A MAJOR PROJECT REPORT

ON

REAL TIME SIGN LANGUAGE RECOGNITION USING TRANSFER LEARNING

Submitted in partial fulfillment of the requirement

for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

BY

G Harsha Vardhan Rao **19P61A0575**

Under the esteemed guidance of

Mr. G. Anil kumar

Associate Professor

Dept. of CSE

JUNE - 2023

Counselling Code : VBIT

VIGNANA BHARATHI Institute of Technology

(A UGC Autonomous Institution, Approved by AICTE, Accredited by NBA & NAAC-A Grade, Affiliated to JNTUH)

Aushapur (V), Ghatkesar (M), Hyderabad, Medchal – Dist, Telangana – 501 301.

DEPARTMENT
OF
COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

*This is to certify that the major project titled "**REAL TIME SIGN LANGUAGE RECOGNITION USING TRANSFER LEARNING** "submitted by **G Harsha Vardhan Rao(19P61A0575)** in B.tech IV-II semester Computer Science & Engineering is a record of the bonafide work carried out by them*

The Design embodied in this report have not been submitted to any other University for the award of any degree

INTERNAL GUIDE
Mr. G. Anil Kumar
(Associate Professor)

HEAD OF THE DEPARTMENT
Dr. M. Venkateswarao Rao

EXTERNAL EXAMINER

DECLARATION

I, **G Harsha vardhan rao** bearing hall ticket number **19P61A0575** hereby declare that the major project report entitled "**REAL TIME SIGN LANGUAGE RECOGNITION USING TRANSFER LEARNING**" under the guidance of **Mr. G. Anil kumar,** Associate Professor, Department of Computer Science and Engineering**, Vignana Bharathi Institute of Technology, Hyderabad**, have submitted to Jawaharlal Nehru Technological University Hyderabad, Kukatpally, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science And Engineering.

This is a record of bonafide work carried out by us and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

G HARSHA VARDHAN RAO (19P61A0575)

ACKNOWLEDGEMENT

I'm extremely thankful to our beloved Chairman **Dr. N. Goutham Rao** who took keen interest to provide us the infrastructural facilities for carrying out the project work.

Self-confidence, hard work, commitment and planning are essential to carry out any task. Possessing these qualities is sheer waste, if an opportunity does not exist. So, I whole-heartedly thank **Dr. P. V. S. Srinivas, Principal**, and **Dr. G. Sreeram**, Head of the Department, Computer Science and Engineering for their encouragement and support and guidance in carrying out the project.

I would like to express our indebtedness to the project coordinator, **Dr. G. Sreeram**, Associate Professor, Department of CSE, **Mr. G. Srikanth Reddy**, Associate Professor, Department of CSE, **Mrs. P. Subhadra**, Associate Professor, Department of CSE, for her valuable guidance during the course of project work.

I thank our Project Guide, **Mr. G. Anil kumar,** Associate Professor, for providing us with an excellent project and guiding us in completing our mini project successfully.

I would like to express our sincere thanks to all the staff of Computer Science and Engineering, VBIT, for their kind cooperation and timely help during the course of our project. Finally, I would like to thank our parents and friends who have always stood by us whenever I were in need of them.

Counselling Code : VBIT

VIGNANA BHARATHI Institute of Technology

(A UGC Autonomous Institution, Approved by AICTE, Accredited by NBA & NAAC-A Grade, Affiliated to JNTUH)

Aushapur(V), Ghatkesar(M), Hyderabad, Medchal – Dist, Telangana – 501 301.

DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING

VISION

To become, a Center for Excellence in Computer Science and Engineering with a focused Research, Innovation through Skill Development and Social Responsibility.

MISSION

M-1: Provide a rigorous theoretical and practical framework across state-of-the-art infrastructure with an emphasis on software development.

M-2: Impart the skills necessary to amplify the pedagogy to grow technically and to meet interdisciplinary needs with collaborations.

M-3: Inculcate the habit of attaining the professional knowledge, firm ethical values, innovative research abilities and societal needs.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO-01: Domain Knowledge: Synthesize mathematics, science, engineering fundamentals, pragmatic programming concepts to formulate and solve engineering problems using prevalent and prominent software.

PEO-02: Professional Employment: Succeed at entry- level engineering positions in the software industries and government agencies.

PEO-03: Higher Degree: Succeed in the pursuit of higher degree in engineering or other by applying mathematics, science, and engineering fundamentals.

PEO-04: Engineering Citizenship: Communicate and work effectively on team-based engineering projects and practice the ethics of the profession, consistent with a sense of social responsibility.

PEO-05: Lifelong Learning: Recognize the significance of independent learning to become experts in chosen fields and broaden professional knowledge.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO-01: Ability to explore emerging technologies in the field of computer science and engineering.
PSO-02: Ability to apply different algorithms indifferent domains to create innovative products.
PSO-03: Ability to gain knowledge to work on various platforms to develop useful and secured applications to the society.
PSO-04: Ability to apply the intelligence of system architecture and organization in designing the new era of computing environment.

PROGRAM OUTCOMES (POs)

Engineering graduates will be able to:

PO-01: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO-02: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO-03: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and cultural, societal, and environmental considerations.

PO-04: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO-05: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO-06: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO-07: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO-08: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO-09: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO-10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO-11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

ABSTRACT

Humans depend heavily on communication since it gives us the ability to express ourselves. Speaking is one of the most prevalent ways we communicate, along with body language, gestures, reading, writing, and visual assistance. Sadly, there remains a communication gap for the minority who are speech and hearing impaired. Deaf and mute persons can learn sign language, which is typically unknown to hearing people. It occurs to us that communication should be made simpler in order to eliminate the communication gap between hearing-impaired people and the general population.

To get over this obstacle, we suggest a technique in which sign language movements are gathered using a webcam and trained a Tensor Flow model using transfer learning to construct a Real-time Sign Language Recognition system. Many individuals will benefit from this when trying to communicate with the deaf and the mute.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER – 1

1. INTRODUCTION

1.1. INTRODUCTION TO THE SYSTEM

Sign languages are one of the means of communication through body movements especially of the hands and arms used when spoken type of communication is not possible. It has become the core form of communication for the communities of deaf and mute people. Hence to overcome this barrier of communication among spoken people and sign language users. We have proposed our model which uses a popular deep learning technique called “Transfer Learning” .

Dumb and deaf persons experience difficulties connecting with computers in the workplace because they cannot hear them. It is also risky to travel places alone since they cannot hear cars, bikes, or other people approaching. They can't immediately adapt to their surroundings or respond to other people, and expressing oneself is difficult. Sign language has a long history in western societies as a visual language or technique of communication, dating back to the 17th century. Traditional gestures, mimics, hand signs, and figure spelling, as well as the use of hand position to represent letters of the alphabet, make up sign language. A sign can also represent an entire thought or statement. The major goal is to deliver speech and text output for deaf persons utilising hand gesture sign language without the use of any sensors in a smart method.

Speech impaired people use hand signs and gestures to communicate. Normal people face difficulty in understanding their language. Hence there is a need of a system which recognizes the different signs, gestures and conveys the information to the normal people. It bridges the gap between physically challenged people and normal people.

Image Processing:

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too. Image processing basically includes the following three steps:

• Importing the image via image acquisition tools.

• Analysing and manipulating the image.

• Output in which result can be altered image or report that is based on image analysis. There are two types of methods used for image processing namely, analogue and digital image processing. Analogue image processing can be used for the hard copies like printouts and photographs. Image

analysts use various fundamentals of interpretation while using these visual techniques. Digital image processing techniques help in manipulation of the digital images by using computers. The three general phases that all types of data have to undergo while using digital technique are preprocessing, enhancement, and display, information extraction.

Digital Image Processing:

Digital image processing consists of the manipulation of images using digital computers. Its use has been increasing exponentially in the last decades. Its applications range from medicine to entertainment, passing by geological processing and remote sensing. Multimedia systems, one of the pillars of the modern information society, rely heavily on digital image processing.

Digital image processing consists of the manipulation of those finite precision numbers. The processing of digital images can be divided into several classes: image enhancement, image restoration, image analysis, and image compression. In image enhancement, an image is manipulated, mostly by heuristic techniques, so that a human viewer can extract useful information from it.

Digital image processing is to process images by computer. Digital image processing can be defined as subjecting a numerical representation of an object to a series of operations in order to obtain a desired result. Digital image processing consists of the conversion of a physical image into a corresponding digital image and the extraction of significant information from the digital image by applying various algorithms.

Pattern recognition: On the basis of image processing, it is necessary to separate objects from images by pattern recognition technology, then to identify and classify these objects through technologies provided by statistical decision theory. Under the conditions that an image includes several objects, the pattern recognition consists of three phases, as shown in Fig.

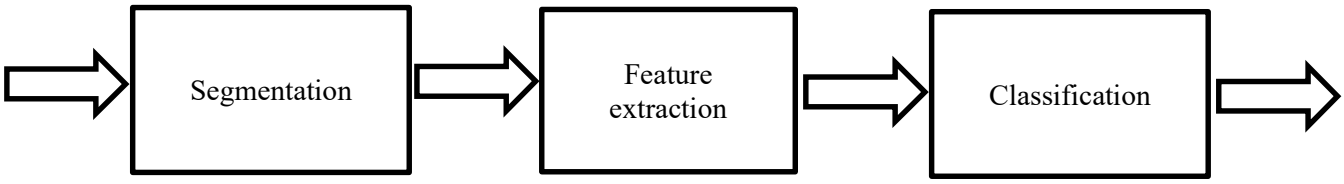


Fig1.1: Phases of pattern recognition

The first phase includes the image segmentation and object separation. In this phase, different objects are detected and separate from other background. The second phase is the feature extraction. In this phase, objects are measured. The measuring feature is to quantitatively estimate some important features of objects, and a group of the features are combined to make up a feature vector during feature extraction. The third phase is classification. In this phase, the output is just a decision to determine 3 which category every object belongs to. Therefore, for pattern recognition, what input

are images and what output are object types and structural analysis of images. The structural analysis is a description of images in order to correctly understand and judge for the important information of images.

