Journal of Information Systems Engineering and Management
2025, 10(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Real-Time Recognition of American Sign Language Using Mediapipe and Machine Learning Techniques

Viswanath Sarma Ch[1*], Mrs. Hima Bindu Gogineni[2], Dr. B Prasad[3], Dr. P Praveen Kumar[4]
[1]Sr. Asst Prof, CSE- AI&ML, GMRIT, Rajam, Email ID: viswanathasarma.ch@gmrit.edu.in*
[2]IT, Anil Neerukonda Institute of Technology and Sciences (ANITS), Email ID: goginenibindu9@gmail.com
[3]Professor in Information Technology, Vignan's Institute of Information Technology(A), Visakhapatnam Email ID: arjunprasad.bode@gmail.com
[4]Department of AI & DS, Koneru Lakshmaiah Education Foundation (Deemed to be University), Guntur, India, Email ID: pk.pinjala@gmail.com

Citation: *Viswanath Sarma Ch et al. (2025) Real-Time Recognition of American Sign Language Using Mediapipe and Machine Learning Techniques, Journal of Information Systems Engineering and Management, 10(3), 01-15*
Doi: xyz

ARTICLE INFO

Received: 10 Nov 2024
Revised: 25 Dec 2024
Accepted: 22 Jan 2025

ABSTRACT

Hand sign recognition has become an important area of study in computer vision and pattern recognition, with applications including sign language translation and human-computer interaction developing humanoid robots in medical applications, corporations and restaurants in some countries. Users can interact with the devices without actually touching them through hand sign recognition. This paper offers a comparative analysis of Machine Learning algorithms for American hand-sign recognition with a bespoke coordinate's dataset including 87,000 photos, each measuring 200x200 pixels. The dataset has 26 classes, each representing a letter of the English alphabet, with 3000 images per class. In the pre-processing stage, MediaPipe, which is based on Convolutional Neural Netwok (CNN), extracts 21 points that represent the angles and directions of the hand for each image. Five models—Logistic Regression, K-Nearest Neighbors, Support Vector Machine, Decision Trees, and Random Forest—are evaluated for how well they can recognize the corresponding letter of each hand gesture. The results provide information on how well each method works for identifying American hand signs. The study shows that the Random Forest Algorithm works better than the others, achieving an accuracy of 98.83%. These findings are significant for developing automated systems that can use American Sign Language (ASL) for communication, such as sign language interpreters, smart home devices, and interactions with robots.

Keywords: American Sign Language, Hand-Sign Recognition, Machine Language Algorithms, Medi- aPipe, Comparative Study

1. INTRODUCTION

In recent times, advances in the domain of computer vision and pattern recognition have come faster, with gesture recognition as a central area for research. A domain ready to be put for practical uses is human-computer interaction enhancement and interpreting sign language. Sign language is a sophisticated method of communication, highly developed and flamboyant, to be used by mute individuals to express their thoughts and ideas to their fellow beings. In order to foster an inclusive society, which takes care of all members, irrespective of their physical conditions, the interpretation of sign language is a necessity.

American Sign Language (ASL) is a visual language broadly used by Deaf and hard-of-hearing individuals in the United States. ASL is the whole linguistic system with its syntax, grammar, and an in-joke vocabulary (1). Despite its popularity, ASL may sometimes be imperfectly understood. So, numerous technologies have been invented to allow communication among deaf and non-vocal persons (2). One of these technologies is hand sign recognition, where deep learning plus machine learning (ML) models have been mostly used to train the devices on how to react (3). Hand sign recognition with machine learning models has been a subject of multiple interests for a variety of reasons. The main aim of this research lies in developing a system which will be able to discern hand signs

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

accurately and translate them to text or speech. Much research has been carried out in this domain, but these models of machine learning have certain shortcomings, essential for them to be resolved.

