**CSE3013 – ARTIFICIAL INTELLIGENCE**

# J Component Report

GESTURE DETECTION USING MEDIAPIPE

*By*

20BCE1740 – PREETI SAI THAANDAVAN

20BCE1681 – ASHAR IRRFANA H

20BCE1787 – TARAN AKSHAY S

BACHELOR OF TECHNOLOGY IN

COMPUTER SCIENCE AND ENGINEERING

*Submitted to*

# PRADEEP K.V

### School of Computer Science and Engineering



*April 2023*

**DECLARATION BY THE CANDIDATE**

I hereby declare that the report titled "**GESTURE DETECTION USING MEDIAPIPE**" submitted to VIT Chennai is a record of bona-fide work undertaken under the supervision of **Dr.Pradeep K.V, Associate Professor, SCOPE, Vellore Institute of Technology, Chennai.**

**Signature of the Candidate**

## ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr.Pradeep KV,** School of Computer Science and Engineering for her consistent encouragement and valuable guidance offered to us throughout the course of the project work.

We are extremely grateful to **Dr. R. Ganesan, Dean,** School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology, Chennai, for extending unstinting support towards our project.

We express our thanks to our **Head of the Department** for his support throughout the course of this project. We also take this opportunity to thank all the faculty for their support and their wisdom imparted to us throughout the courses.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

## BONAFIDE CERTIFICATE

Certified that this project report entitled "**GESTURE DETECTION USING MEDIAPIPE**" is a bona-fide work of **Preeti Sai Thandavan, Ashar Irrfana H, Taran Akshay S**carried out under my supervision and guidance for CSE3013- Artificial Intelligence

## Dr. Pradeep K.V

SCOPE

### ABSTRACT:

For human-computer interaction, hand gesture recognition is essential.

Gesture recognition is the process of determining the user's gestures using a computer. Many hand gesture recognition technologies have been developed up to this point, but Video-Based Hand Gesture Detection with high efficiency and low computational cost remains a difficult task.

The goal is to create a cutting-edge prototype for creating a real-time online hand gesture detection system with minimal latency and great efficiency. At any moment, a user can make a hand gesture in front of a web camera that is coupled to high-end computing that captures motions and analyses and recognizes them. The gesture is recognized and displayed on the process's screen. For online action detection, the recognition procedure must be quick and precise. A feature extraction tool Media pipe has been used for extracting the co-ordinates from the data points of the human hand made into a dataset which is acted upon by several models and the performance of the model is measured through accuracy of classification and the results are presented.

**Keyword:** Mediapipe, hand-gesture detection.

### INTRODUCTION:

There exist many cases where a deaf person may not know

the sign language, or a normal person doesn't understand the sign language used by the dumb person, or even the deaf/dumb students who have not yet learned the sign language.

In order to solve this issue, we have come up with a new and a more efficient method (using Mediapipe to extract the co-ordinates of the hand- landmarks alone instead of using CNN-based models (for e.g., Yolo V5) which uses all the pixels of the image to process it and detect the gesture). MediaPipe is an open-source framework that provides pre-built solutions for various tasks such as hand tracking, face detection, and pose estimation. In the case of hand gesture recognition, MediaPipe provides a hand tracking solution that can be used to detect the position of the hand and the landmarks of the fingers.

The hand tracking solution in MediaPipe uses a machine learning model to predict the hand landmarks based on the input video frames. The model is trained on a large dataset of hand images and is designed to work in real-time on mobile and desktop platforms.

The hand landmarks detected by the model can be used to recognize various hand gestures. For example, a closed hand with the thumb sticking out can be recognized as a "thumbs up" gesture, while a spread-out hand can be recognized as a "stop" gesture.

To use the MediaPipe hand tracking solution for hand gesture recognition, one would need to write a script that captures video frames, passes them through the MediaPipe hand tracking model, and then uses a gesture recognition algorithm to identify the hand gestures.

Overall, hand gesture recognition using MediaPipe is a powerful and flexible solution that can be used in a wide range of applications

Top of Form

### LITERATURE REVIEW:

1. **A Deep Convolutional Neural Network Approach for Static Hand Gesture Recognition,** Adithya V., Rajesh R, Volume 171, 2020, Pages 2353-2361, ISSN: 1877-0509

The deaf-dumb community's main social issues are the communication barrier and the hearing majority, both of which prohibit them from receiving basic and needed life services. Despite advances in automatic sign language identification, an effective solution has yet to be found due to a variety of vexing aspects. The majority of present research attempts to construct vision- based recognizers using a traditional pattern analysis technique, which involves extracting sophisticated handcrafted feature descriptors from collected pictures of gestures. However, when working with a vast sign vocabulary collected under complicated and uncontrolled backdrop settings, the effectiveness of such approaches is severely constrained. Based on an efficient deep convolutional neural network (CNN) architecture, this study provides a methodology for the recognition of hand movements, which are the primary component in sign language lexicon. The strategy was evaluated on two publicly available datasets (the NUS hand posture dataset and the American fingerspelling A dataset) and was shown to improve recognition accuracy.

[**https://www.sciencedirect.com/science/article/pii/S1877050920312473?via%**](https://www.sciencedirect.com/science/article/pii/S1877050920312473?via%3Dihub)[**3Dihub**](https://www.sciencedirect.com/science/article/pii/S1877050920312473?via%3Dihub)

identification, an effective solution has yet to be found due to a variety of vexing aspects. The majority of present research attempts to construct vision- based recognizers using a traditional pattern analysis technique, which involves extracting sophisticated handcrafted feature descriptors from collected pictures of gestures. However, when working with a vast sign vocabulary collected under complicated and uncontrolled backdrop settings, the effectiveness of such approaches is severely constrained. Based on an efficient deep convolutional neural network (CNN) architecture, this study provides a methodology for the recognition of hand movements, which are the primary component in sign language lexicon. The strategy was evaluated on two publicly available datasets (the NUS hand posture dataset and the American fingerspelling A dataset) and was shown to improve recognition accuracy.

[**https://www.sciencedirect.com/science/article/pii/S1877050920312473?via%**](https://www.sciencedirect.com/science/article/pii/S1877050920312473?via%3Dihub)[**3Dihub**](https://www.sciencedirect.com/science/article/pii/S1877050920312473?via%3Dihub)

1. **Deep Learning Approach in Intra -Prediction of High Efficiency Video Coding,** Joy, Helen & Kounte, Manjunath R & Joy, Ajin, 2020, Pages 134-138, Electronic ISBN: 978-1-7281-7213-2.

CTU is HEVC's fundamental processing unit. It comes in a variety of sizes ranging from 6464 to 808 bytes, and as the size grows larger, the coding efficiency improves. The computational complexity is a concern that should be addressed, as HEVC offers several advantages that make it a viable video compression approach. This work uses a combination of depth decision and deep learning approaches to reduce the computational complexity of high- efficiency video coding (HEVC) in intra prediction. From the database supplied, the suggested technique produces a neural network for CTU depth analysis, followed by a deep learning network with several sizes of kernels for convolution and trainable ubiquitous parameters. The picture frame data in the database indicates the image value of CU and a 16x1 vector based on CU's encoding specifications. It features a label that says whether or not the CU is divided. Initially, a large picture frame is sorted into various sizes and a split is formed. The divisions were then turned into a three-level classification issue.

To answer the classification problem, a deep learning-based CNN structure with varied size kernels and convolution parameters is constructed, which should be examined and learnt using a database. For the supplied database, the findings demonstrate a decrease in intra mode HEVC encoding time. [**https://ieeexplore.ieee.org/document/9277189**](https://ieeexplore.ieee.org/document/9277189)

1. **Deep Learning-Based Fast Hand Gesture Recognition Using Representative Frames,** V. John, A. Boyali, S. Mita, M. Imanishi and N. Sanma, 2016, Pages 1-8, Electronic ISBN: 978-1-5090-2896-2.