1.2. PROBLEM STATEMENT

The problem statement centres around the concept of a camera-based sign language recognition system for the deaf, which would transform sign language gestures to text and subsequently text to speech. Our goal is to create a user-friendly and straightforward solution.

Dumb individuals communicate via hand signs, thus normal folks have a hard time understanding what they're saying. Speech impaired people use hand signs and gestures to communicate. Normal people face difficulty in understanding their language. Hence there is a need of a system which recognizes the different signs, gestures and conveys the information to the normal people. It bridges the gap between physically challenged people and normal people. As a result, systems that recognise various signs and deliver information to ordinary people are required.

1.3. OBJECTIVE

The main Objective of Sign Language Recognition (SLR) systems is to provide an efficient and accurate way to convert sign language into text or voice has aids for the hearing impaired for example, or enabling very young children to interact with computers (recognizing sign language), among others.

Goal of a sign language detecting system is to provide a practical mechanism for normal and deaf individuals to communicate through hand gestures. The proposed system will be used with a webcam or any other in-built camera that detects and processes indicators for recognition.

1.4. AIM OF THE PROJECT

Main Aim of Sign Language Recognition (SLR) systems is to provide an efficient and accurate way to convert sign language into text or voice has aids for the hearing impaired.

CHAPTER - 2

2. LITERATURE SURVEY

Literature review of our proposed system shows that there have been many explorations done to tackle the sign recognition in videos and images using several methods and algorithms.

Siming proposed a system having a dataset of 40 common words and 10,000 sign language images. To locate the hand regions in the video frame, Faster R-CNN with an embedded RPN module is used. It improves performance in terms of accuracy. Detection and template classification can be done at a higher speed as compared to single stage target detection algorithm such as YOLO.On the problem of RGB sign language image or video recognition in practical problems , the paper merges the hand locating network, 3D CNN feature extraction network and LSTM encoding and decoding to construct the algorithm for extraction. This paper has achieved a recognition of 99% in common vo- cabulary dataset.

Let's approach the research done by Rekha. which made use of YCbCr skin model to detect and fragment the skin region of the hand gestures. Using Principal Curvature based Region Detector, the image features are extracted and classified with Multi class SVM, DTW and non-linear KNN. The experimental result obtained were 94.4%for static and 86.4% for dynamic.

In Pigou L, a low cost approach has been used for image processing. The capture of images was done with a green background so that during processing, the green colour can be easily subtracted from the RGB colourspace and the image gets converted to black and white. The prototype has correctly recognised 92% of the sign gestures.

The paper by M. Geetha and U. C. Manjusha[7], make use of 50 specimens of every alphabets and digits in a vision based recognition of Indian Sign Language characters and numerals using B-Spine approximations. The region of interest of the sign gesture is analysed and the boundary is removed . The boundary obtained is further transformed to a B-spline curve by using the MaximumCurvature Points(MCPs) as the Control points. The B-spline curve undergoes a series of smoothening process so fea- tures can be extracted. Support vector machine is used to classify the images and the accuracy is 90.00%.

A similar work was done by J Huang [10].He created his own dataset using Kinect and got a total a total of 25 vocabularies which are used in everyday lives. He then applied a 3D CNN in which all kernels are also in 3D. The input of his model consisted of 5 important channels which are colour-r, colour-b, colour-g, depth and body skeleton. He got an average accuracy of 94.2%.

The domain analysis that we have done for the project mainly involved understanding the neural networks.

Year	Title	Journel name	Author	Methodology used	Performance metrics & Accuracy
2022	Dynamic hand gesture recognition of sign language using geometric features learning	IEEE	Joudaki & Rehman	Neural network model, geometrical sign language recognition, Artificial Intelligence and Data Analytics.	Proposed framework is proficient for sign language recognition using dynamic hand gesture and produces an accuracy of up to 89.52%.
2021	Evaluation of deep models and optimizers for Indian sign language recognition	IEEE	Sharma & Anand	Gradient-based optimizers, ImageNet, like AlexNet, GoogleNet and Gesture recognition.	A three-layered CNN model is also proposed and trained from scratch, which attained the best recognition accuracy of 99.0% and 97.6% on numerals and alphabets of a public ISL dataset.
2021	Deep learning-based sign language recognition system for static signs	IEEE	Wadhawan & Kumar	Deep learning-based convolutional neural networks, Pooling layer.	The proposed approach has achieved the highest training accuracy of 99.72% and 99.90% on colored and grayscale images, respectively.
2020	Deep Learning-Based subscribe Language Recognition System for Static Signs	IEEE	Kolivand, Joudaki S	Artificial Neural Network , Neural Computing and Applications, Deep learning-based convolutional neural networks.	ASLNN is proficient to hand posture recognition and provides accuracy up to 96.78%.
2022	Multi-Model Ensemble Gesture Recognition Network for High- Accuracy Dynamic Hand Gesture Recognition	IEEE	MohammedA, Lv J, IslamM & Sang Y	Hand gesture and action recognition, RNNs-based models, temporal convolutional network (TCN), and 3D convolutional neural networks.	Approach has achieved the highest training accuracy of 99.72% and 99.90%.
2020	A robust sign language and hand gesture recognition system using convolution neural networks	IEEE	A Varaprasadh, NSV Krishna Reddy, D Krishna Vamsi	Sign LanguageRecognition, Convolution Neural Network, Image Processing, Edge Detection, Hand Gesture Recogniton.	The system produces 88% accuracy.

TensorFlow:

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

Features: TensorFlow provides stable Python (for version 3.7 across all platforms) and C APIs; and without API backwards compatibility guarantee: C++, Go, Java, JavaScript and Swift (early release). Third-party packages are available for C#, Haskell Julia, MATLAB,R, Scala, Rust, OCaml, and Crystal."New language support should be built on top of the C API. However, not all functionality is available in C yet." Some more functionality is provided by the Python API.

Application: Among the applications for which TensorFlow is the foundation, are automated image-captioning software, suchas DeepDream.

Opencv:

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision.[1] Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel[2]). The library is cross-platform and free for use under the open-source BSD license.

OpenCV's application areas 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

Stereopsis stereo vision: depth perception from 2 cameras

- Structure from motion (SFM).
- Motion tracking
- Augmented reality

To support some of the above areas, OpenCV includes a statistical machine learning library that contains:

- 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)

AForge.NET, a computer vision library for the Common Language Runtime (.NET Framework and Mono).

ROS (Robot Operating System). OpenCV is used as the primary vision package in ROS.

VXL, an alternative library written in C++.

Integrating Vision Toolkit (IVT), a fast and easy-to-use C++ library with an optional interface to OpenCV.

CVIPtools, a complete GUI-based computer-vision and image-processing software environment, with C function libraries, a COM-based DLL, along with two utility programs for algorithm development and batch processing.

OpenNN, an open-source neural networks library written in C++.

List of free and open source software packages

- OpenCV Functionality
- Image/video I/O, processing, display (core, imgproc, highgui)
- Object/feature detection (objdetect, features2d, nonfree)
- Geometry-based monocular or stereo computer vision (calib3d, stitching, videostab)
- Computational photography (photo, video, superres)
- Machine learning & clustering (ml, flann)
- CUDA acceleration (gpu)

Image-Processing:

Image processing is a method to perform some operations on an image, in order to get an enhanced image and or to extract some useful information from it.

If we talk about the basic definition of image processing then “Image processing is the analysis and manipulation of a digitized image, especially in order to improve its quality”.

Digital-Image :

An image may be defined as a two-dimensional function f(x, y), where x and y are spatial(plane) coordinates, and the amplitude of fat any pair of coordinates (x, y) is called the intensity or grey level

of the image at that point.

In another word An image is nothing more than a two-dimensional matrix (3-D in case of coloured images) which is defined by the mathematical function f(x, y) at any point is giving the pixel value at that point of an image, the pixel value describes how bright that pixel is, and what colour it should be.

Image processing is basically signal processing in which input is an image and output is image or characteristics according to requirement associated with that image.

Image processing basically includes the following three steps :

Importing the image

Analysing and manipulating the image

Output in which result can be altered image or report that is based on image analysis

Applications of Computer Vision:

Here we have listed down some of major domains where Computer Vision is heavily used.

- Robotics Application
- Localization − Determine robot location automatically
- Navigation
- Obstacles avoidance
- Assembly (peg-in-hole, welding, painting)
- Manipulation (e.g. PUMA robot manipulator)
- Human Robot Interaction (HRI) − Intelligent robotics to interact with and serve people
- Medicine Application
- Classification and detection (e.g. lesion or cells classification and tumor detection)
- 2D/3D segmentation
- 3D human organ reconstruction (MRI or ultrasound)
- Vision-guided robotics surgery
- Industrial Automation Application
- Industrial inspection (defect detection)
- Assembly
- Barcode and package label reading

- Object sorting
- Document understanding (e.g. OCR)
- Security Application
- Biometrics (iris, finger print, face recognition)
- Surveillance − Detecting certain suspicious activities or behaviors
- Transportation Application
- Autonomous vehicle
- Safety, e.g., driver vigilance monitoring

2.1. EXISTING SYSTEM

Sign Language Recognition (SLR) system, which is required to recognize sign languages, has been widely studied for years.The studies are based on various input sensors, gesture segmentation, extraction of features and classifcation methods.This paper aims to analyze and compare the methods employed in the SLR systems, classi cations methods that have been used, and suggests the most promising method for future research. Due to recent advancement in classifcation methods, many of the recent proposed works mainly contribute on the classifcation methods, such as hybrid method and Deep Learning. This paper focuses on the classifcation methodsused in prior Sign Language Recognition system. Based on our review, HMMbased approaches have been explored extensively in prior research, including its modifcations.