Machine Learning has proven essential in numerous applications, including image identification and natural language processing. Hand sign recognition is a sophisticated endeavor within computer vision that entails identifying and analyzing human hand motions to derive meaning. The study results indicate that the Random Forest Algorithm surpasses other algorithms, providing significant insights into the efficacy of various machine learning algorithms for American hand-sign identification. This work emphasizes the potential of integrating Mediapipe with machine learning techniques to augment the precision of ASL identification systems. The area of hand sign recognition has developed significantly due to the emergence of novel deep learning (DL) techniques, in particular Convolutional Neural Networks (CNNs) (4). However, there remains the challenge that results obtained purely from Machine Learning Algorithms are not always accurate. Hence, the combination of these algorithms with other methods has proven to be effective. This paper presents the application of MediaPipe, a versatile open-source framework for building cross-platform Machine Learning pipelines, which includes convolutional neural networks for the processing of images. Google created it to ease the development of real-time cross-platform computer vision applications. It significantly simplifies the development of CNN-based Machine Learning pipelines for image processing. Due to MediaPipe's powerful tools, pre-built models, and hand sign recognition-oriented pipelines, developers can more easily create custom pipelines for a variety of tasks. With the help of MediaPipe's capabilities, a robust and accurate system for recognizing hand gestures in real-time can be developed.

This paper analyzes the potential of Mediapipe and machine learning techniques in real-time ASL recognition. The application of machine learning methods like Random Forest, together with the Mediapipe architecture, can substantially improve the accuracy of hand sign recognition in an impressive as for refining these models to boost their performance. The implications of this study are immense since it opens the doors for the creation of more inclusive and accessible societies which caters to all communities, regardless of their physical capabilities. This cutting edge technology has vast potential to be applied in different fields such as automated sign language interpretation, man-machine interaction, and even in the realms of gaming and virtual reality.

The results of the study claim that the random forest algorithm is among the other algorithms and provides important information about the effectiveness of the various algorithms for machine learning that are used to handmark identify the Americans. This work underlines the possibility of using Mediapipe together with some machine learning techniques in order to improve the accuracy of ASL identification systems. It contains suggestions on how to improve these models to increase your results. Research improves social welfare because it forms the basis for the creation of functional and reaction -fast societies that present people regardless of their physical abilities. Such a technology is expected to have far-reaching functions, including, but not limited to sign language interpretation, human computer interfaces and virtual reality.

The results of this research show that the random forest algorithm has exceeded all other analyzed algorithms in terms of accuracy. This is impressive, especially if it is CNN, which is known as one of the strongest algorithms for image recognition. Our approach, which combines a random forest with a CNN modified with Mediapipe, achieves remarkable results.

2. LITERATURE SURVEY

Recently, the recognition of the ASL using algorithms for ML and DL has been given more attention. The research interests are increased when evaluating how well these most modern techniques recognize and decode ASL characters. Significant research work on the use of ML and DL techniques in the ASL was carried out in order to identify best practice for increasing accuracy and efficiency in this area.

The aim of the current work is to create a comprehensive analysis of research on deep learning methods and mechanical learning for ASL. We strive to identify the most effective methods and techniques for the fastest and most precise ASL symbol detection by reviewing the latest research. We will take the different strategies into account in the literature, such as:

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

W.K. Wong et al. (5) proposed a capacity sensors on finger phalanges were used in a detection sensor device that was successfully implemented to recognize static ASL hand gestures. The potential data compression in the first 15 -feature data record was determined by data analysis. The test data record was successfully recognized by the classification models ECOC-SVM and KNN, which had recognition rates of 97 percent for the data between the participants and 99 percent for intra participant's data.

N. Gopinath et al. (6) presented a method for the classification and natural language processing (NLP) used for data regulations to identify hand gestures, which enables mutual communication between normal and impaired people.

Vedak Omkar et al. (7) presented a system that translates Indian Sign Language (ISL) into English, focusing on its use for deaf and mute individuals. The system uses webcam images to extract features, extract and classify the sign language, and converts it into speech using a text-to-speech API.

Nandy et al. (8) proposed a technique for identifying the Indian sign language (ISL), which is used by deaf and stupid people to communicate both static and dynamic hand gestures. Euclidical removal and K metrics for K-Nearstern neighbors are used for recognition in a video database with a large number of characters.

Chen, J.K. et al., (9) proposed an automatic sign language gesture recognition system using computer vision and machine learning tools. The basic skin segmentation model was found to be more effective than complex ones. The project also highlighted the time constraints and difficulties of creating a dataset from scratch. Although the system works well, there is still potential for future work.

Liao et al., (10) introduced a novel 3D Residual ConvNet and bi-directional LSTM network model for dynamic sign language recognition. The model effectively recognizes hand gestures by extracting spatiotemporal features and analyzing features sequence. Experimental results show accurate and effective distinction of sign languages, even for similar pairs of hand gestures.

