**Hand Sign Language Recognition**

**Using Machine Learning And**

**Computer Vision**

*Final Year Project Report*

*Submitted in partial fulfillment of the requirements for the degree of*

***Bachelor of Technology***

*in*

***Information Technology Engineering***

|  |  |
| --- | --- |
| **Submited *by*** |  |

**Shreetama Mukherjee - 19101106026**

**Akash Kumar Shaw - 19101106027**

**Abhishek Srivastav - 20101106095**

**Shouvik De - 20101106088**

Supervisor:

**Dr. Subrata Kumar Mandal**



**Department of Information Technology Engineering**

### Jalpaiguri Government Engineering College Jalpaiguri\_735102 (India)

10th June 2023

**Certificate**

Department of Information Technology Engineering Jalpaiguri Government Engineering College, Jalpaiguri

It is certified that the work contained in the project report entitled “**Hand Sign Language Recognition using Machine Learning and Computer Vision”**

by the following students has been carried out under my/our supervision and that this work has not been submitted elsewhere for a degree.

Shreetama Mukherjee - 19101106026

Akash Kumar Shaw - 19101106027

Abhishek Srivastav - 20101106095

Shouvik De - 20101106088

**Dr. Subrata Kumar Mandal**

This project report entitled “**Hand Sign Language Recognition using Machine Learning and Computer Vision**” sub- mitted by the group is approved for the degree of Bachelor of Technology.

|  |
| --- |
|  |
|
| Date: |

The viva-voce examination has been held on .

. .

Examiner(s) Head, Dept. of Information Technology.



# 

# **Declaration**

Jalpaiguri Government Engineering College

10th June 2023

We declare that this written submission represents our ideas in our own words and where others’ ideas or words have been included, We have adequately cited and referenced the original sources. We declare that We have properly and accurately acknowledged all sources used in the production of this report. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Shreetama Mukherjee - 19101106026

Akash Kumar Shaw - 19101106027

Abhishek Srivastav - 20101106095

Shouvik De - 20101106088



# **Abstract**

This report proposes a Machine learning and Computer Vision based solution to the challenges faced by deaf and hard-of-hearing community due to the lack of understanding of sign language by the general public.

The proposed system allows an essential tool for communication for these individuals, and the inability of others to comprehend it can lead to social exclusion and communication barriers in various aspects of life, such as education, employment, and social interactions.

**Table of Contents**



Introduction

Hand Sign Language Translation using Computer Vision is a technology that aims to recognize and translate hand signs into spoken or written language without relying on neural networks. This technology uses computer vision techniques to detect and track hand movements in real-time video streams, and then applies image processing algorithms to recognize the hand signs.

This approach typically involves detecting the hand region, segmenting the hand from the background, and extracting relevant features, such as hand shape and orientation. These features are then matched with a pre-defined dictionary of hand signs to recognize the sign and translate it into spoken or written language.

# **PROPOSED WORK**

**1.Feature extraction**: Developing more robust and efficient feature extraction methods for recognizing hand signs, such as using shape analysis, texture analysis, or motion analysis.

**2.Sign dictionary:** Developing a comprehensive sign dictionary that includes a wider range of hand signs, including more complex signs and phrases.

**3.Real-time tracking:** Developing more advanced real-time tracking methods that can track hand movements in 3D space, which could help to improve the recognition accuracy and robustness of the system.

**4.User interface:** Developing a more user-friendly and accessible interface for the system, such as a mobile application or a web-based platform.

**5.Multilingual support:** Expanding the system to recognize and translate other sign languages used around the world, which could improve communication access for the global deaf and hard-of-hearing community.



ONE STAGE FREE DETECTORS

Where Object detection is a simple regression problem that takes input ans learns probability classes and bounding box coordinators. YOLO, YOLO V2, SSD, RetinaNet etc. fall under one phase detector. Object detection is a advance form of imaging classification where a neural network predicts objects in an image and draws attention to them in the form of bounding boxes.