This study is based on various input sensors, gesture segmentation, extraction of features and classification methods. This paper aims to analyze and compare the methods employed in the SLR systems, classifications methods that have been used, and suggests the most reliable method for future research. Due to recent advancement in classification methods, many of the recently proposed works mainly contribute to the classification methods, such as hybrid method and Deep Learning. Based on our review, HMM-based approaches have been explored extensively in prior research, including its modifications.Hybrid CNN-HMM and fully Deep Learning approaches have shown promising results and offer opportunities for further exploration.

The Disadvantages are

- Highly Expensive.
- Even if performed by any other ML techniques and With online, they Require tons of images to train for a single gesture.

2.2. PROPOSED SYSTEM

The proposed system can recognize the static word sign and represent its label for better communication. The deaf person should submit a gesture or sign image to the system in the proposed system. The system uses a mat lab image processing technique to analyse the sign input and classifies it for recognised identification. When the input image matches the specifi ed dataset, it then starts the voice media through the system. In addition, the output will be displayed in text format. This is a working prototype for the conversion of sign language to speech and text.

Hence, We are implementing using a popular Deep Learning technique known as "Transfer Learning" which help us fasten the training process and also require very less number of images for training the model.

2.3. SCOPE OF THE PROJECT

Sign languages are developed primarily to aid deaf and dumb people. They use a concurrent and specific combination of hand movements, hand shapes and orientation in order to convey particular information. The scope of sign language recognition encompasses various aspects, including technology, applications, and impact. Here are some key points related to the scope of sign language recognition:

Technology Advancements: Sign language recognition has been influenced by advancements in computer vision, machine learning, and deep learning techniques. These technologies have enabled the development of sophisticated algorithms capable of analyzing sign language gestures.

Gesture Recognition: Sign language recognition involves recognizing and interpreting the complex hand and body movements, facial expressions, and other visual cues used in sign language. It requires the development of computer vision algorithms that can accurately detect and track these gestures.

Real-Time Recognition: Real-time sign language recognition is an essential aspect of its scope. Systems need to process sign language gestures and provide instantaneous translations or responses, allowing for effective communication between sign language users and non-signers.

Application Areas: Sign language recognition has diverse applications. It can be used to develop assistive technologies for deaf or hearing-impaired individuals, facilitate communication between sign language users and non-signers, and enable accessibility in various domains such as education, healthcare, customer service, and more.

CHAPTER - 3

3. ANALYSIS

The major step in analysis is to verify the feasibility of the proposed system. “All projects are feasible given unlimited resources and infinite time“. But in reality, both resources and time are scarce. Project should confirm to be time effective and should be optimal in their consumption of resources. This plays a constant role in approval of any project.

Three key considerations involved in the feasibility analysis are

- Technical Feasibility
- Operational Feasibility
- Economical Feasibility

3.1. Technical Feasibility

To determine whether the proposed system is technically feasible, we should take into consideration the technical issues involved behind the system. Android project uses the android-based technologies, which is rampantly employed these days worldwide. The world without the internet is incomprehensible today. That goes to render that the proposed system is technically feasible.

Computer Vision Techniques: Advances in computer vision algorithms have made it possible to accurately detect and track hand and body movements, facial expressions, and other visual cues used in sign language. Techniques like convolutional neural networks (CNNs), recurrent neural networks (RNNs), and deep learning have been applied successfully to analyze sign language gestures.

Sensor Technologies: Sign language recognition systems can utilize various sensor technologies to capture sign language gestures accurately. These include cameras, depth sensors (e.g., Microsoft Kinect), wearable devices, or even gloves with integrated sensors. These sensors provide the necessary input for the recognition algorithms to analyze and interpret sign language gestures.

Integration with Applications: Sign language recognition can be integrated into various applications and devices, such as smartphones, tablets, smart glasses, or interactive displays. This integration allows for on-the-go communication and accessibility in different domains, such as education, healthcare, customer service, and more.

3.2. Operational Feasibility

To determine the operational feasibility of the system we should take into consideration the awareness level of the users. This system is operationally feasible since the users are familiar with the android technologies and hence there is no need to gear up or train the personnel to use the cell phones. Also, the system (android phones) is very friendly and easy to use.

3.3. Economical Feasibility

Economical feasibility is an important aspect to consider when evaluating the potential of sign language recognition systems. Here are some considerations related to the economical feasibility of sign language recognition are :

Cost of Development: The development of sign language recognition systems involves significant investment in research and development. It requires expertise in computer vision, machine learning, and data collection. The cost includes hiring skilled professionals, acquiring relevant datasets, and investing in computational resources.

Hardware and Infrastructure: Sign language recognition systems may require specialized hardware and infrastructure to achieve real-time and accurate recognition. This can include high-performance GPUs or dedicated hardware accelerators to process the visual input efficiently. The cost of acquiring and maintaining such hardware should be considered.

Model Training and Optimization: Training and optimizing sign language recognition models can be computationally intensive, requiring significant computational resources. This may lead to increased costs related to cloud computing or dedicated hardware infrastructure.

Deployment and Integration: Once the sign language recognition system is developed, there may be costs associated with deploying and integrating it into existing platforms or applications. This can involve software engineering efforts, integration with user interfaces, and compatibility testing.

Maintenance and Updates: Sign language recognition systems may require ongoing maintenance and updates to address performance issues, improve accuracy, and adapt to evolving sign language variations. This can incur additional costs in terms of personnel, infrastructure, and software updates.

Scalability and Accessibility: The economical feasibility of sign language recognition systems also depends on their scalability and accessibility. If the system needs to cater to a large user base or be

widely accessible, considerations should be given to scaling the infrastructure, managing user support, and ensuring cost-effective deployment options

Potential Benefits and Market Opportunities: Assessing the economical feasibility of sign language recognition should also consider the potential benefits and market opportunities. Sign language recognition systems can enable accessibility in various sectors, such as education, healthcare, customer service, and entertainment. The market demand and willingness to pay for such solutions can contribute to their economical feasibility.

It is worth noting that the economical feasibility of sign language recognition systems can vary depending on factors such as the specific application domain, target user base, available resources, and the level of maturity of the technology. Proper cost-benefit analysis and business planning are crucial to determining the viability and sustainability of sign language recognition projects.

CHAPTER - 4

4. HARDWARE AND SOFTWARE REQUIREMENTS

4.1 Hardware Requirements

- The Hardware Interfaces Required are:
- Camera: Good quality,3MP
- Ram: Minimum 8GB or higher
- GPU: 4GB dedicated
- Processor: Intel Pentium 4 or higher
- HDD: 10GB or higher 7

4.2 Software Requirements

- Software requirements Operating System : Windows, Mac, Linux
- SDK: OpenCV ,TensorFlow, Keros, Numpy

CHAPTER - 5

5 . SYSTEM DESIGN

System design is the transition from a user-oriented document to programmers or database personnel. The design is a solution, specifying how to approach to the creation of a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Designing goes through logical and physical stages of development. Logical design reviews the present physical system, prepare input and output specification, details of implementation plan and prepare a logical design walkthrough.

The database tables are designed by analysing functions involved in the system and format of the fields is also designed. The fields in the database tables should define their role in the system. The unnecessary fields should be avoided because it affects the storage areas of the system. Then, in the input and output screen design, the design should be made user friendly. The menu should be precise and compact.

5.1 SOFTWARE DESIGN

In designing the software, the following principles are followed:

- Modularity and partitioning: software is designed in such a way that each system should consist of hierarchy of modules and serve to partition into separate function.
- Coupling: modules should have little dependency on the other modules of a system.
- Cohesion: modules should carry out the operations in a single processing function.
- Shared use: avoid duplication by allowing a single module which is called by other, that needsthe function it provides.

5.2 INPUT DESIGN

Considering the requirements, procedures are adopted to collect the necessary input data in most efficiently designed format. The input design has to be done keeping in view that, the interaction of the user with the system should be in the most effective and simplified way. Also, the necessary measures are taken for the following

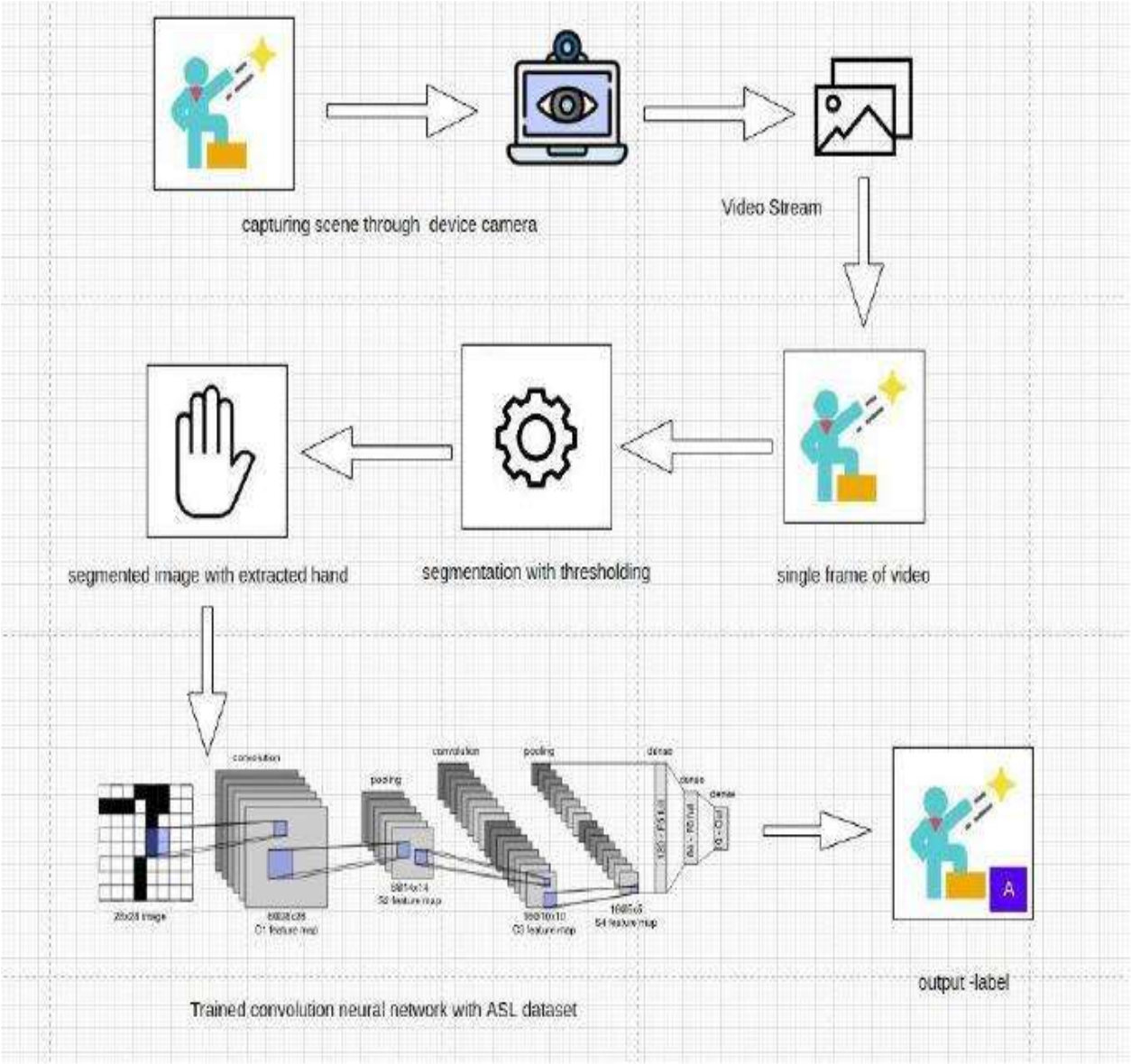
- Controlling the amount of input
- Avoid unauthorized access to the users
- Eliminating the extra steps
- Keeping the process simple
- At this stage the input forms and screens are designed.