Wazalwar et al. (11) emphasizes sign language interpretation and its importance in converting deaf/dumb sign language into sentences. It tests this on a limited database and suggests future work with larger databases and NLP algorithms for long sentences. The paper suggests incorporating facial expression detection for improved sign recognition efficiency. The study aims to bridge communication gaps between hearing disabled and normal people.

Although these studies show promising results for the identification of ASL -Alphabet signs, they also have restrictions. This study shows that the random forest model exceeds other algorithms with an accuracy rate of 98.83 %.

3. METHODOLOGY

3.1. Dataset Collection

The data record contained 26 folders that correspond to the letters of the ASL alphabet, together with 3 other folders for space, delete and nothing. Each folder consisted of more than 3,000 photographs of the hands of different people in different positions, backgrounds and lighting conditions. The underlying data record consists of a total of 87,000 photographs. This leads to 29 classes: 26 for A and 3 for the rest of the alphabet of 200x200 pixels.

3.2. Data Pre-processing and Dataset Preparation

The further development of algorithms for machine learning for hand signal detection is a diverse and dynamic domain. A crucial element of this process is the preliminary processing of data in which raw input data is converted into a format that is suitable for analysis and prediction by machine learning models. In this study, we describe a new data processing technology that the Mediapipe Library uses to provide real-time hand tracking and to generate 21 important point coordinates for each hand in one picture. The Mediapipe Hands model created by Google uses a deep architecture for neural networks that is tailored to the detection of the hand mark. This model was trained on an extensive data record of hand photos and videos, so that it can identify and monitor the location of the sights in real -time video. The model receives a grayscale image of the hand of the user and uses a sequence of folding layers

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

to extract relevant properties. The model creates a collection of trust worthy for every pioneering point, accompanied by the 2D coordinates of each point.

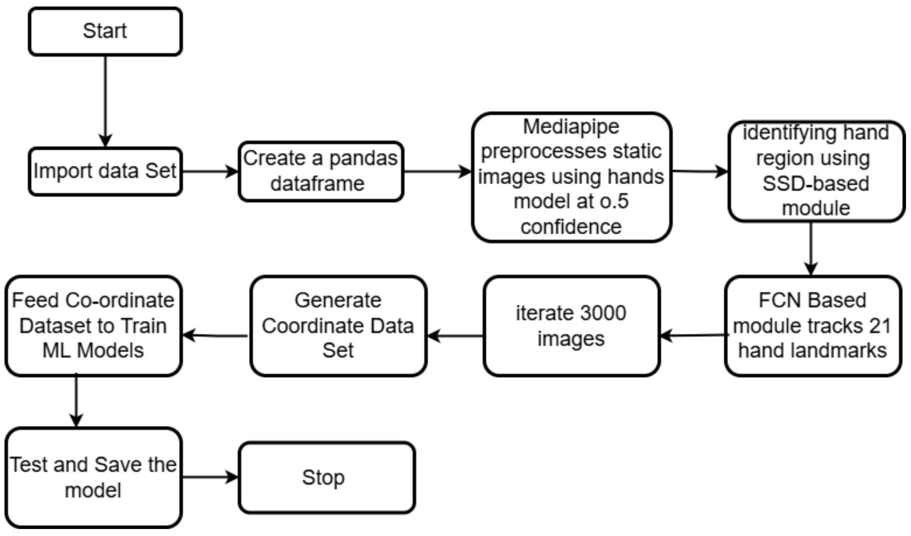

Start
Import data Set
Create a pandas dataframe
Mediapipe preprocesses static images using hands model at o.5 confidence
identifying hand region using SSD-based module
FCN Based module tracks 21 hand landmarks
iterate 3000 images
Generate Coordinate Data Set
Feed Co-ordinate Dataset to Train ML Models
Test and Save the model
Stop