SSD (SINGLE SHOT DETECTOR)

The term SSD stands for Single Shot Detector. The SSD technique is based on a forward convolutional network that generates a collection of fixed-size bounding boxes and a score for the presence of object class instances in those boxes. The main features of SSD are speed, high accuracy, and learning ability.

Speed – This algorithm has improved speed for real-time object detection.

High Accuracy – This Technique gives accurate Outcomes with minimal background errors.

Learning Ability – The excellent learning ability Of this algorithm allows it to learn object Representations and apply them in object detection.

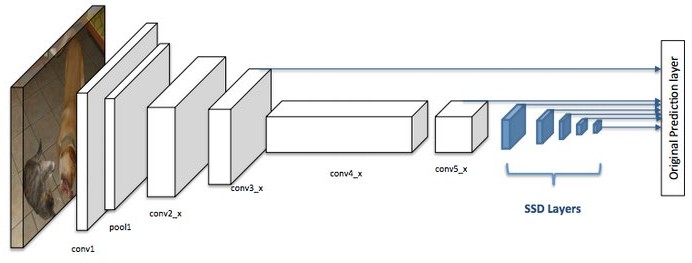




MOBILENET SSD

Mobilenet SSD is an object detection model that computes the output bounding box and object class from the input image. This Single Shot Detector (SSD) object detection model uses Mobilenet as a backbone and can achieve fast object detection optimized for mobile devices.

Convolutional neural networks are used to develop a model that consists of multiple layers for classifying given objects into one of the defined classes. These objects are detected using higher resolution feature maps made possible by recent advances in deep learning with image processing.

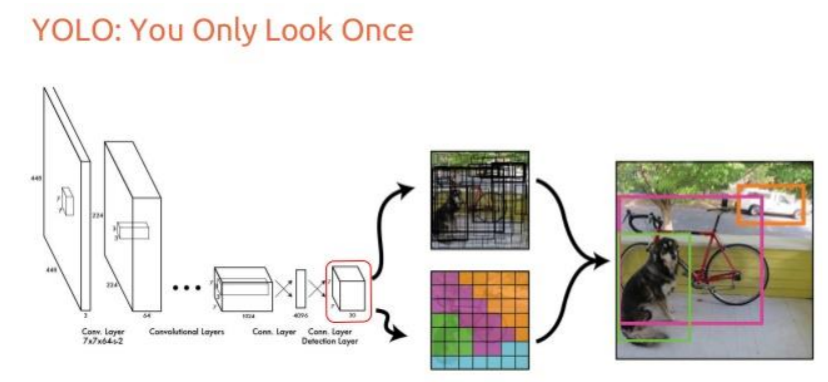


# YOLO (You Only Look Once)

It is a method / way to do object detection. It is the algorithm /strategy behind how the code is going to detect objects in the image.

Earlier Detection frameworks, looked at different parts of the image multiple times at different scales and repurposed image classification technique to detect objects. This approach is slow and inefficient.

YOLO takes entirely different approach. It looks at the entire image only once and goes through the network once and detects objects. Hence the name. It is very fast. That’s the reason it has got so popular.



Project Requirements

**1.CVZONE (Version 1.5.6) :** It serves as a place where individuals interested in computer vision algorithm, techniques, applications and research.CV Zone may include forums, blogs, tutorials, code repositories and resource.

**2.Ultralytics(Version 8.0.26) :** Ultralytics YOLOv8 is a cutting-edge, state-of-the-art (SOTA) model that builds upon the success of previous YOLO versions and introduces new features and improvements to further boost performance and flexibility.

**3.Hydra-core (Version 1.2.0) :** Hydra is an open-source Python framework that simplifies the development of research and other complex applications. The key feature is the ability to dynamically create a hierarchical configuration by composition and override it through config files and the command line.

**4.Matplotlib(version 3.2.2) :** Matplotlib is a low level graph plotting library in python that serves as a visualization utility. Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Create publication quality plots. Make interactive figures that can zoom, pan, update.