5.3 OUTPUT DESIGN

All the screens of the system are designed with a view to provide the user with easy operations in a simpler and efficient way, with minimum key strokes possible. Important information is emphasized on the screen. Almost every screen is provided with no error and important messages and option selection facilitates. Emphasis is given for faster processing and speedy transactions between the screens. Each screen assigned to make it as much user friendly as possible by using interactive procedures. In other words, we can say that the user can operate the system without much help from the operating manual.

5.4 ARCHITECTURE

Convolutional neural networks (CNN) is a special architecture of artificial neural networks, proposed by Yann LeCun in 1988. CNN uses some features of the visual cortex. One of the most popular uses of this architecture is image classification. For example Facebook uses CNN for automatic tagging algorithms, Amazon — for generating product recommendations and Google — for search through among users' photos.



5.4.1 Architecture of Sign Language recognition System

TRAINING MODULE:

Supervised machine learning:

It is one of the ways of machine learning where the model is trained by input data and expected output data. To create such model, it is necessary to go through the following phases:

1. Model construction

2. Model training

3. Model testing

4. Model evaluation

Model construction:

It depends on machine learning algorithms. In this projectscase, it was neural networks.Such an agorithm looks like:

1. begin with its object: model = Sequential()

2. then consist of layers with their types: model.add(type_of_layer())

3. after adding a sufficient number of layers the model is compiled. At this moment Keras communicates with TensorFlow for construction of the model. During model compilation it is important to write a loss function and an optimizer algorithm. It looks like: model.comile(loss= 'name_of_loss_function', optimizer= 'name_of_opimazer_alg') The loss function shows the accuracy of each prediction made by the model.

Before model training it is important to scale data for their further use.

Model training:

After model construction it is time for model training. In this phase, the model is trained using training data and expected output for this data. It's look this way: model.fit(training_data, expected_output). Progress is visible on the console when the script runs. At the end it will report the final accuracy of the model.

Model Testing:

During this phase a second set of data is loaded. This data set has never been seen by the model and therefore it's true accuracy will be verified. After the model training is complete, and it is understood that the model shows the right result, it can be saved by: model.save("name_of_file.h5"). Finally, the saved model can be used in the real world. The name of this phase is model evaluation. This means that the model can be used to evaluate new data.

Preprocessing:

Uniform aspect ratio

Understanding aspect ratios: An aspect ratio is a proportional relationship between an image's width and height. Essentially, it describes an image's shape.Aspect ratios are written as a formula of width to height, like this: For example, a square image has an aspect ratio of 1:1, since the height and width are the same. The image could be 500px × 500px, or 1500px × 1500px, and the aspect ratio would still be 1:1.As another example, a portrait-style image might have a ratio of 2:3. With this aspect ratio, the height is 1.5 times longer than the width. So the image could be 500px × 750px, 1500px × 2250px, etc.

Cropping to an aspect ratio

Aside from using built in site style options , you may want to manually crop an image to a certain aspect ratio. For example, if you use product images that have same aspect ratio, they'll all crop the same way on your site. 7 Option 1 - Crop to a pre-set shape Use the built-in Image Editor to crop images to a specific shape. After opening the editor, use the crop tool to choose from preset aspect ratios. Option 2 - Custom dimensions To crop images to a custom aspect ratio not offered by our built-in Image Editor, use a third-party editor. Since images don’t need to have the samedimensions to have the same aspect ratio, it’s better to crop them to a specific ratio than to try to matchtheir exact dimensions. For best results, crop the shorter side based on the longer side. • For instance, if your image is 1500px × 1200px, and you want an aspect ratio of 3:1, crop the shorter side to make the image 1500px × 500px. • Don't scale up the longer side; this can make your image blurry.

Image scaling:

• In compuer graphics and digital imaging , image scaling refers to the resizing of a digital image. In video technology, the magnification of digital material is known as upscaling or resolution enhancement .

• When scaling a vector graphic image, the graphic primitives that make up the image can be scaled using geometric transformations, with no loss of image quality. When scaling a raster graphics image, a new image with a higher or lower number of pixels must be generated. In the case of decreasing the pixel number (scaling down) this usually results in avisible quality loss. From the standpoint of digital signal processing, the scaling of raster graphics is a twodimensional example of sample-rate conversion, the conversion of a discrete signal from a sampling rate (in this case the local sampling rate) to another.

5.5 UML DIAGRAMS

Unified Modelling Language:

UML is an acronym that stands for **Unified Modeling Language**. Simply put, UML is a modern approach to modeling and documenting software. In fact, it's one of the most popular business process modeling techniques. It is based on **diagrammatic representations** of software components. As the old proverb says: "a picture is worth a thousand words". By using visual representations, we are able to better understand possible flaws or errors in software or business processes.

The elements are like components which can be associated in different ways to make a complete UML picture, which is known as diagram. Thus, it is very important to understand the different diagrams to implement the knowledge in real-life systems.

Any complex system is best understood by making some kind of diagrams or pictures. These diagrams have a better impact on our understanding. If we look around, we will realize that the diagrams are not a new concept but it is used widely in different forms in different industries.

We prepare UML diagrams to understand the system in a better and simple way. A single diagram is not enough to cover all the aspects of the system. UML defines various kinds of diagrams to cover most of the aspects of a system.

You can also create your own set of diagrams to meet your requirements. Diagrams are generally made in an incremental and iterative way.

5.5.1 USE CASE DIAGRAM

A use case diagram contains four components:

- The boundary, which defines the system of interest in relation to the world aroundit.
- The actors, usually individuals involved with the system defined according to theirroles.
- The use cases, which are the specific roles played by the actors within and aroundthe system.
- The relationships between and among the actors and the use cases.

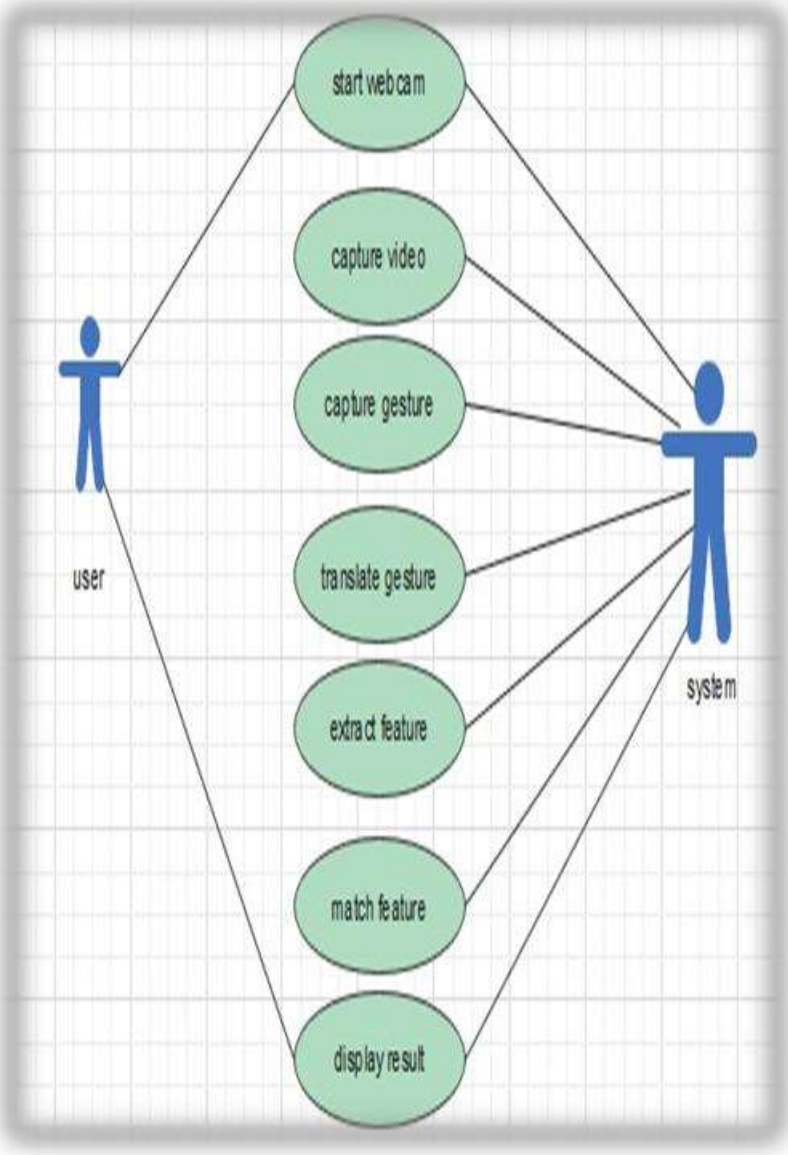


Fig:5.5.1 Use case Diagram for Real Time Sign Language Recognition

Usecase name	Sign Language Recognition
Participating actors	User, System
Flow of events	Start the system(u) Capturing videos Capture gestures Translate gestures Extract gestures Match features Recognizing gestures Display result
Entry condition	Run the code
Exit condition	Displaying the label
Quality requirements	Cam pixels clarity, good light condition

Table 5.5.1: Usecase Scenario for Sign Language Recognition System

5.5.2 CLASS DIAGRAM

The purpose of the class diagram is to model the static view of an application. The class diagrams are the only diagrams which can be directly mapped with object oriented languages and thus widely used at the time of construction.