Figure 1: Process Flow for Dataset and Model Creation

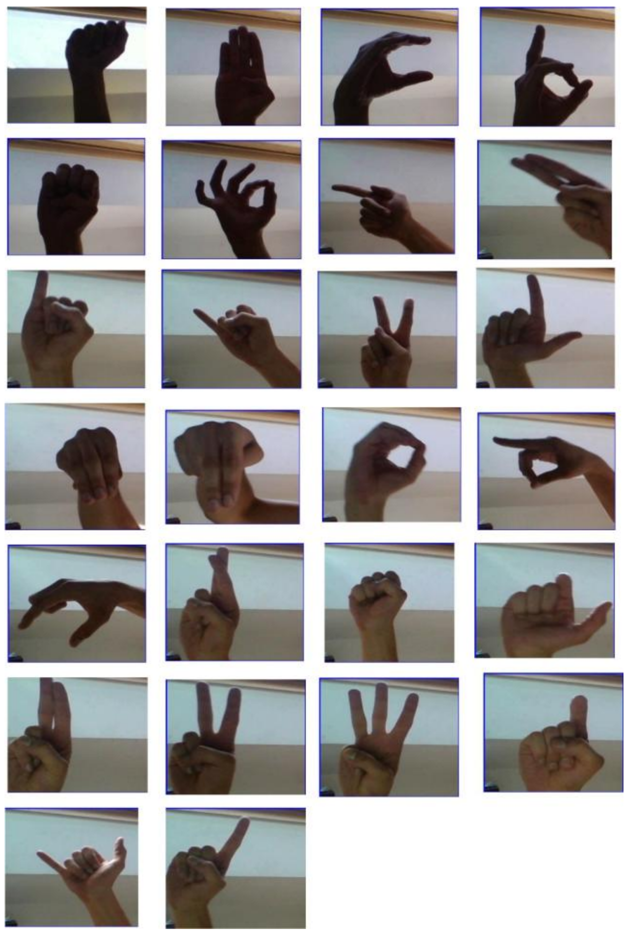


Figure 2: Sample Images from The American Hand Sign Dataset

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The model can also appreciate hand orientation and recognize hand movements such as fists and openings. To prepare the data record, we have created a Pandas data

frame that consists of 26 columns, one for each alphabet, and 3000 lines that consist of image paths for each image that corresponds to every alphabet.

We used the hand model of the Mediapipe Library to prepare the data by determining the parameters to process static images with a minimal confidence level of 0.5. The model initially identifies the area of the input picture in which there are probably hands, whereby a healing identification module is used based on the Single Shot Multibox Detector (SSD) architecture. As soon as the region of interest is identified, the hand markings with a manual marking module of the hand are recognized and followed, which is based on a variation of the Figoly Falharnet Network (FCN) architecture . The hand mark assessment module takes the hand region as input and generates a set of 21 3D noticeable coordinates that represent the position of different points on hand, e.g. B. the fingertips, the base of the thumb and the middle of the palm. These sights are followed over time so that the model can recognize hand movements and gestures.

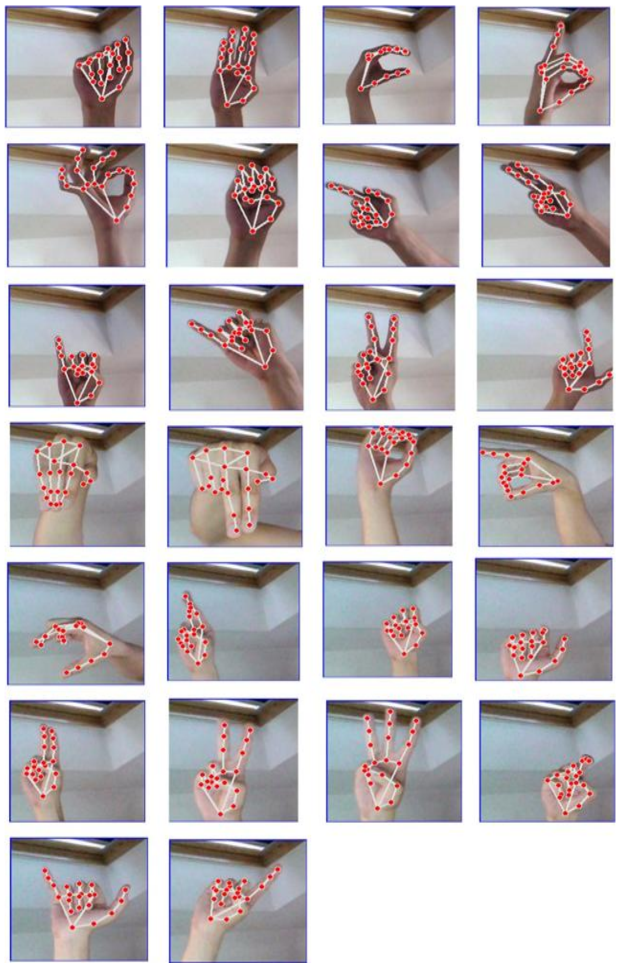


Figure 3: After Pre-processing and Land Marking

In order to shorten the computing time and complexity, we set a success variable in order to take the value to 1000 successfully pioneering images and to be terented across all 3000 images.