We provide a vision-based hand gesture recognition system for intelligent cars in this study. In vehicle user interfaces, vision-based gesture recognition technologies are used to improve driver comfort without jeopardizing their safety. The long-term recurrent convolution network is employed in our technique to categorize video sequences of hand motions. Multiple frames taken from the video sequence are sent into the network to conduct classification in the usual long-term recurrent convolution network-based action classifier. However, using additional frames increases computing complexity while also lowering the classifier's classification accuracy. To overcome these difficulties, we suggest collecting a smaller number of representative frames from the video stream and feeding them into a long-term recurrent convolution network. We propose using innovative tiled picture patterns and tiled binary patterns inside a semantic segmentation-based deep learning framework, the deconvolutional neural network, to extract sample frames. The new tiled picture patterns are made up of many non-overlapping blocks that portray the whole gesture-based video sequence in a single image. These picture patterns are created from the video sequence and used as input to the deconvolution network. The unique tiled binary pattern also includes many non-overlapping blocks that represent the video sequence's representative frames. The deconvolution network's output is represented by these binary patterns. The dictionary learning and sparse modelling framework is used to build the training binary patterns from the training video sequences. On the public Cambridge gesture recognition dataset, we test our suggested approach. When compared to baseline methods, there is a significant improvement in classification accuracy. The suggested method is also subjected to a thorough parametric examination. We present a 91 percent accuracy in gesture categorization and a computational complexity of $110$ms per video sequence in near real-time.

[**https://ieeexplore.ieee.org/document/7797030**](https://ieeexplore.ieee.org/document/7797030)

1. **Real-time American Sign Language Recognition with Convolutional Neural Networks,** Garcia, Brandon, and Sigberto Alarcon Viesca, Convolutional Neural Networks for Visual Recognition 2, 2016, Pages: 225-232

The development of a real-time sign language translator is a significant step forward in improving communication between the deaf community and the general public. Creation and implementation of an American Sign Language fingerspelling translator for the American Sign Language (ASL) based on a a Convolutional Neural Network (CNN) is a type of neural network that uses we make use of a well-trained team. The ILSVRC2012 was used to train the GoogLeNet architecture. as well as the Surrey University and Massey University's ASL datasets in order to apply Transfer Learning to this task. With

first-time users, our model consistently classifies letters a-e correctly and another model that detects letters a-k correctly in the vast majority of cases given the datasets limitations and we are convinced that, as a result of the good results obtained, we can make a better product with additional research and data.

[**http://vision.stanford.edu/teaching/cs231n/reports/2016/pdfs/214\_Report.pd**](http://vision.stanford.edu/teaching/cs231n/reports/2016/pdfs/214_Report.pdf)[**f**](http://vision.stanford.edu/teaching/cs231n/reports/2016/pdfs/214_Report.pdf)

1. **Sign Language Recognition using Convolutional Neural Networks,** Pigou, Lionel, Sander Dieleman, Pieter-Jan Kindermans, and Benjamin Schrauwen, European Conference on Computer Vision, 2014, Pages: 572-578

The Deaf population and the hearing majority have an undeniable communication challenge. Automatic sign language recognition innovations are attempting to break down this communication barrier. A recognition system based on the Microsoft Kinect, convolutional neural networks (CNNs), and GPU acceleration is the focus of this contribution. CNNs can automate the process of feature building rather than creating intricate handcrafted features.

They have a high level of accuracy in recognising 20 Italian gestures. With a cross-validation accuracy of 91.7 percent, the predictive model may generalise to users and environments that were not present during training. In the ChaLearn 2014 Looking at People gesture spotting competition, this model received a mean Jaccard Index of 0.789. [**https://link.springer.com/chapter/10.1007/978-3-319-16178-5\_40**](https://link.springer.com/chapter/10.1007/978-3-319-16178-5_40)

1. **Static Sign Language Recognition using Deep Learning,** Tolentino, Lean Karlo S., Ronnie O. Serfa Juan, August C. Thio-ac, Maria Abigail B. Pamahoy, Joni Rose R. Forteza, and Xavier Jet O. Garcia, International Journal of Machine Learning and Computing 9, no. 6, 2019, Pages: 821-827

A system that uses hand detection as a learning tool for beginners in sign language has been developed. This system is based on explicit skin-color space thresholding, which is a skin-color modelling technique. The skin-color range has been predetermined to separate pixels (hand) from non-pixels (background). For image categorization, the photos were loaded into a model called the Convolutional Neural Network (CNN). Keras was utilised for image training. The system achieved an average testing accuracy of 93.67 percent when given suitable illumination and a uniform background, with 90.04 percent credited to ASL alphabet recognition, 93.44 percent for number recognition, and 97.52 percent for static word identification, outperforming prior related studies. The method is used to perform quick computations.

[**https://www.researchgate.net/publication/337285019\_Static\_Sign\_Langua**](https://www.researchgate.net/publication/337285019_Static_Sign_Language_Recognition_Using_Deep_Learning)[**ge\_Recognition\_Using\_Deep\_Learning**](https://www.researchgate.net/publication/337285019_Static_Sign_Language_Recognition_Using_Deep_Learning)

1. **Arabic Sign Language Recognition System for Alphabets Using Machine Learning Techniques,** Gamal Tharwat , Abdelmoty M. Ahmed , and Belgacem Bouallegue, Procedia Computer Science, Volume 171, 2021, Pages 1-17

In recent years, the role of pattern recognition in human-computer interaction (HCI) systems has grown in terms of computer vision applications and machine learning, with one of the most important of these applications being the recognition of deaf people's hand gestures, particularly the dashed letters in Quran surahs. We propose an Arabic Alphabet Sign Language Recognition System based on a vision-based technique in this research. The suggested system is divided into four stages: data processing, data preprocessing, feature extraction, and classification. This system works with three sorts of datasets: data involving naked hands and a dark background, data involving bare hands but a light background, and data involving bare hands but a light background.

The system works with three sorts of datasets: data about hands wearing dark- colored gloves, data about hands wearing light-colored gloves, and data about hands wearing light-colored gloves. AArSLRS starts with obtaining an image of the alphabet movements, then revealing the hand from the image and isolating it from the background using one of the offered approaches, and then extracting the hand features using the selection method. We employed supervised learning approaches for the categorization of the 28-letter Arabic alphabet employing 9240 photos in our system's classification process. In the Quranic sign language, we focused on the classification of 14 alphabetic letters that represent the first Quran surahs (QSL). For the K-Nearest Neighbor (KNN) classifier, AArSLRS attained a 99.5 percent accuracy. [**https://www.researchgate.net/publication/352787419\_Arabic\_Sign\_Langu**](https://www.researchgate.net/publication/352787419_Arabic_Sign_Language_Recognition_A_Review)[**age\_Recognition\_A\_Review**](https://www.researchgate.net/publication/352787419_Arabic_Sign_Language_Recognition_A_Review)

1. **Sign Language Recognition using CNN,** Satwik Ram Kodandaram, N. Pavan Kumar, Sunil Gl, 2021

Deaf (hard of hearing) and dumb individuals mostly utilise sign language to communicate within their communities and with others. It's a kind of communication in which people communicate by hand gestures because they can't talk or hear. SLR is concerned with recognising the acquisition of hand gestures and continues until text or voice is generated for the associated hand motions. Hand motions for sign language can be divided into two categories: static and dynamic. Although static hand gesture identification is easier than dynamic hand gesture recognition, both are critical to the human community. We may utilise Deep Learning Computer Vision to recognise hand gestures by

creating Deep Neural Network designs (Convolution Neural Network Architectures), in which the model learns to recognise hand motions over time. After the model correctly recognises the gesture, an English text is created, which may subsequently be turned to speech. This model will be more efficient, making it simpler to communicate with the deaf (hard of hearing) and disabled people. In this study, we'll look at how Deep Learning is used to recognise Sign Language. [**https://www.researchgate.net/publication/353141966\_Sign\_Language\_Rec**](https://www.researchgate.net/publication/353141966_Sign_Language_Recognition)[**ognition**](https://www.researchgate.net/publication/353141966_Sign_Language_Recognition)