PURPOSE OF CLASS DIAGRAMS

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering

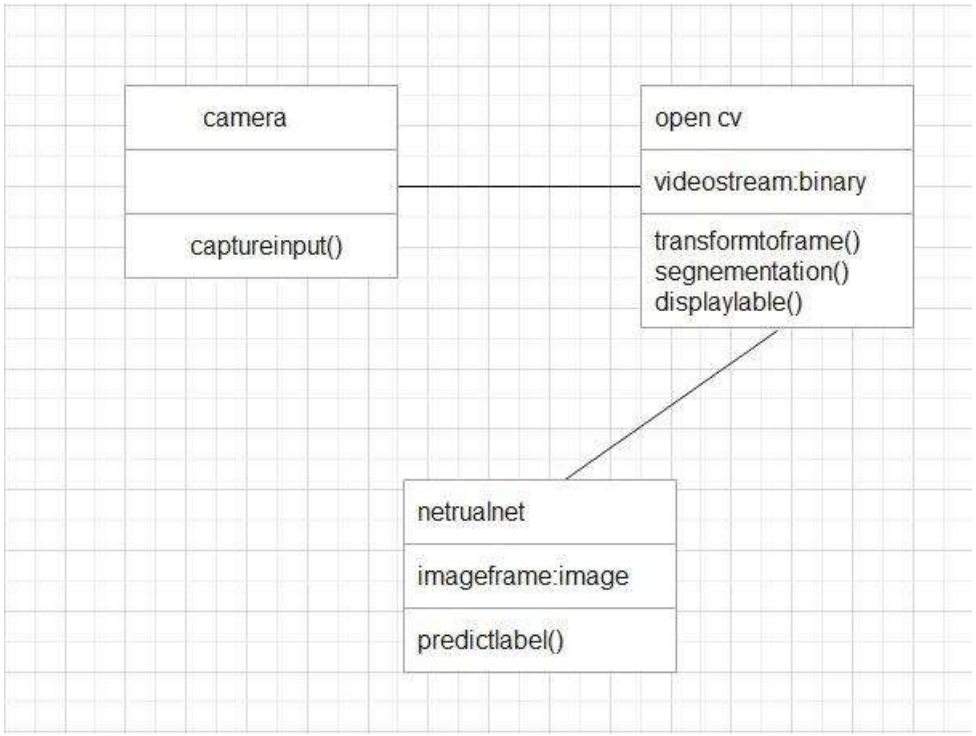


Fig:5.5.2 Class Diagram for Real Time Sign Language Recognition

5.5.3 ACTIVITY DIAGRAM

- It shows the flow of the various activities that are undergone from the beginning tillthe end.
- It consists of the activities that are held and carried out throughout the session fromstarting till the ending stage.

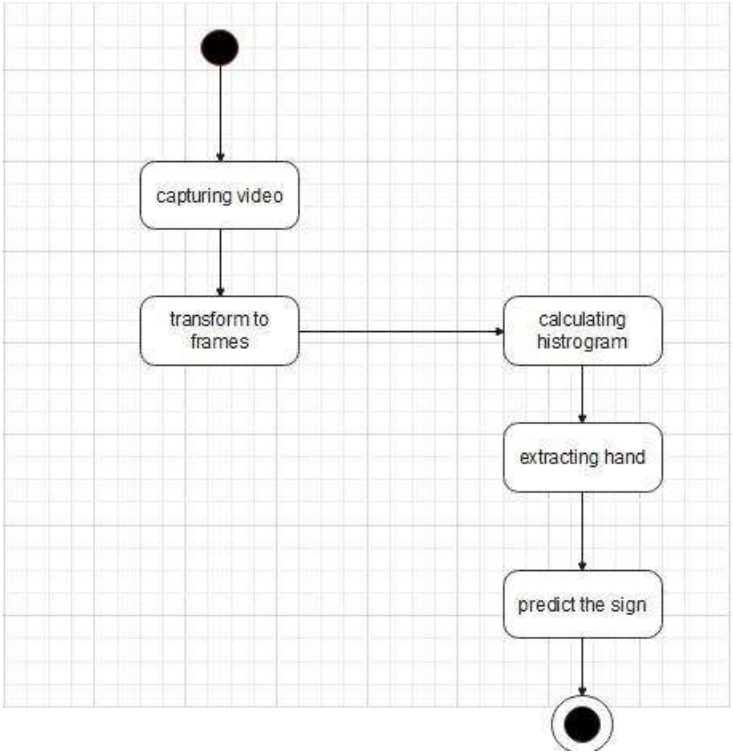


Fig:5.5.3 Activity Diagram for Real Time Sign Language Recognition

5.5.4 SEQUENCE DIAGRAM

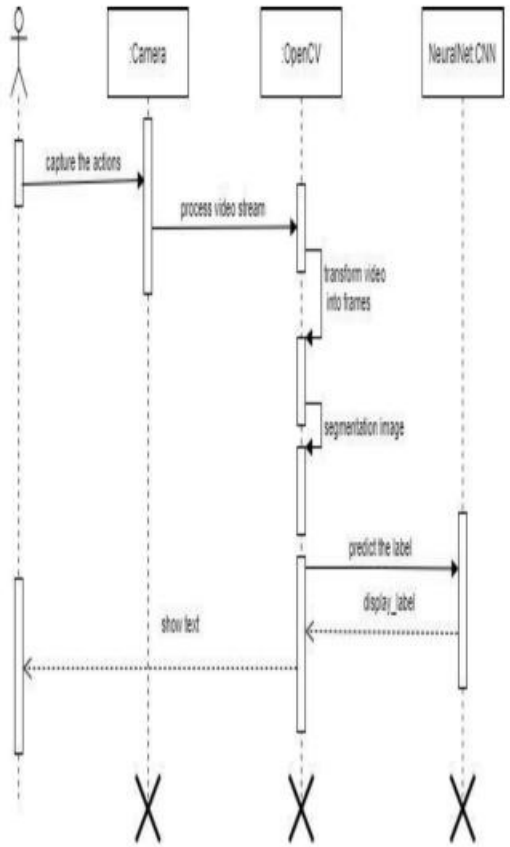


Fig:5.5.4 Sequence Diagram for Real Time Sign Language Recognition

- It shows the sequence of the steps that are carried out throughout the process of execution.
- It involves lifelines or life time of a process that shows the duration for which the process is alive while the steps are taking place in the sequential manner.
- Sequence diagram specifies the order in which the various steps are executed.

5.5.5 STATE CHART

A state chart diagram describes a state machine which shows the behaviour of classes. It shows the actual changes in state not processes or commands that create those changes and is the dynamic behaviour of objects over time by model1ing the life cycle of objects of each class.

It describes how an object is changing from one state to another state. There are mainly two states in State Chart Diagram:

1. Initial State
2. Final-State.

Some of the components of State Chart Diagram are:

State: It is a condition or situation in life cycle of an object during which it's satisfies same condition or performs some activity or waits for some event.

Transition: It is a relationship between two states indicating that object in first state performs some actions and enters into the next state or event.

Event: An event is specification of significant occurrence that has a location in time and space.