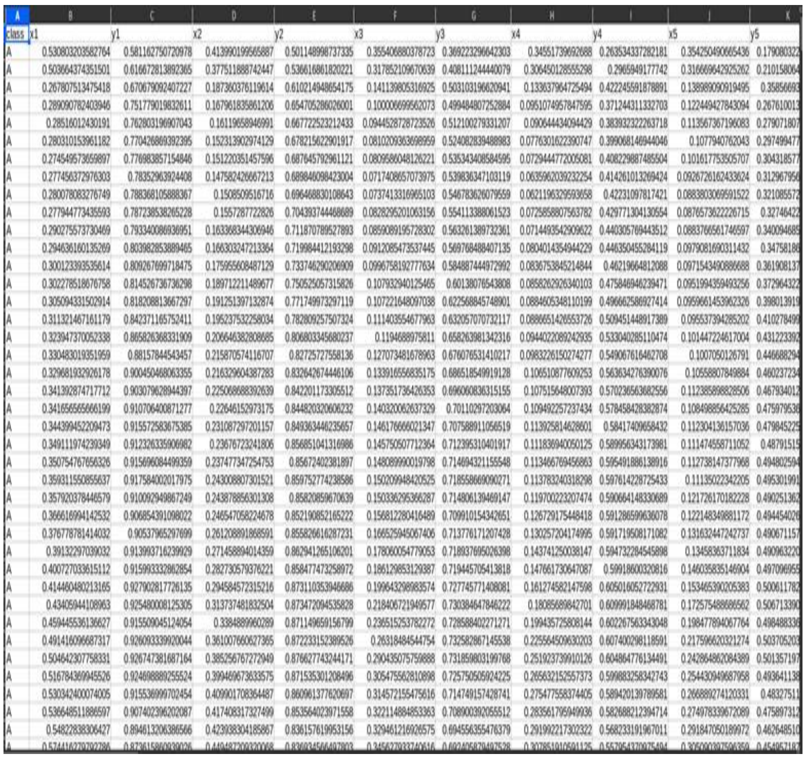
The resulting pandas data frequency comprised 26,000 lines with 1,000 correct, fine -marked photos for each letter of the alphabet and 43 columns. The 42 columns refer to the X and Y coordinates of 21 landmarks, while the last column means the associated alphabet class. Mediapipe facilitates the creation of numerical input data for hand signal identification and reduces computing time and complexity in image data processing. This method facilitates more

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

effective training of models for mechanical learning and improves predictability, which leads to a superior performance in hand signal marking applications.

3.3. Classification

Classification (12) is a technology for machine learning that divides incoming data into a number of certain groups according to a number of attributes or properties. Image and audio recognition, text classification and anomaly recognition are among the applications. Decision trees (DT), random forests (RF), K-Nearest neighbors (KNN) and support vector machines (SVM) are examples of classification methods.