1. **Generalization of Bangla Sign Language Recognition using Angular Loss Functions,** Samiya Kabir Youme, Towsif Alam Chowdhury, Hossain Ahamed, Md. Sayeed Abid, Labib Chowdhury, and Nabeel Mohammed, 2021, Pages 1- 17

Deaf and mute people can communicate with each other using Sign Language. Because Sign Language is not widely understood, effective communication with the general public is a significant barrier for the deaf and mute community. Many academics have worked on datasets in foreign languages like as English, French, and Japanese. However, much work remains to be done in Bangla, one of the world's most commonly spoken languages. The majority of Bangla Sign Language research is done on small datasets and yields excellent results. However, when limited datasets are reviewed for generalizability, especially when utilizing deep learning-based solutions, these models do not reproduce to the same level of performance. This analysis is carried out for Bangla using two widely used Bangla Sign Language datasets. The inter-dataset performance is unsurprisingly poor, and numerous ways to enhance it are investigated and reported, including the usage of angular margin based loss functions. The findings indicate the need of such an assessment and show that one of the proposed ways performs well, albeit there is still space for improvement. This highlights the necessity for a standardized dataset to address the issue of generalization in real-world applications, as well as the need to encourage future research to focus on difficult evaluations rather than deceptively impressive intra-dataset performance. [**https://www.researchgate.net/publication/356948321\_Generalization\_of\_B**](https://www.researchgate.net/publication/356948321_Generalization_of_Bangla_Sign_Language_Recognition_Using_Angular_Loss_Functions)[**angla\_Sign\_Language\_Recognition\_Using\_Angular\_Loss\_Functions**](https://www.researchgate.net/publication/356948321_Generalization_of_Bangla_Sign_Language_Recognition_Using_Angular_Loss_Functions)

1. **A Pattern Recognition Model for Static Gestures in Malaysian Sign Language Based on Machine Learning Techniques,** Ali.H.Alrubayib, M.A.Ahmed, A.A.Zaidana, A.S.Albahri, B.B.Zaidan, O.S.Albahri, A.H.Alamoodi, Mamoun Alazabe, Year: 2021, Pages: 1 – 15, Article ID:107383

Not every person can pay attention or speak, and a few be afflicted by associated impairments. Such people use unique conversation approaches rather than their voices to engage with society. Individuals with listening to and speech impairments depend upon hand gestures and the accompanying actions to deliver their meant ideas. These gestures, that are established with syntax, grammar, semantics, pragmatics and morphology are called signal language (SL). The SL has end up the herbal language for individuals who are deaf, mute, or produce other listening to impairments. This work proposes a pattern recognition model for static gestures in Malaysian Sign Language (MSL) based on Machine Learning (ML) techniques. The proposed model is divided into two phases, namely, data acquisition and data processing. The first phase involves capturing the required sign data, such as the shape and orientation of the hand, to construct a sensor based SL dataset. The dataset is collected using a DataGlove device. This device is used to measure the motions of the fingers and wrists. 64 functions represent each character in the dataset. The collected sensory datasets are cleaned up in the second phase by removing the redundant data. The features are then scaled and normalized to show symmetric behavior and eliminate outliers. Next, 10 different ML techniques based on real-time data are used for SL gesture recognition. The experimental results confirmed the effectiveness of the proposed pattern recognition model in comparison with previous studies. [**https://www.sciencedirect.com/science/article/abs/pii/S0045790621003529**](https://www.sciencedirect.com/science/article/abs/pii/S0045790621003529)

1. **Computer Vision-Enabled Character Recognition of Hand Gestures for Patients with Hearing and Speaking Disability,** Sapna Juneja, Abhinav Juneja, Gaurav Dhiman, Shashank Jain, Anu Dhankhar, and Sandeep Kautish, Year: 2021, Pages: 1 – 10, Article ID: 4912486

Machine studying isn't a brand new era that has lately evolved. There has been a quest to make the arena higher the usage of synthetic intelligence and system studying because the 1970s. Wang exhibited flip invariant stances using restrict histogram. A digital digicam has been used to steady the facts photo. The channel for pores and skin shading discovery is being used by the bunching manner to find out the restrict for every accumulating withinside the grouped photo, using a not unusualplace shape following calculation. The photo turned into divided into a couple of networks, and the bounds had been normalized.

The restrict turned into known as harmony`s length chain which turned into applied withinside the shape of a histogram, via way of means of setting apart the photo into some of regions N in an outspread structure, in keeping with the express edge. For the type process, neural networks, MLP and dynamic programming, and DP coordinating had been applied. Numerous analyses were carried out on diverse spotlight positions however using unique harmony`s length histogram and harmony`s length FFT. Convolution neural networks have a longtime function in photograph reputation as has been evidentially

validated via way of means of numerous researchers in past. The particular contribution of CNNs to clinical sickness analysis with the assist of referring to the scanned photos to the presence or absence of diseases is an superb software with confirmed performance and reliability. ReLU is one of the maximum famous nonlinear activation capabilities depended on via way of means of researchers for deep studying projects. 26 static stances from American signal language had been applied withinside the trials. A homogeneous basis turned into implemented withinside the work. Stergiopoulou proposed every other self-developing and self-prepared neural gas (SONG) network for hand movement acknowledgment. For hand district discovery, a shading department

manner depending on the pores and skin shading channel withinside the YCbCr shading space turned into applied, and an estimation of hand-fashioned morphology has been diagnosed using SONG organize; 3 highlights had been extricated, using the finger-distinguishing evidence manner that makes a decision the amount of the lifted palms and fine of form of the hand, and the Gaussian move version turned into applied for acknowledgment. [**https://www.hindawi.com/journals/misy/2021/4912486/**](https://www.hindawi.com/journals/misy/2021/4912486/)

1. **Training of a Deep Learning Algorithm for Quadcopter Gesture Recognition,** Calvin Ng, Alvin Chua, Year: 2020, Pages: 1 – 6, Electronic ISBN: 978-1-5386-2092-2

Traditional methods to control Unmanned Aerial Vehicles are unintuitive and susceptible to radio interference. Recent research has shown that hand gestures are the most intuitive method for quadcopter control. Also, deep learning in the form of a convolutional neural network is a more compatible approach to gesture recognition than other methods. This paper presents the design, and training of a deep learning convolutional neural network for gesture recognition and tracking of a quadrotor Unmanned Aerial Vehicle. The neural network was coded in Python using the Keras library and was trained on a laptop computer. Inference was performed on a Raspberry Pi 4 computer that is intended for use as a companion computer aboard a quadcopter. [**https://ieeexplore.ieee.org/abstract/document/8336782/authors#authors**](https://ieeexplore.ieee.org/abstract/document/8336782/authors)

1. **MediaPipe: A Framework for Perceiving and Processing Reality,** Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg and Matthias Grundmann, Year:2019, Pages: 1 – 4