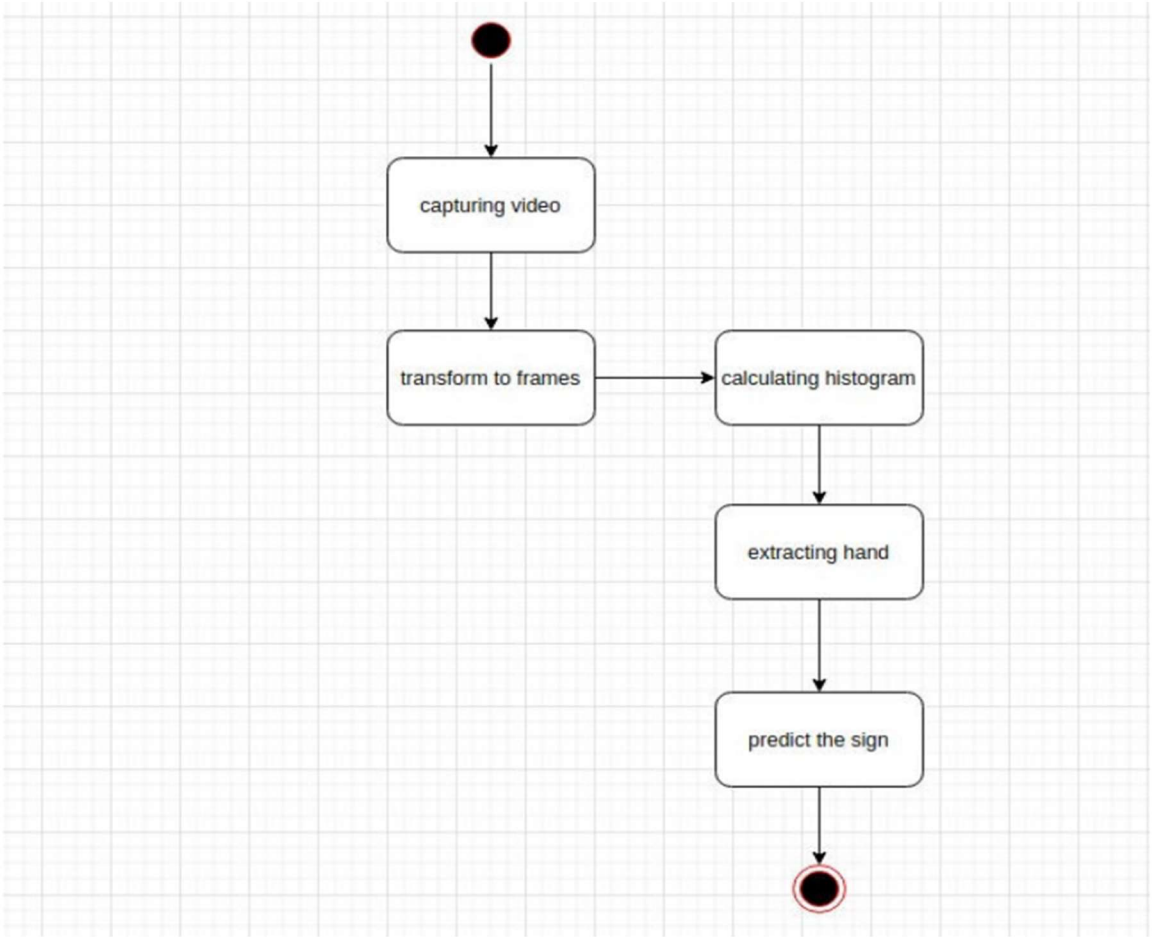


Fig:5.5.5 State Chart diagram for Real Time Sign Language Recognition

CHAPTER - 6

6. RESULTS AND PERFORMANCE EVALUATION

6.1 CODE

6.1.1 DATA COLLECTION

HISTOGRAM CALCULATION:

Histograms are collected counts of data organized into a set of predefined bins

When we say data we are not restricting it to be intensity value. The data collected can be whatever feature you find useful to describe your image.

Let's see an example. Imagine that a Matrix contains information of an image (i.e. intensity in the range 0−255):

What happens if we want to count this data in an organized way? Since we know that the range of information value for this case is 256 values, we can segment our range in subparts (called bins) like:

[0,255]=[0,15]∪[16,31]∪....∪[240,255]range=bin1∪bin2∪....∪binn=15 and we can keep count of the number of pixels that fall in the range of each bini

BackPropogation: Back-propagation is the essence of neural net training. It is the method of fine-tuning the weights of a neural net based on the error rate obtained in the previous epoch (i.e., iteration). Proper tuning of the weights allows you to reduce error rates and to make the model reliable by increasing its generalization.

Backpropagation is a short form for "backward propagation of errors." It is a standard method of training artificial neural networks. This method helps to calculate the gradient of a loss function with respects to all the weights in the network.

Optimizer(Adam): Adam can be looked at as a combination ofRMSprop and Stochastic Gradient Descent with momentum. It uses the squared gradients to scale the learning rate like RMSprop and it takes advantage of momentum by using moving average of the gradient instead of gradient itself like SGD with momentum. Adam is an adaptive learning rate method, which means, it computes individual learning rates for different parameters. Its name is derived from adaptive moment estimation, and the reason it's called that is because Adam uses estimations of first and second moments of gradient to adapt the learning rate for each weight of the neural network. Now, what is moment ? N-th moment of a random variable is defined as the expected value of that variable to the power of n. More formally:

Loss Function(categorical cross entrophy): Categorical crossentropy is a loss function that is used for single label categorization. This is when only onecategory isapplicable for each data point. In other words, an example can belong to one class only.

Note. The block before the Target block must use the activation function Softmax.

SEGMENTATION

Image segmentation is the process of partitioning a digital image into multiple segments(sets of pixels, also known as image objects). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyse.Modern image segmentation techniques are powered by deep learning technology. Here are several deep learning architectures used for segmentation:

Why does Image Segmentation even matter?

If we take an example of Autonomous Vehicles, they need sensory input devices like cameras, radar, and lasers to allow the car to perceive the world around it, creating a digital map. Autonomous driving is not even possible without object detection which itself involves image classification/segmentation.

How Image Segmentation works ?

Image Segmentation involves converting an image into a collection of regions of pixels that are represented by a mask or a labeled image. By dividing an image into segments, you can process only the important segments of the image instead of processing the entire image. A common technique is to look for abrupt discontinuities in pixel values, which typically indicate edges that define a region.Another common approach is to detect similarities in the regions of an image. Some techniques that follow this approach are region growing, clustering, and thresholding. A variety of other approaches to perform image segmentation have been developed over the years using domain-specific knowledge to effectively solve segmentation problems in specific application areas.

CLASSIFICATION :CONVOLUTION NEURAL NETWORK

Image classification is the process of taking an input(like a picture) and outputting its class or probability that the input is a particular class. Neural networks are applied in the following steps:

1)One hot encode the data: A one-hot encoding can be applied to the integer representation. This is where the integer encoded variable is removed and a new binary variable is added for each unique integer value.

2)Define the model: A model said in a very simplified form is nothing but a function that is used to take in certain input, perform certain operation to its beston the given input (learning and then predicting/classifying) and produce the suitable output.

3)Compile the model: The optimizer controls the learning rate. We will be using 'adam' as our optmizer. Adam is generally a good optimizer to use for many cases. The adam optimizer adjusts the learning rate throughout training. The learning rate determines how fast the optimal weights for the model are calculated. A smaller learning rate may lead to more accurate weights (up to a certain point), but the time it takes to compute the weights will be longer.

4)Train the model: Training a model simply means learning (determining) good values for all the weights and the bias from labeled examples. In supervised learning, a machine learning algorithm builds a model by examining many examples and attempting to find a model that minimizes loss; this process is called empirical risk minimization.

5)Test the model: A convolutional neural network convolves learned featured with input data and uses 2D convolution layers.

ConvolutionOperation:

In purely mathematical terms, convolution is a function derived from two given functions by integration which expresses how the shape of one is modified by the other.

Convolution formula:

$(f * g)(t) = \int f(r)g(t – r)dr$

Here are the three elements that enter into the convolution operation:

• Input image

• Feature detector

• Feature map

Steps to apply convolution layer:

• You place it over the input image beginning from the top-left corner within the borders you see demarcated above, and then you count the number of cells in which the feature detector matches the input image.

• The number of matching cells is then inserted in the top-left cell of the feature map

• You then move the feature detector one cell to the right and do the same thing. This movement is called a and since we are moving the feature detector one cell at time, that would be called a stride of one pixel.

• What you will find in this example is that the feature detector's middle-left cell with the number 1 inside it matches the cell that it is standing over inside the input image. That's the only matching cell, and so you write “1” in the next cell in the feature map, and so on and so forth.

• After you have gone through the whole first row, you can then move it over to the next row and go through the same process.

There are several uses that we gain from deriving a feature map. These are the most important of them: Reducing the size of the input image, and you should know that the larger your strides (the movements across pixels), the smaller your feature map.

Relu Layer:

Rectified linear unit is used to scale the parameters to non negativevalues.We get pixel values as negative values too . Inthis layer we make them as 0’s. The purpose of applying the rectifier function is to increase the non-linearity in our images. The reason we want to do that is that images are naturally non-linear. The rectifier serves to break up the linearity even further in order to make up for the linearity that we might impose an image when we put it through the convolution operation. What the rectifier function does to an image like this is remove all the black elements from it, keeping only those carrying a positive value (the grey and white colors).The essential difference between the non-rectified version of the image and the rectified one is the progression of colors. After we rectify the image, you will find the colors changing more abruptly. The gradual change is no longer there. That indicates that the linearity has been disposed of.

Pooling Layer:

The pooling (POOL) layer reduces the height and width of the input. It helps reduce computation, as well as helps make feature detectors more invariant to its position in the input This process is what provides the convolutional neural network with the “spatial variance” capability. In addition to that, pooling serves to minimize the size of the images as well as the number of parameters which, in turn, prevents an issue of “overfitting” from coming up. Overfitting in a nutshell is when you create an excessively complex model in order to account for the idiosyncracies we just mentioned The result ofusing a pooling layer and creating down sampled or pooled feature maps is a summarized version of the features detected in the input. They are useful as small changes

in the location of the feature in the input detected by the convolutional layer will result in a pooled feature map with the feature in the same location. Thiscapability added by pooling is called the model's invariance to local translation.

Fully Connected Layer:

The role of the artificial neural network is to take this data and combine the features into a wider variety of attributes that make the convolutional network more capable of classifying images, which is the whole purpose from creating a convolutional neural network. It has neurons linked to each other ,and activates if it identifies patterns and sends signals to output layer .the outputlayer gives output class based on weight values, For now, all you need to know is that the loss function informs us of how accurate our network is, which we then use in optimizing our network in order to increase its effectiveness. That requires certain things to be altered in our network. These include the weights (the blue lines connecting the neurons, which are basically the synapses), and the feature detector since the network often turns out to be looking for the wrong features and has to be reviewed multiple times for the sake of optimization.This full connection process practically works as follows:

• The neuron in the fully-connected layer detects a certain feature; say, a nose.

• It preserves its value.