**5.Numpy(Version 1.18.5) :** NumPy is a Python library used for working with arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

**6.OpenCV-Python(Version 4.5.4.60) :** OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

**7.Pillow(Version 7.1.2) :** The Python Imaging Library adds image processing capabilities to your Python interpreter.

This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities.

The core image library is designed for fast access to data stored in a few basic pixel formats. It should provide a solid foundation for a general image processing tool.

**8.PyYAML(Version 5.3.1) :** YAML is a data serialization format designed for human readability and interaction with scripting languages. PyYAML is a YAML parser and emitter for Python.

PyYAML features a complete YAML 1.1 parser, Unicode support, pickle support, capable extension API, and sensible error messages. PyYAML supports standard YAML tags and provides Python-specific tags that allow to represent an arbitrary Python object.

PyYAML is applicable for a broad range of tasks from complex configuration files to object serialization and persistence.

**9. Request(Version 2.23.0) :** The Request library is the de facto standard for making HTTP requests in Python. It abstracts the complexities of making requests behind a beautiful, simple API so that you can focus on interacting with services and consuming data in your application.

**10.Scipy(Version 1.4.1) :** The SciPy is an open-source scientific library of Python that is distributed under a BSD license. It is used to solve the complex scientific and mathematical problems. It is built on top of the Numpy extension, which means if we import the SciPy, there is no need to import Numpy.

**11.PyTorch(Version 1.7.0) :** PyTorch is a machine learning framework based on the Torch library, used for applications such as computer vision and natural language processing.



**12.TorchVision (Version 0.8.1) :** The torchvision library consists of popular datasets, model architectures, and image transformations for computer vision. It consists of: Training recipes for object detection, image classification, instance segmentation, video classification and semantic segmentation.

**13.tqdm(Version 4.64.0) :** The range of tqdm in Python refers to the number of iterations or elements in the iterable that the progress bar will track. It can be specified using Python's range() function or any other iterable object.

**14.Filterpy(Version 1.4.5) :** Filterpy is a Python package that provides a set of tools for implementing and testing Kalman filters, a type of algorithm used for state estimation and tracking in a wide range of applications.

**15.Scikit-image (Version 0.19.3) :** scikit-image is an open-source image processing library for the Python programming language. It includes algorithms for segmentation, geometric transformations, color space manipulation, analysis, filtering, morphology, feature detection, and more.

**16.Lap(Version 0.4.0) : lap** is a Linear Assignment Problem solver using Jonker-Volgenant algorithm for dense matrics. Select a recommended open source package. Minimize your risk by selecting secure & well maintained open source packages.



**Code**



Results and Discussion



**Observations**

* SSD performs worse than Faster R-CNN for small-scale objects. In SSD, small objects can only be detected in higher resolution layers (leftmost layers). But those layers contain low-level features, like edges or color patches, that are less informative for classification.
* Accuracy increases with the number of default boundary boxes at the cost of speed.
* Multi-scale feature maps improve the detection of objects at a different scale.
* Design better default boundary boxes will help accuracy.
* SSD has lower localization error comparing with R-CNN but more classification error dealing with similar categories. The higher classification errors are likely because we use the same boundary box to make multiple class predictions.
* SSD512 has better accuracy (2.5%) than SSD300 but run at 22 FPS instead of 59.



**FUTURE WORK**

* Future work could involve further improving the recognition accuracy of the proposed method, particularly for more complex hand signs. This could involve exploring more advanced computer vision techniques, such as deep learning approaches or 3D tracking of hand movements.
* The system will be able to accurately recognize and translate the hand signs in real-time, with a very low latency.

 **CONCLUSION**

After completing this project“**Hand Sign Language Recognition using Machine Learning and Computer Vision”** We gaintremendous knowledge on Machine Learning & Computer Vision Application in Real Life.

In conclusion, hand sign recognition using computer vision is a complex and challenging task, but one with great potential for impact. With continuous testing and evaluation, attention to important factors affecting the system's performance, and consideration of potential applications, it is possible to create a robust and useful hand sign recognition system.