A	B	C	D	E	F	G	H	I	J	K
class	x1	y1	x2	y2	x3	y3	x4	y4	x5	y5
A	0.530803203582764	0.581162750720978	0.413990199565887	0.501148998737335	0.355406880378723	0.369223296642303	0.34551739692688	0.263534337282181	0.354250490665436	0.179080322
A	0.503664374351501	0.616672813892365	0.377511888742447	0.536616861820221	0.317852109670639	0.408111244440079	0.306450128555298	0.2965949177742	0.316669642925262	0.210158064
A	0.267807513475418	0.670679092407227	0.187360376119614	0.610214948654175	0.141139805316925	0.503103196620941	0.133637964725494	0.422245591878891	0.138989090919495	0.35856693
A	0.289090782403946	0.751779019832611	0.167961835861206	0.654705286026001	0.100006699562073	0.499484807252884	0.0951074957847595	0.371244311332703	0.122449427843094	0.267610013
A	0.28516012430191	0.762803196907043	0.16119658946991	0.667722523212433	0.0944528728723526	0.512100279331207	0.090644434094429	0.383932322263718	0.113567367196083	0.279071807
A	0.280310153961182	0.770426869392395	0.152313902974129	0.678215622901917	0.0810209363698959	0.524082839488983	0.0776301622390747	0.399068146944046	0.1077940762043	0.297499477
A	0.274549673659897	0.776983857154846	0.151220351457596	0.687645792961121	0.0809586048126221	0.535343408584595	0.0729444772005081	0.408229887485504	0.101617753505707	0.304318577
A	0.277456372976303	0.78352963924408	0.147582426667213	0.689846098423004	0.0717408657073975	0.539836347103119	0.0635962039232254	0.414261013269424	0.0926726162433624	0.312967956
A	0.280078083276749	0.788368105888367	0.1508509516716	0.696468830108643	0.0737413316965103	0.546783626079559	0.0621196329593658	0.42231097817421	0.0883803069591522	0.321085572
A	0.277944773435593	0.787238538265228	0.1557287722826	0.704393744468689	0.0828295201063156	0.554113388061523	0.0725858807563782	0.429771304130554	0.0876573622226715	0.32746422
A	0.290275573730469	0.793340086936951	0.163368344306946	0.711870789527893	0.0859089195728302	0.563261389732361	0.0714493542909622	0.440305769443512	0.0883766561746597	0.340094685
A	0.294636160135269	0.803982853889465	0.166303247213364	0.719984412193298	0.0912085473537445	0.569768488407135	0.0804014354944229	0.446350455284119	0.0979081690311432	0.34758186
A	0.300123393535614	0.809267699718475	0.175955608487129	0.733746290206909	0.0996758192777634	0.584887444972992	0.0836753845214844	0.46219664812088	0.0971543490886688	0.361908137
A	0.302278518676758	0.814526736736298	0.189712211489677	0.750525057315826	0.107932940125465	0.60138076543808	0.0858262926340103	0.475846946239471	0.0951994359493256	0.372964322
A	0.305094331502914	0.818208813667297	0.191251397132874	0.771749973297119	0.107221648097038	0.622568845748901	0.0884605348110199	0.496662586927414	0.0959661453962326	0.398013919
A	0.311321467161179	0.842371165752411	0.195237532258034	0.782809257507324	0.111403554677963	0.632057070732117	0.0886651426553726	0.509451448917389	0.095537394285202	0.410278499
A	0.323947370052338	0.865826368331909	0.206646382808685	0.806803345680237	0.1194688975811	0.658263981342316	0.0944022089242935	0.533040285110474	0.101447224617004	0.431223392
A	0.330483019351959	0.88157844543457	0.215870574116707	0.82725727558136	0.127073481678963	0.676076531410217	0.0983226150274277	0.549067616462708	0.1007050126791	0.446688294
A	0.329681932926178	0.900450468063355	0.216329604387283	0.832642674446106	0.133916556835175	0.686518549919128	0.106510877609253	0.563634276390076	0.10558807849884	0.460237234
A	0.341392874717712	0.903079628944397	0.225068688392639	0.842201173305512	0.137351736426353	0.696060836315155	0.107515648007393	0.570236563682556	0.112385898828506	0.467934012
A	0.341656565666199	0.910706400871277	0.22646152973175	0.844820320606232	0.140320062637329	0.70110297203064	0.109492257237434	0.578458428382874	0.108498856425285	0.475979536
A	0.344399452209473	0.915572583675385	0.231087297201157	0.849363446235657	0.146176666021347	0.707588911056519	0.113925814628601	0.58417409658432	0.112304136157036	0.479845225
A	0.349111974239349	0.912326335906982	0.23676723241806	0.856851041316986	0.145750507712364	0.712395310401917	0.111836940050125	0.589956343173981	0.111474558711052	0.48791515
A	0.350754767656326	0.915696084499359	0.237477347254753	0.85672402381897	0.148089990019798	0.714694321155548	0.113466769456863	0.595491886138916	0.112738147377968	0.494802594
A	0.359311550855637	0.917584002017975	0.243008807301521	0.859752774238586	0.150209948420525	0.718558669090271	0.113783240318298	0.597614228725433	0.11135022342205	0.495301991
A	0.357920378446579	0.910092949867249	0.243878856301308	0.85820859670639	0.150336295366287	0.714806139469147	0.119700223207474	0.590664148330689	0.121726170182228	0.490251362
A	0.366616994142532	0.906854391098022	0.246547058224678	0.852190852165222	0.156812280416489	0.709910154342651	0.126729175448418	0.591286599636078	0.122148349881172	0.494454026
A	0.376778781414032	0.90537965297699	0.261208891868591	0.855826616287231	0.166525945067406	0.713776171207428	0.130257204174995	0.591719508171082	0.131632447242737	0.490671157
A	0.39132297039032	0.913993716239929	0.271458894014359	0.862941265106201	0.178060054779053	0.718937695026398	0.143741250038147	0.594732284545898	0.13458363711834	0.490963220
A	0.400727033615112	0.915993332862854	0.282730579376221	0.858477473258972	0.186129853129387	0.719445705413818	0.147661730647087	0.59918600320816	0.146035835146904	0.497096955
A	0.414460480213165	0.927902817726135	0.294584572315216	0.873110353946686	0.199643298983574	0.727745771408081	0.161274582147598	0.605016052722931	0.153465390205383	0.500611782
A	0.43405944108963	0.925480008125305	0.313737481832504	0.873472094535828	0.218406721949577	0.730384647846222	0.18085689842701	0.609991848468781	0.172575488686562	0.506713390
A	0.459445536136627	0.915509045124054	0.3384889960289	0.871149659156799	0.236515253782272	0.728588402271271	0.199435725808144	0.602267563343048	0.198477894067764	0.498488336
A	0.491416096687317	0.926093339920044	0.361007660627365	0.872233152389526	0.26318484544754	0.732582867145538	0.225564509630203	0.607400298118591	0.217596620321274	0.503705203
A	0.504642307758331	0.926747381687164	0.385256767272949	0.876627743244171	0.290435075759888	0.731859803199768	0.251923739910126	0.604864776134491	0.242864862084389	0.501357197
A	0.516784369945526	0.924698880255524	0.399469673633575	0.871535301208496	0.305475562810898	0.725750505924225	0.265632152557373	0.599883258342743	0.254430949687958	0.493641138
A	0.530342400074005	0.915536999702454	0.409901708364487	0.860961377620697	0.314572155475616	0.714749157428741	0.275477558374405	0.589420139789581	0.266889274120331	0.48327511
A	0.536648511886597	0.907402396202087	0.417408317327499	0.853564023971558	0.322114884853363	0.708900392055512	0.283561795949936	0.582688212394714	0.274978339672089	0.475897312
A	0.54822838306427	0.894613206386566	0.423938304185867	0.836157619953156	0.329461216926575	0.694556355476379	0.291992217302322	0.568233191967011	0.291847050189972	0.462648510
A	0.574416279792786	0.873615860939026	0.449487209320068	0.836934566497803	0.345627933740616	0.692405879497528	0.307851910501125	0.557054370975494	0.305000397506359	0.454957187