More than just running an ML model is required to create an application that handles perceptual inputs. Developers must take use of a wide range of devices' capabilities, balance resource utilisation and result quality, conduct numerous processes in parallel and with pipelining, and guarantee that time-series data is

appropriately synced. These issues are addressed by the MediaPipe architecture. Individual perceptual models are abstracted and connected into maintainable pipelines in MediaPipe, which overcomes these difficulties. A perception pipeline may be designed using MediaPipe as a graph of modular components, such as inference models and media processing algorithms.Sensory data such as audio and video streams enter the graph, and perceived descriptions such as object detection results or face landmark annotations leave the graph.MediaPipe consists of three main parts: (1) a framework for inference from sensory data, (2) a set of tools for performance evaluation, (3) a collection of reusable inference and processing components. Real-time object recognition from a live camera feed is a typical need for AR applications. ML-based object identification at full frame rate might be resource intensive or impossible to implement owing to large inference times, depending on the target device platform. Another typical perception application is face landmark estimation. MediaPipe makes it easy to build a perception pipeline, optimize and improve it using its rich configuration language and performance evaluation tools.The pipelines can run on a variety of platforms, enabling the developer to build the application on workstations and then deploy it on mobile. A key element of MediaPipe’s success is the ecosystem of reusable calculators and graphs. [**https://static1.squarespace.com/static/5c3f69e1cc8fedbc039ea739/t/5e130ff**](https://static1.squarespace.com/static/5c3f69e1cc8fedbc039ea739/t/5e130ff310a69061a71cbd7c/1578307584840/NewTitle_May1_MediaPipe_CVPR_CV4ARVR_Workshop_2019.pdf)[**310a69061a71cbd7c/1578307584840/NewTitle\_May1\_MediaPipe\_CVPR\_**](https://static1.squarespace.com/static/5c3f69e1cc8fedbc039ea739/t/5e130ff310a69061a71cbd7c/1578307584840/NewTitle_May1_MediaPipe_CVPR_CV4ARVR_Workshop_2019.pdf)[**CV4ARVR\_Workshop\_2019.pdf**](https://static1.squarespace.com/static/5c3f69e1cc8fedbc039ea739/t/5e130ff310a69061a71cbd7c/1578307584840/NewTitle_May1_MediaPipe_CVPR_CV4ARVR_Workshop_2019.pdf)

1. **MediaPipe Hands: On-device Real-time Hand Tracking,** Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, A. Tkachenka, George

Sung, Chuo-Ling Chang, Matthias Grundmann, Year: 2020, Pages: 1 – 5

A real-time on-device hand tracking solution that predicts a hand skeleton of a human from a single RGB camera for AR/VR applications has been presented. Hand tracking is a critical component for providing a natural approach for AR/VR engagement and communication, and it's been a hot issue in the industry for quite some time. The hand tracking method makes use of a machine learning pipeline that consists of two models that operate together: A palm detector that locates palms using an aligned hand bounding box and operates on a whole input picture. A hand landmark model that uses the palm detector's clipped hand bounding box to generate high-fidelity 2.5D landmarks. They have used a single-shot detector model intended for mobile real-time applications, comparable to BlazeFace, which is also available in MediaPipe, to identify initial hand placements. First, they trained a palm detector instead of a hand detector, since estimating bounding boxes of rigid objects like palms and fists is significantly simpler than detecting hands with articulated fingers. To obtain ground truth data, many datasets have been used. The wild dataset contains 6K photos with a wide range of characteristics, such

as geographical diversity, lighting circumstances, and hand appearance. House gesture dataset: This collection comprises 10K photos that depict all physically conceivable hand movements from various angles. They render a high-quality synthetic hand model over diverse backdrops and map it to the associated 3D coordinates to better cover all potential hand positions and give extra depth monitoring. With MediaPipe, hand tracking pipeline can be built as a directed graph of modular components, called Calculators. Mediapipe comes with an extensible set of Calculators to solve tasks like model inference, media processing, and data transformations across a wide variety of devices and platforms. This pipeline predicts 2.5D landmarks without any specialized hardware and thus, can be easily deployed to commodity devices. [**https://arxiv.org/pdf/2006.10214.pdf**](https://arxiv.org/pdf/2006.10214.pdf)

1. **Feature Extraction Technique for Static Hand Gesture Recognition,** Haitham Badi, Sameem Abdul Kareem and Sabah Husien, Year: 2015, Pages: 1 – 23

The purpose of static hand gesture recognition is to categorise provided hand gesture data represented by specific attributes into a finite number of preset gesture classes. The major goal of this project is to investigate the efficacy of two feature extraction approaches, namely hand contour and complex moments, in solving the problem of hand gesture identification by identifying the fundamental benefits and drawbacks of each method.The author designed an interactive environment called computer-controlled responsive environment, a space within which everything the user saw or heard was in response to what he/she did. Rather than sitting down and moving only the user's fingers, he/she interacted with his/her body. The glove-based approach employs sensors (mechanical or optical) attached to a glove that acts as transducer of finger flexion into electrical signals to determine hand posture. The second approach, vision based analysis, is based on how humans perceive information about their surroundings. Another method extracted the features from color images as in

[37] where they pre- sented a real-time static isolated gesture recognition application using a hidden Markov model approach. The pro- cess of hand gesture recognition composes mainly of four stages: (1) hand gesture images collection, (2) gesture image preprocessing using some techniques including edge-detection, fltering and normalization, (3) capture the main characteristics of the gesture images using feature extraction algorithms, and (4) the evaluation (or classification) stage where the image is classified to its corresponding gesture class. There are many methods that have been used in the classifcation stage of hand gesture recognition such as Artificial Neural Networks, Template Matching, Hidden Markov Models and Dynamic Time Warping. [**https://sciencegatepub.com/books/gcsr/gcsr\_vol3/GCSR\_Vol3\_Ch2.pdf**](https://sciencegatepub.com/books/gcsr/gcsr_vol3/GCSR_Vol3_Ch2.pdf)

# EXISTING ARCHITECTURE

### CNN (Convolution Neural Network)

Convolutional Neural Networks (CNNs) are a common architecture choice for hand gesture identification because of their capacity to recognize spatial relationships between pixels and learn features from images. An illustration of a CNN architecture that could be applied to hand gesture recognition is shown below:

This layer receives the input image in the input layer. An image of a hand gesture would be the input for hand gesture recognition.

Layers that perform convolution operations on the input image to extract features are known as convolutional layers. The output of each convolutional layer is passed through an activation function (e.g. ReLU) to introduce non-linearity. More complicated features can be extracted using many convolutional layers.

Pooling layers: By down sampling the output of the convolutional layers, these layers lower its spatial dimensions. This makes the model more computationally efficient by lowering the number of parameters in it.

Fully connected layers: These layers classify the data produced by the convolutional and pooling layers. They typically consist of numerous fully linked layers with several neurons per layer.

The final classification output, which corresponds to the identified hand motion, is provided by the output layer.

### RNNs (Recurrent Neural Networks)

Another neural network architecture that can be utilized for hand gesture identification is recurrent neural networks (RNNs). When the input data is a series of frames, like in a video of a hand gesture, RNNs are especially helpful. Here is an illustration of an RNN architecture that might be applied to the recognition of hand gestures:

This layer receives the input frames in the form of an input layer. A series of frames from a video of a hand gesture would be the input in the instance of hand gesture recognition.

The network can take into account the temporal dependencies between the frames in the input sequence using recurrent layers. Each recurrent layer's output is subjected to an activation function (such as tanh) to provide nonlinearity.

Fully linked layers: These layers classify the information produced by the recurrent layers. They typically consist of numerous fully linked layers with several neurons per layer.

The final classification output, which corresponds to the identified hand motion, is provided by the output layer.

# LSTM (Long Short Term Memory Network)

Recurrent neural networks (RNNs) of the sort known as Long Short-Term Memory Networks (LSTMs) are particularly good at simulating long-term dependencies in sequential data. LSTMs can be used to capture the temporal dependencies in a video sequence of hand movements for hand gesture recognition.An illustration of an LSTM architecture that could be applied to hand gesture recognition is shown below:

This layer receives the input frames in the form of an input layer. A series of frames from a video of a hand gesture would be the input in the instance of hand gesture recognition.

LSTM layers: The network can recognise long-term dependencies in the input sequence using these layers. In order to introduce non-linearity, the output of each LSTM layer is routed through an activation function (such as tanh).

Fully connected layers: These layers do categorization on the LSTM layer output. They typically consist of numerous fully linked layers with several neurons per layer.

The final classification output, which corresponds to the identified hand motion, is provided by the output layer.

**DRAWBACKS OF EXISTING ARCHITECTURE**

### CNN (Convolution Neural Network)

Although Convolutional Neural Networks (CNNs) are an effective tool for image processing tasks, employing them for hand motion detection has significant disadvantages.

