dataset and more tweaking of our model , we believe we could develop accurate and reliable technology for signers.

CHAPTER-7:

CONCLUSIONS AND RECOMMENDATION

7.1. Outline

The goal of this project is to predict ISL alphanumeric hand gestures in real time. Previous work shows that it can be solved more precisely if we look at segmented RGB hand gestures. By using depth-based targeting, we eliminate dynamic background overloads. We were able to achieve a training accuracy of 89.30% and a test accuracy of 98.5%. Our model showed good accuracy in predicting the results.

7.2. Limitation/Constraints of the System

The training accuracy attained using the picture dataset without any augmentation was quite good (about 99 percent), however the real-time performance was not up to par. Because hand-gestures were not positioned exactly in the middle and aligned vertically in real time, it was predicting inaccurately most of the time. We trained our model by supplementing our dataset to solve this limitation. Although the training accuracy was dropped to 89%, the real-time predictions were mostly accurate.

7.3. Future Enhancements

The growth of the schooling photo length and accuracy betters itself at some stage in schooling and actual time implementation (because the variety of parameters to gain knowledge increases). However, schooling snap shots with massive length is without delay proportional to the computational energy of the system.

7.4. Inference

Due to the limited computational power of our laptop we were able to compute the result with accuracy of 89% but a person with a better performing engine can yield better results.

REFERENCES

1. 1. Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., Bengio, Y.: Theano: a CPU and GPU math expression compiler. In: Proceedings of the Python for Scientific Computing Conference (SciPy), June 2010, oral Presentation
2. Chai, X., Li, G., Lin, Y., Xu, Z., Tang, Y., Chen, X., Zhou, M.: Sign Language Recognition and Translation with Kinect (2013). Language Recognition and Translation with Kinect.pdf. http://vipl.ict.ac.cn/sites/default/files/papers/files/2013 FG xjchai Sign

3. Cire¸san, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3642–3649. IEEE (2012)
4. Baum, L. 1972. An inequality and associated maximization technique in statistical estimation of probabilistic functions of Markov processes. Inequalities 3:1-8.
5. Charayaphan, C., and Marble, A. 1992. Image processing system for interpreting motion in American Sign Language. Journal of Biomedical Engineering 14:419-425.
6. https://www.skyfilabs.com/project-ideas/sign-language-translator-using-python
7. https://data-flair.training/blogs/sign-language-recognition-python-ml-opencv
8. https://cse.iitk.ac.in/users/cs365/2015/_submissions/vinsam/report.pdf

Book Name:

1.Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning Series.

2.The Elements of Statistical Learning: Data Mining, Inference and Prediction.