• It communicates this value to the classes trained images.

datacollection.py

```
import cv2
from cvzone.HandTrackingModule import HandDetector
import numpy as np
import math
import time

cap = cv2.VideoCapture(0)
detector = HandDetector(maxHands=1)

offset = 20
imgSize = 300

folder = "Data/A"
counter = 0

while True:
    success, img = cap.read()
    hands, img = detector.findHands(img)
    if hands:
        hand = hands[0]
```

```
        x, y, w, h = hand['bbox']

        imgWhite = np.ones((imgSize, imgSize, 3), np.uint8) * 255
        imgCrop = img[y - offset:y + h + offset, x - offset:x + w + offset]

        imgCropShape = imgCrop.shape

        aspectRatio = h / w

        if aspectRatio > 1:
            k = imgSize / h
            wCal = math.ceil(k * w)
            imgResize = cv2.resize(imgCrop, (wCal, imgSize))
            imgResizeShape = imgResize.shape
            wGap = math.ceil((imgSize - wCal) / 2)
            imgWhite[:, wGap:wCal + wGap] = imgResize

        else:
            k = imgSize / w
            hCal = math.ceil(k * h)
            imgResize = cv2.resize(imgCrop, (imgSize, hCal))
            imgResizeShape = imgResize.shape
            hGap = math.ceil((imgSize - hCal) / 2)
            imgWhite[hGap:hCal + hGap, :] = imgResize

        cv2.imshow("ImageCrop", imgCrop)
        cv2.imshow("ImageWhite", imgWhite)

    cv2.imshow("Image", img)
    key = cv2.waitKey(1)
    if key == ord("s"):
        counter += 1
        cv2.imwrite(f'{folder}/Image_{time.time()}.jpg',imgWhite)
        print(counter)
```

6.1.2 TESTING

The purpose of testing is to discover errors. Testing is a process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner.

Software testing is an important element of the software quality assurance and represents the ultimate review of specification, design and coding. The increasing feasibility of software as a system and the cost associated with the software failures are motivated forces for well planned through testing.

Testing Objectives:

There are several rules that can serve as testing objectives they are:

- Testing is a process of executing program with the intent of finding an error.
- A good test case is the one that has a high probability of finding an undiscovered error.

Types of Testing:

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at different phases of software development are :

Unit Testing:

Unit testing is done on individual models as they are completed and becomes executable. It is confined only to the designer's requirements. Unit testing is different from and should be preceded by other techniques, including:

- Inform Debugging
- Code Inspection

Black Box testing:

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program.

This testing has been used to find error in the following categories:

Incorrect or missing functions

- Interface errors
- Errors in data structures are external database access
- Performance error
- Initialisation and termination of errors
- In this testing only the output is checked for correctness

- The logical flow of data is not checked

White Box testing:

In this the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases.

It has been used to generate the test cases in the following cases:

- Guarantee that all independent paths have been executed
- Execute all loops at their boundaries and within their operational bounds.
- Execute internal data structures to ensure their validity.

IntegrationTesting:

Integration testing ensures that software and subsystems work together a whole. It test the interface of all the modules to make sure that the modules behave properly when integrated together. It is typically performed by developers, especially at the lower, module to module level. Testers become involved in higher levels.

System Testing:

Involves in house testing of the entire system before delivery to the user. The aim is to satisfy the user the system meets all requirements of the client's specifications. It is conducted by the testing organization if a company has one. Test data may range from and generated to production.

Requires test scheduling to plan and organize:

- Inclusion of changes/fixes.
- Test data to use

One common approach is graduated testing: as system testing progresses and (hopefully) fewer and fewer defects are found, the code is frozen for testing for increasingly longer time periods.

Acceptance Testing:

It is a pre-delivery testing in which entire system is tested at client's site on real world data to find errors.

User Acceptance Test (UAT)

"Beta testing": Acceptance testing in the customer environment. Requirements traceability:

- Match requirements to test cases.
- Every requirement has to be cleared by at least one test case.
- Display in a matrix of requirements vs. test cases.

testing.py

```
import cv2
from cvzone.HandTrackingModule import HandDetector
from cvzone.ClassificationModule import Classifier
import numpy as np
import math

cap = cv2.VideoCapture(0)
detector = HandDetector(maxHands=1)
classifier = Classifier("Model/keras_model.h5", "Model/labels.txt")

offset = 20
imgSize = 300

counter = 0

labels = ["A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T",
"U", "V", "W", "X", "Y", "Z"]

while True:
    success, img = cap.read()
    imgOutput = img.copy()
    hands, img = detector.findHands(img)
    if hands:
        hand = hands[0]
        x, y, w, h = hand['bbox']

        imgWhite = np.ones((imgSize, imgSize, 3), np.uint8) * 255
        imgCrop = img[y - offset:y + h + offset, x - offset:x + w + offset]

        imgCropShape = imgCrop.shape

        aspectRatio = h / w

        if aspectRatio > 1:
            k = imgSize / h
            wCal = math.ceil(k * w)
            imgResize = cv2.resize(imgCrop, (wCal, imgSize))
            imgResizeShape = imgResize.shape
            wGap = math.ceil((imgSize - wCal) / 2)
            imgWhite[:, wGap:wCal + wGap] = imgResize
            prediction, index = classifier.getPrediction(imgWhite, draw=False)
            print(prediction, index)

        else:
```

```
            k = imgSize / w
            hCal = math.ceil(k * h)
            imgResize = cv2.resize(imgCrop, (imgSize, hCal))
            imgResizeShape = imgResize.shape
            hGap = math.ceil((imgSize - hCal) / 2)
            imgWhite[hGap:hCal + hGap, :] = imgResize
            prediction, index = classifier.getPrediction(imgWhite, draw=False)

        cv2.rectangle(imgOutput, (x - offset, y - offset-50),
                      (x - offset+90, y - offset-50+50), (255, 0, 255), cv2.FILLED)
        cv2.putText(imgOutput, labels[index], (x, y -26), cv2.FONT_HERSHEY_COMPLEX, 1.7, (255, 255,
255), 2)
        cv2.rectangle(imgOutput, (x-offset, y-offset),
                      (x + w+offset, y + h+offset), (255, 0, 255), 4)


        cv2.imshow("ImageCrop", imgCrop)
        cv2.imshow("ImageWhite", imgWhite)

    cv2.imshow("Image", imgOutput)
    cv2.waitKey(1)
```

Id	Test case	Input description	Expected output	Test status
1	Loadind model	Initializing trained model and load it into ON	Loaded model without errors	Pass
2	Converting video to frames	Capturing video and converting it into frames	Image frames of captured video stream	Pass
3	Recognize hand gesture	Image frame that contains hand object	label	Pass