Restricted Viewpoints: CNNs are trained using pictures taken from particular angles and hand gestures. The CNN might not correctly identify the gesture if a new hand stance or angle is encountered.

Data augmentation: For CNNs to generalise effectively, a lot of training data is needed. Yet, it could be challenging to find a sizable dataset containing a variety of hand positions and angles for hand gesture detection. Techniques for artificially extending the dataset's size can be applied, although they may not always be successful in reproducing the variety of real-world hand motions.

Localization: CNNs are not made to identify where certain items are located inside a picture. Prior to recognising the gesture in hand gesture recognition, it is crucial to precisely localise the hand in the image. This can be difficult, especially if the image contains several hands or other objects.

Difficulty of the computations: CNNs are memory-intensive and expensive to compute. Real-time hand gesture recognition on low-power devices may become challenging as a result.

Restricted Context: CNNs are only able to record local details within an image; they do not take the gesture's environment into account. For instance, the context in which a hand gesture is employed can affect its meaning.

### RNNs (Recurrent Neural Networks)

There are certain restrictions on utilising recurrent neural networks (RNNs) for hand gesture detection, despite the fact that they are useful for modelling temporal relationships in sequential data, such as videos:

Gradient Vanishing/Exploding: RNNs may experience vanishing or exploding gradient issues, which can make network training challenging. When dealing with lengthy sequences of frames, as is frequently the case with hand gesture identification, this issue might become quite significant.

The need to keep a hidden state for each time step in the input sequence makes RNNs memory-intensive. This can make it difficult to train and deploy RNN models on resource-constrained devices.

RNNs are susceptible to overfitting, particularly when working with tiny datasets or intricate models. Poor generalisation performance on unknown data may result from this.

Restricted Context: RNNs are constrained in their capacity to capture context outside of the temporal dimension, similar to CNNs. This may be a drawback in hand gesture identification, because a gesture's meaning depends on its context.

RNN training can take a while, particularly when working with huge datasets or complicated models. This can make it difficult to iterate quickly and explore different model architectures and hyperparameters.

# LSTM (Long Short Term Memory Network)

There are certain restrictions on employing Long Short-Term Memory Networks (LSTMs), a form of Recurrent Neural Network (RNN) that is particularly good at simulating long-term relationships in sequential data:

Memory Constraints: Because LSTMs must keep a hidden state for each time step in the input sequence, they might be memory-intensive. As a result, developing and deploying LSTM models on devices with limited resources may be challenging.

Training Time: When working with huge datasets or complicated models, training LSTMs might take a while. This can make it challenging to investigate various model topologies and hyperparameters fast.

Overfitting is a possibility with LSTMs, particularly when working with limited datasets or intricate models. Poor generalisation performance on unknown data may result from this.

Restricted Context: LSTMs are constrained in their capacity to capture context outside of the temporal dimension, similar to other neural networks. This may be a drawback in hand gesture identification, because a gesture's meaning depends on its context.

LSTMs are built to handle fixed-length sequences, which might be a drawback in hand gesture recognition where the length of the gesture can vary from one instance to another. Trouble Handling Variable-Length Sequences. Variable-length sequences need to be handled with additional preprocessing, such as padding or truncation, which might result in information loss.

**PROPOSED MODELS:**

**Naive Bayes**

The straightforward probabilistic approach known as Naive Bayes is frequently employed for classification jobs. Naive Bayes can be used to categorise various hand gestures according to their properties or qualities in the context of hand gesture recognition.

Using Bayes' theorem to determine the likelihood of each potential class given the observed data is the fundamental concept behind Naive Bayes. The algorithm makes the assumption that each characteristic is independent, therefore the presence or absence of one feature has no bearing on the likelihood of any other feature.We would first need to extract characteristics from the input data, such as the location and orientation of the hand, the form of the fingers, or the movement of the hand over time, in order to employ Naive Bayes for hand gesture identification. The Naive Bayes classifier could then be trained using these features to discover the relationships between various feature values and various hand motions.

The likelihood of each potential gesture given the observed features would be calculated by the Naive Bayes classifier during classification, and the gesture with the highest probability would be chosen as the predicted gesture. One advantage of Naive Bayes is that it is computationally efficient and can be trained quickly on large datasets. However, its assumption of feature independence may not hold true in some cases, which can lead to inaccurate predictions. Additionally, Naive Bayes may not be able to capture complex relationships between features and gestures, which can limit its performance in more challenging recognition tasks.

**Stochastic Gradient Descent**

A well-liked optimisation approach in machine learning for training models, including those used in hand gesture detection, is stochastic gradient descent (SGD). To minimise a loss function based on the discrepancy between the projected output and the actual output, SGD iteratively changes the model's parameters.

SGD can be used to improve the weights and biases of a neural network or other classification model in the context of hand gesture recognition. The discrepancy between the anticipated and actual class labels for a specific set of input features can serve as the basis for the loss function employed in SGD.

SGD estimates the gradient of the loss function during training by randomly selecting a subset of the training data (sometimes referred to as a mini-batch). The model parameters are then updated using the gradient, shifting them in the direction that minimises the loss.

SGD has the benefit of handling huge datasets and high-dimensional feature spaces effectively, which makes it a good choice for jobs requiring hand motion detection. In addition, SGD is comparatively easy to implement and is easily parallelizable for computing speed.

**K Nearest Neighbors (KNN)**

A straightforward and efficient approach for classification problems, such as hand gesture detection, is K Nearest Neighbors (KNN). KNN operates by locating the input sample's k-nearest neighbours in the training dataset and assigning the majority of those neighbours' class labels to the input sample.

KNN can be used to categorise fresh hand gesture inputs in the context of hand gesture recognition based on how similar their feature vectors are to the feature vectors of known hand gestures in the training dataset. The feature vectors may comprise characteristics like the hand's location and orientation, the design of its fingers, or its temporal movement.

KNN must first be trained on a labelled dataset of hand gestures in order to be used for hand gesture recognition. Each gesture in the training set would have its feature vectors and matching class labels stored during training.

KNN would select the k-nearest neighbours based on some distance metric during classification by comparing the feature vector of the input gesture to the feature vectors of the training gestures (e.g. Euclidean distance). The input gesture would then be given the class label of the majority of those neighbours.

KNN's simplicity and interpretability—the algorithm is simple to comprehend and may be used for a variety of classification tasks—are two benefits. KNN can also handle non-linearly separable data and be excellent at capturing complex decision boundaries.

**Decision Tree**

A well-liked technique for classification problems, such as hand gesture recognition, is decision trees. Each leaf node of the tree is given a class name, and decision trees function by recursively partitioning the feature space based on the values of specific features.

Decision trees can be used to categorise new hand gestures in the context of hand gesture identification based on their feature vector. The hand's location and orientation, the design of its fingers, and its temporal movement can all be included in the feature vector. The algorithm would first need to be trained on a labelled dataset of hand gestures in order to employ decision trees for hand gesture recognition. In order to maximise the purity of the resulting subsets, the decision tree would recursively divide the feature space based on the values of individual features during training.

The decision tree would move through the tree during classification based on the values of the input characteristics, finally reaching a leaf node with a class label attached to it. The input gesture would subsequently be given this class label.

Decision trees have the benefit of being straightforward and understandable, making it simple to picture and comprehend the resulting tree. Decision trees can also manage data that cannot be separated linearly and can record interactions.

**Random forests**

An extension of decision trees, known as random forests, is a well-liked technique for classification problems like hand gesture detection. Several decision trees are built using random subsets of the training data and feature space in random forests, and their predictions are then combined using a voting system.

Random forests can be used to categorise brand-new hand motions according to their feature vector in the context of hand gesture identification. The hand's location and orientation, the design of its fingers, and its temporal movement can all be included in the feature vector.

The algorithm would first need to be trained on a labelled dataset of hand gestures in order to employ random forests for hand gesture recognition. In order to increase accuracy and minimise overfitting, several decision trees would be constructed during training on random subsets of the training data and feature space.

The input gesture feature vector would be sent through each decision tree by the random forest during classification to produce a set of class predictions. The final class designation would subsequently be decided by a voting process, such a majority vote.

Random forests have the advantage of handling high-dimensional feature spaces, noisy data, and the capacity to record feature interactions. When random subsampling is used, random forests are less likely to overfit than individual decision trees.

**Logistic Regression**

For classification applications, such as hand gesture recognition, a common approach is logistic regression. Based on a feature vector, logistic regression models the likelihood that a hand gesture belongs to a particular class.

Logistic regression can be used to categorise brand-new hand gestures according to their feature vector in the context of hand gesture identification. The hand's location and orientation, the design of its fingers, and its temporal movement can all be included in the feature vector.

The algorithm would need to be trained on a labelled dataset of hand gestures before it could utilise logistic regression to recognise hand gestures. Via the optimisation of a loss function like cross-entropy, the logistic regression model would discover during training the relationship between the input characteristics and the likelihood that a gesture belongs to a particular class.

The logistic regression model would use the input gesture's feature vector to classify the gesture, calculating the likelihood that each class it belongs to. The class with the highest probability would then be chosen as the final class label.

Logistic regression has the benefit of being straightforward and interpretable, making it simple to see and comprehend the final model. Also, by utilising non-linear modifications of the input features, logistic regression is able to handle data that cannot be separated linearly.

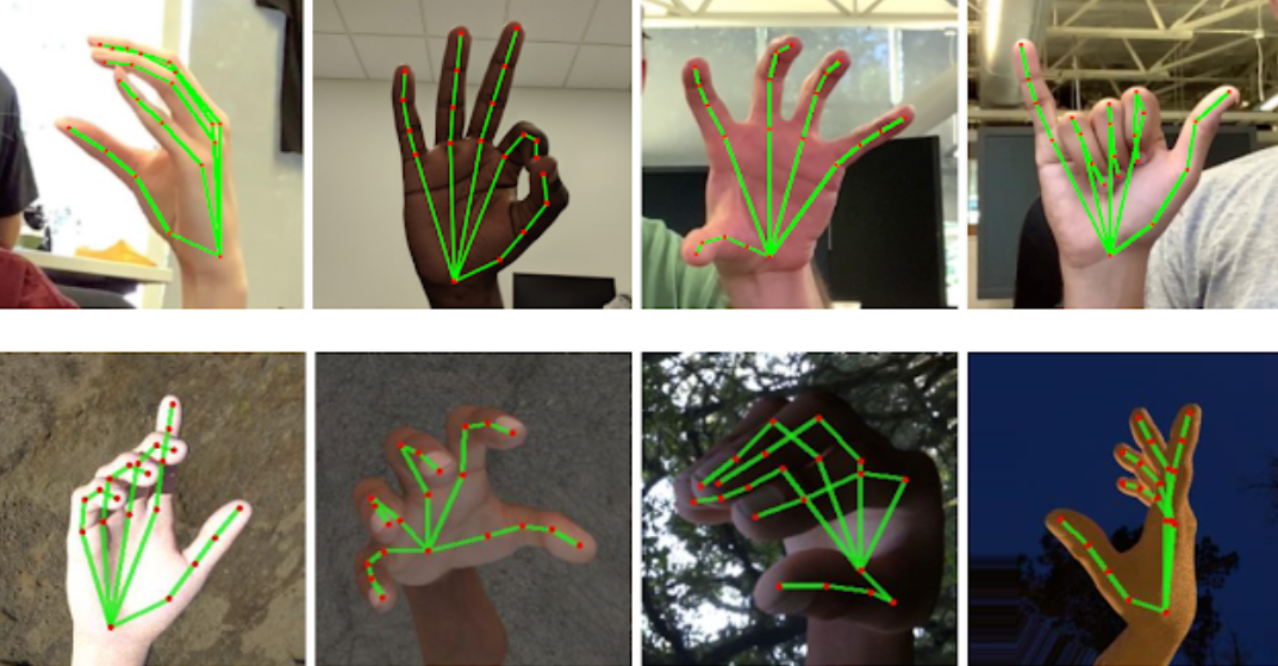
**Support Vector Machine**

A well-liked technique for classification tasks, such as hand gesture recognition, is Support Vector Machines (SVMs). Finding a hyperplane that divides the data into distinct classes and maximising the distance between the hyperplane and the nearest data points is how SVMs operate.

SVMs can be used to categorise new hand gestures according to their feature vector in the context of hand gesture recognition. The hand's location and orientation, the design of its fingers, and its temporal movement can all be included in the feature vector. The algorithm would first need to be trained on a labelled dataset of hand gestures in order to employ SVMs for hand gesture recognition. By maximising the distance between the hyperplane and the nearest data points, the SVM would discover the hyperplane that best categorises the data during training. The SVM can handle non-linearly separable data throughout the optimisation phase thanks to the usage of several kernels.

The SVM would analyse the input gesture's feature vector during classification to identify which side of the hyperplane it lies on. Assigning the input gesture to the class corresponding to the side of the hyperplane it lands on would then yield the final class label. Through the use of different kernels, such as the radial basis function (RBF) kernel or polynomial kernel, SVMs have the advantage of being able to handle high-dimensional feature spaces and non-linearly separable data. SVMs are also less prone to overfitting than other algorithms like neural networks or decision trees.

**MEDIAPIPE**



MediaPipe is an open-source, cross-platform framework developed by Google for building real-time computer vision and machine learning applications. One of the functionalities of MediaPipe is hand tracking and gesture recognition, which can be used for a variety of applications such as sign language recognition, augmented reality, and virtual reality.

MediaPipe's hand tracking and gesture recognition pipeline involves several steps, including hand landmark detection, hand region cropping, and gesture recognition. The hand landmark detection step involves using a neural network to detect and track the landmarks of the hand in real-time. The detected landmarks can then be used to crop the region of interest around the hand and extract relevant features for gesture recognition.

Real-time speed, which enables seamless integration into applications that need for quick and precise hand tracking and gesture detection, is one feature of MediaPipe's hand gesture recognition pipeline. A user-friendly API and a selection of pre- and post-processing tools are also provided by MediaPipe to make it easier to create and implement hand gesture recognition apps.

# SOURCE CODE:

# <https://drive.google.com/file/d/1QiRsnL3nLxstKHNHxn6OcGt7_HKPDUgf/view?usp=share_link>

*# %%*

*# Importing Libraries import* cv2

*import* mediapipe *as* mp *import* pandas *as* pd *import* os

*import* warnings

*from* sklearn *import* preprocessing

*from* sklearn.model\_selection *import* train\_test\_split *from* sklearn.metrics *import* classification\_report *from* sklearn.svm *import* SVC

*from* sklearn.naive\_bayes *import* GaussianNB

*from* sklearn.linear\_model *import* SGDClassifier *from* sklearn.neighbors *import* KNeighborsClassifier *from* sklearn.tree *import* DecisionTreeClassifier

*from* sklearn.linear\_model *import* LogisticRegression *from* sklearn.ensemble *import* RandomForestClassifier mp\_drawing = mp.solutions.drawing\_utils mp\_drawing\_styles = mp.solutions.drawing\_styles mp\_hands = mp.solutions.hands warnings.filterwarnings("ignore")

*# %%*

*# Feature Extraction* tab\_features = [] tab\_label = []

directory = 'Dataset\Dataset1\_Train'

*for* folder *in* os.listdir(directory):

f = os.path.join(directory, folder)

*for* file *in* os.listdir(f):

sample\_img = cv2.imread(os.path.join(f, file))

*with* mp\_hands.Hands(static\_image\_mode=True, max\_num\_hands=2, min\_detection\_confidence=0.5) *as* hands:

results = hands.process(cv2.cvtColor( sample\_img, cv2.COLOR\_BGR2RGB))

image = cv2.imread(os.path.join(f, file)) image\_height, image\_width, \_ = image.shape *if* results.multi\_hand\_landmarks:

x, y, z, label = [], [], [], []