Table 6.1: Verification of Testcases

6.2 SCREENSHOTS OF RESULTS

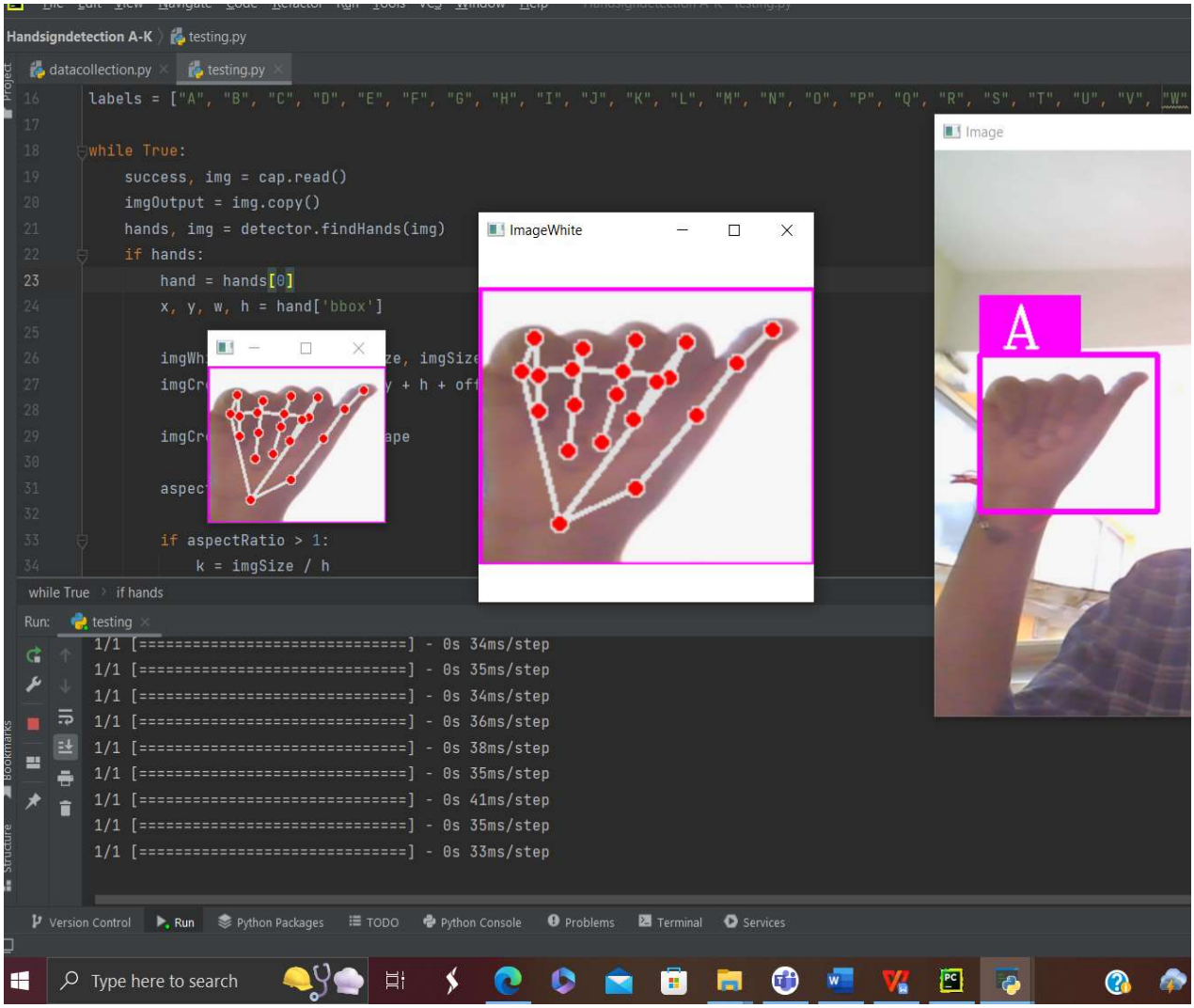


Fig 6.2.1 Result obtained for letter A

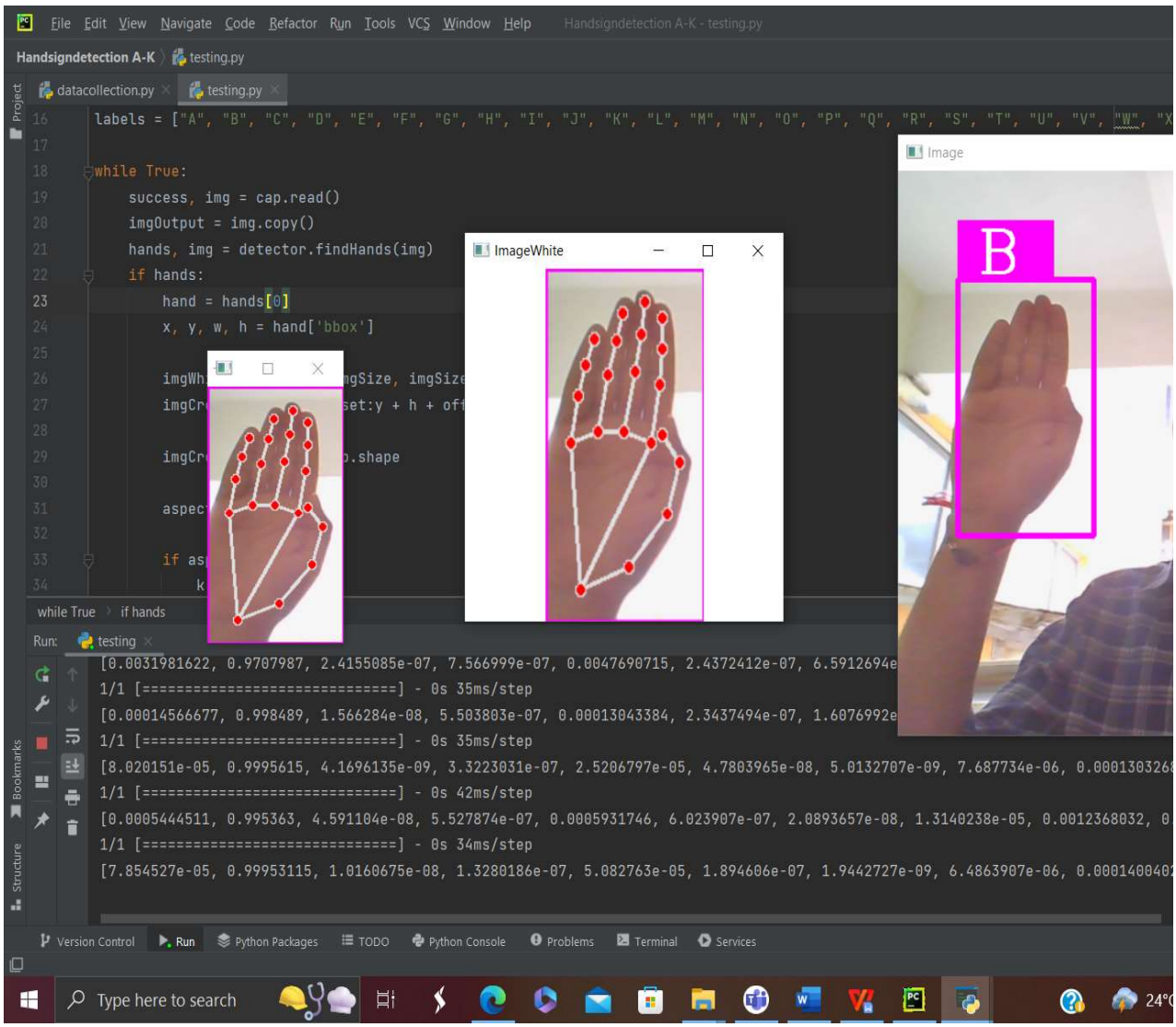


Fig 6.2.2 Result obtained for letter B

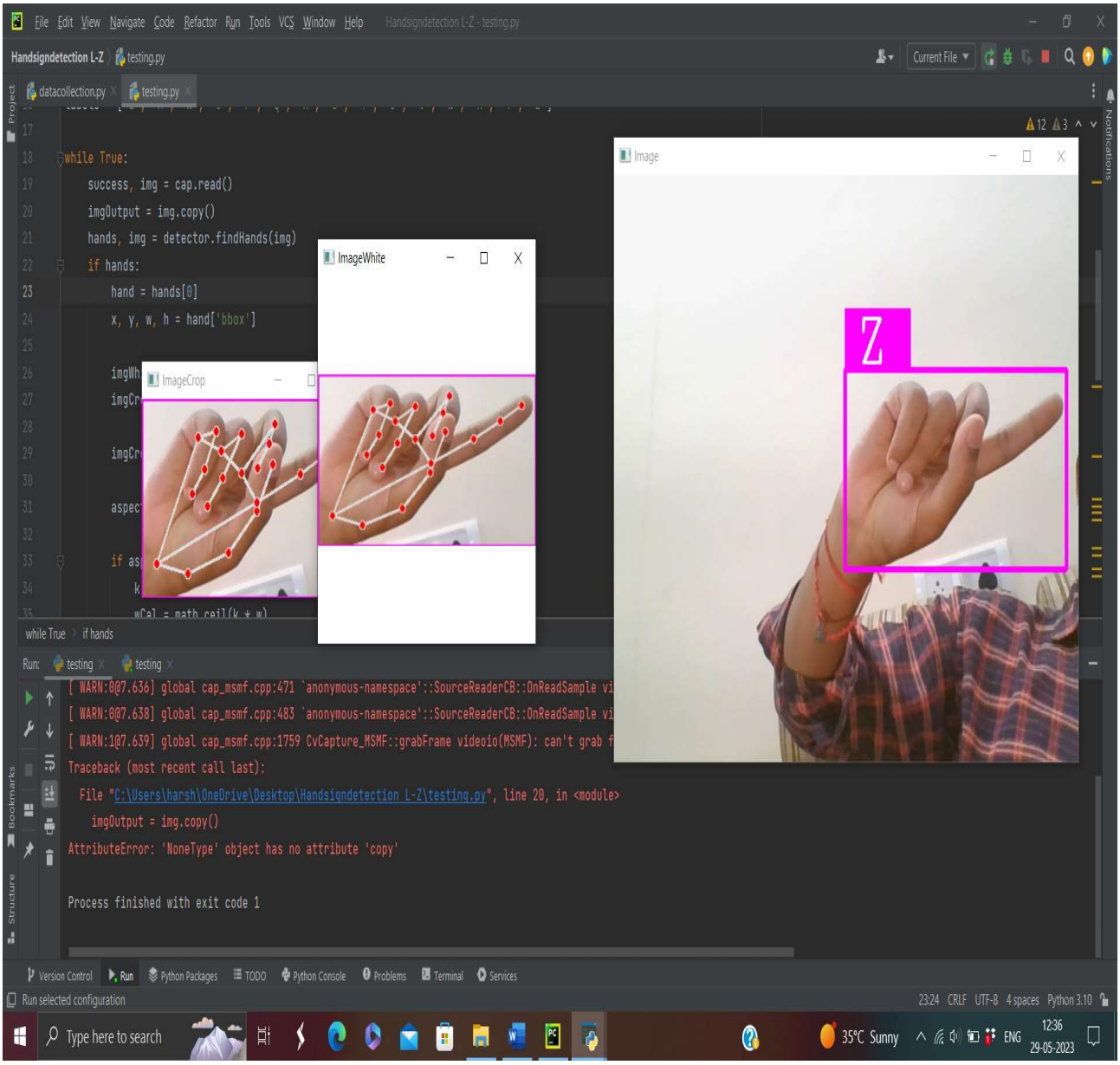


Fig 6.2.3 Result obtained for letter Z

CHAPTER – 7

7. CONCLUSION AND FUTURE SCOPE

Visual languages known as sign languages use facial expressions, hand and body gestures, and other physical motions to convey meaning. For persons with disabilities to be able to communicate, sign languages are crucial. They are able to share their emotions, communicate, and express `themselves via it.TensorFlow's object detection API was used to do it in this study.

The Indian Sign Language alphabet recognition was used to train and. Python and OpenCV were used to collect input images from camera for data gathering. Despite the system's high confidence rate, dataset are trained on certain methods and limitations.

Future work

The proposed sign language recognition system used to recognize sign language letters can be further extended to recognize gestures facial expressions. Instead of displaying letter labels it will be more appropriate to display sentences as more appropriate translation of language.This also increases readability.The scope of different sign languages can be increased. More training data can be added to detect the letter with more accuracy. This project can further be extended to convert the signs to speech.

CHAPTER - 8

8. REFERENCES

1) **Joudaki & Rehman**, Hand Gesture Recognition.

https//doi.org/10.1504/ijcvr.2022.119239

2) **Sharma & Anand**, Implementation of Deep projects and optimizers.

https//doi.org/10.1016/j.gvc.2021.200032

3) **Wadhawan & Kumar**, Deep Learning- Based Language Recognition System.

4) **Kolivand, Joudaki S**, Framework for Language Alphabet Hand Posture Recognition.

5) **MohammedA, Lv J, IslamM & Sang Y**, Gesture Recognition Network for High- Dynamic Hand Gesture Recognition.

6) **A Varaprasadh, NSV Krishna Reddy, D Krishna Vamsi**, A Robust Sign Language And Hand Gesture Recognition System Using Convolution Neural Networks.

PROJECT MAPPING

Topic	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3	PS O4
Real time sign language recognition Using transfer learning	✔	✔	✔	✔	✔			✔	✔	✔	✔	✔	✔	✔	✔	✔