*for* hand\_no, hand\_landmarks *in*

enumerate(results.multi\_hand\_landmarks):

*for* i *in* range(21): x.append(hand\_landmarks.landmark[mp\_hands.HandLandmark(

i).value].x \* image\_width - hand\_landmarks.landmark[mp\_hands.HandLandmark(0).value].x \* image\_width)

y.append(hand\_landmarks.landmark[mp\_hands.HandLandmark( i).value].y \* image\_height -

hand\_landmarks.landmark[mp\_hands.HandLandmark(0).value].y \* image\_height)

z.append(hand\_landmarks.landmark[mp\_hands.HandLandmark( i).value].z \* image\_width -

hand\_landmarks.landmark[mp\_hands.HandLandmark(0).value].z \* image\_width) col = x + y + z

tab\_features.append(col) tab\_label.append(folder)

*# %%*

*# Dataset Making*

df\_features = pd.DataFrame(tab\_features).iloc[:, :63] df\_label = pd.DataFrame(tab\_label)

*# %%*

*# Printing the Dataset*

print(df\_features)

*# %%*

*# Printing the Labels*

print(df\_label)

*# %%*

*# Label Encoding*

label\_encoder = preprocessing.LabelEncoder() df\_label = label\_encoder.fit\_transform(df\_label)

*# %%*

*# Printing the Labels after Label Encoding*

print(df\_label)

*# %%*

*# Train Test Split* target = df\_label features = df\_features

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split( features, target, test\_size=0.2, random\_state=10)

*# %%*

*# Support Vector Machine* svc = SVC() svc.fit(X\_train, Y\_train)

svc\_pred = svc.predict(X\_test) print(classification\_report(svc\_pred, Y\_test))

*# %%*

*# Naive Bayes*

nb = GaussianNB() nb.fit(X\_train, Y\_train) nb\_pred = nb.predict(X\_test)

print(classification\_report(nb\_pred, Y\_test))

*# %%*

*# Stochastic Gradient Descent* sgd = SGDClassifier() sgd.fit(X\_train, Y\_train) sgd\_pred = sgd.predict(X\_test)

print(classification\_report(sgd\_pred, Y\_test))

*# %%*

*# K Nearest Neighbour*

knn = KNeighborsClassifier() knn.fit(X\_train, Y\_train) knn\_pred = knn.predict(X\_test)

print(classification\_report(knn\_pred, Y\_test))

*# %%*

*# Decision Tree*

dt = DecisionTreeClassifier() dt.fit(X\_train, Y\_train) dt\_pred = dt.predict(X\_test)

print(classification\_report(dt\_pred, Y\_test))

*# %%*

*# Random Forest*

rf = RandomForestClassifier() rf.fit(X\_train, Y\_train) rf\_pred = rf.predict(X\_test)

print(classification\_report(rf\_pred, Y\_test))

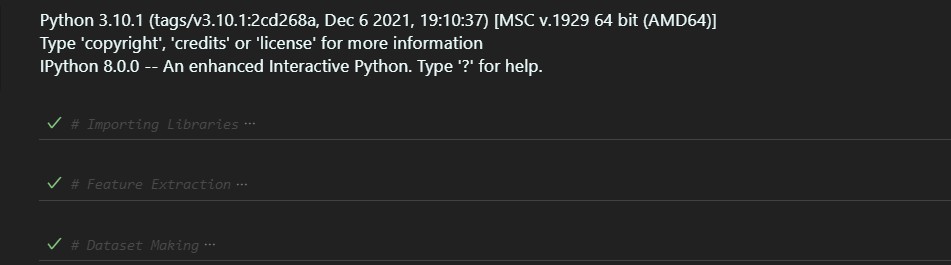
*# %%*

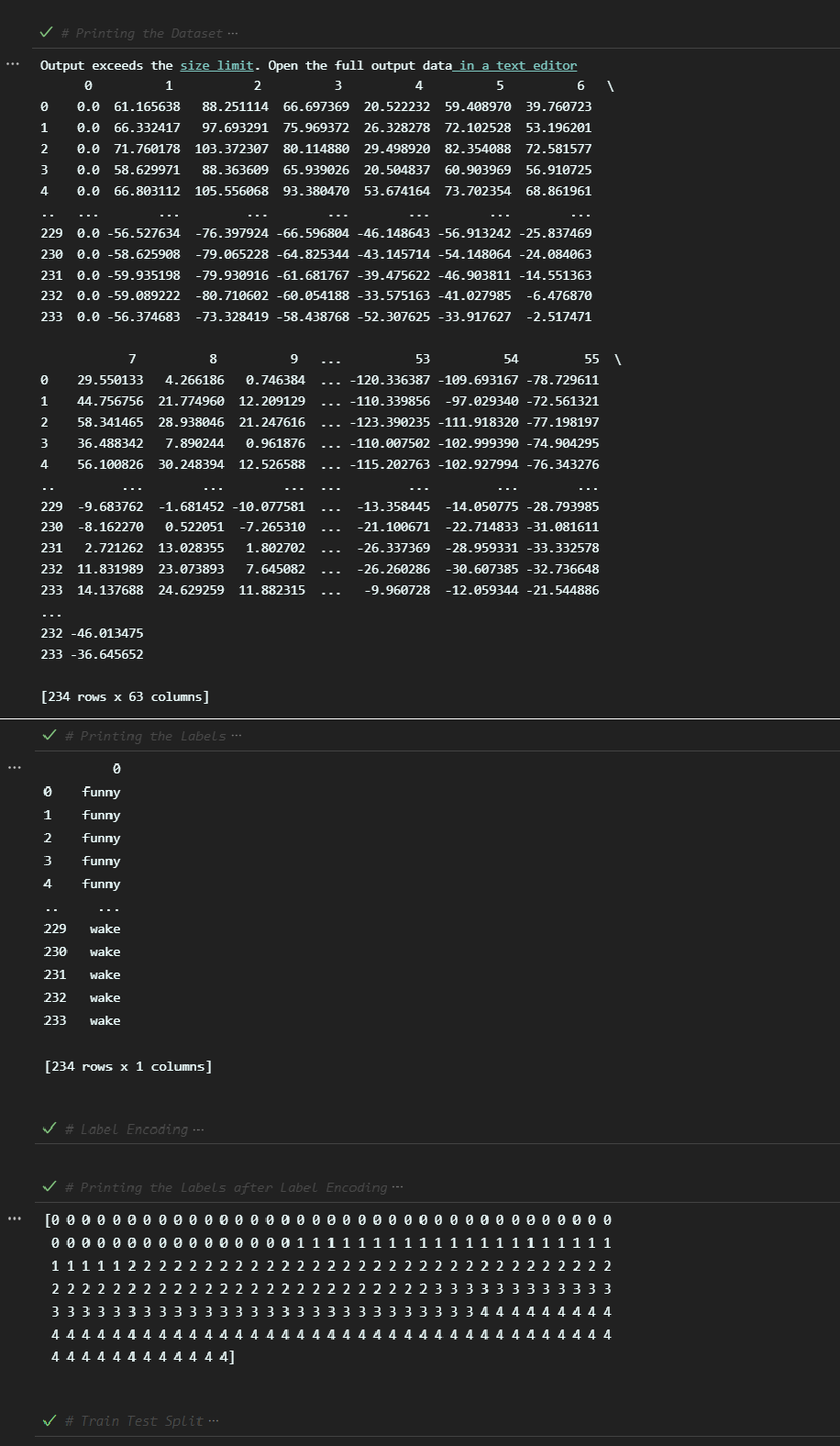
*# Logistic Regression*

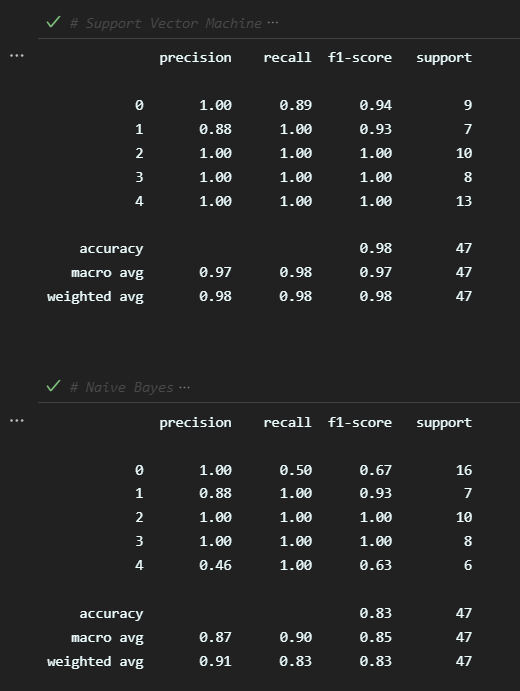
lr = LogisticRegression() lr.fit(X\_train, Y\_train) lr\_pred = lr.predict(X\_test)

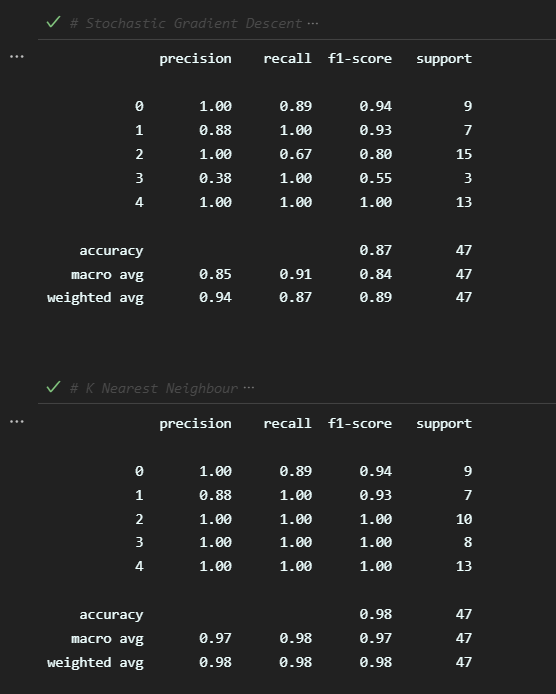
print(classification\_report(lr\_pred, Y\_test))

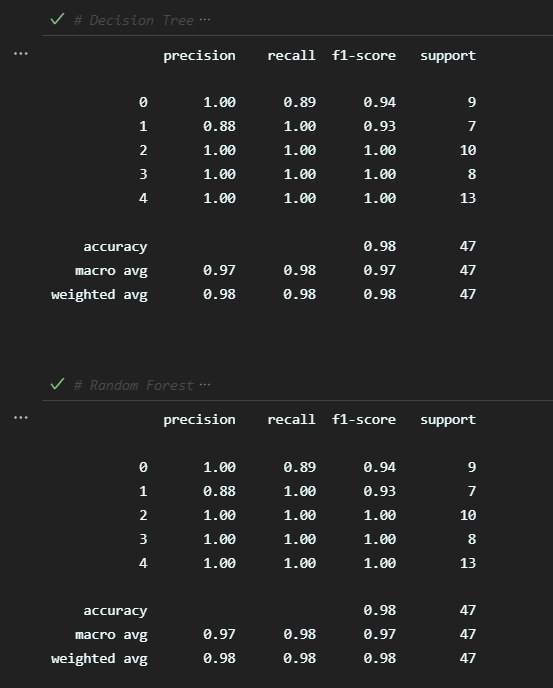
# OUTPUT Screenshots (Interactive ipynb Terminal):

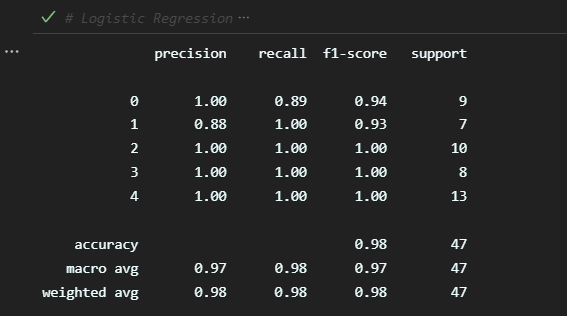












**Inferences:**

By comparing the f1 scores of all the seven classification models (Support Vector Machine, Naïve Bayes Classifier, Stochastic Gradient Descent, K Nearest Neighbour, Decision Tree, Random Forest and Logistic Regression), we find out that the models Support Vector Machine, K Nearest Neighbour, Decision Tree, Random Forest and Logistic Regression yield the best accuracy (~98%). Therefore, we can select any one of these classification models to build a successful model.

# Conclusion:

By looking at the accuracy of the model with a relatively smaller dataset, we can confirm that our model (which uses Mediapipe to extract hand co-ordinates) is definitely more efficient than the existing CNN-based Image Processing models (which used the pixels of the image to predict data) and that, the model will further improve on addition of more datas to the dataset.

# :

References

1. Gamal Tharwat , Abdelmoty M. Ahmed , and Belgacem Bouallegue, Procedia Computer Science, Volume 171, 2021, Pages 1-. (n.d.). *Arabic Sign Language Recognition System for Alphabets Using Machine Learning Techniques*.
2. Adithya V., Rajesh R, Volume 171, 2020, Pages 2353-2361, ISSN: 1877-. (n.d.). *A Deep Convolutional Neural Network Approach for Static Hand Gesture Recognition*.
3. Ali.H.Alrubayib, M.A.Ahmed, A.A.Zaidana, A.S.Albahri, B.B.Zaidan, O.S.Albahri, A.H.Alamoodi, Mamoun Alazabe, Year: 2021, Pages: 1 – 15, Article ID: (n.d.). *A Pattern Recognition Model for Static Gestures in Malaysian Sign Language Based on Machine Learning Techniques*.
4. Garcia, Brandon, and Sigberto Alarcon Viesca, Convolutional Neural Networks for Visual Recognition 2, 2016, Pages: 225-. (n.d.). *Real-time American Sign Language Recognition with Convolutional Neural Networks,*.
5. Joy, Helen & Kounte, Manjunath R & Joy, Ajin, 2020, Pages 134-138, Electronic ISBN: 978-1-7281-7213-2. (n.d.). *Deep Learning Approach in Intra -Prediction of High Efficiency Video Coding*.
6. Pigou, Lionel, Sander Dieleman, Pieter-Jan Kindermans, and Benjamin Schrauwen, European Conference on Computer Vision, 2014, Pages: 572-. (n.d.). *Sign Language Recognition using Convolutional Neural Networks*.
7. Samiya Kabir Youme, Towsif Alam Chowdhury, Hossain Ahamed, Md. Sayeed Abid, Labib Chowdhury, and Nabeel Mohammed, 2021, Pages 1-. (n.d.). *Generalization of Bangla Sign Language Recognition using Angular Loss Functions*.
8. Satwik Ram Kodandaram, N. Pavan Kumar, Sunil Gl. (n.d.). *Sign Language Recognition using CNN*.
9. Tolentino, Lean Karlo S. , Ronnie O. Serfa Juan, August C. Thio-ac, Maria Abigail B. Pamahoy, Joni Rose R. Forteza, and Xavier Jet O. Garcia, International Journal of Machine Learning and Computing 9, no. 6, 2019, Pages: 821-. (n.d.). *Static Sign Language Recognition using Deep Learning*.
10. V. John, A. Boyali, S. Mita, M. Imanishi and N. Sanma, 2016, Pages 1-8, Electronic ISBN: 978-1-5090-2896-. (n.d.). *Deep Learning-Based Fast Hand Gesture Recognition Using Representative Frames*.