Figure 4: Generated Numerical Co-Ordinates Dataset With 26,000 Rows And 43 Columns

Training data quantity and quality, feature selection and algorithm selection all influence the accuracy and efficiency of the classification model. Classification models are currently being constructed using random forests, statistical regression, K-Nearest neighbor, support vector machines and decision-

making trees.

3.4. K-Nearest Neighbors

K-Nearest Neighbors (KNN) (13,18) is a non-parametric classification technology that

assigns the test data point with the highest natural neighboring presentation in the training of the class. Sample recognition and image processing applications often use KNN, a simple but effective technology, since the functions for the development and multi-class handling. However, the choice of the K value is of crucial importance for KNN and has a major impact on performance.

$$d\left(X_{new,}X_i\right) = \sqrt{\sum_{j=1}^{p}\left(X_{new,j} \ - X_{i,j}\right)^2} \qquad (1)$$

Where, p represents the number of features. Choose the K nearest neighbors to Xnew based on the measured distances. Assign the class label of the new data point Xnew as the majority class label among the K closest neighbors. This can be accomplished with a simple voting mechanism:

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

$$Y_{new} = \underset{c \in C}{\operatorname{argmax}} \sum_{i=1}^{K} w_i . I(Y_i = c) \qquad (2)$$

Where, C is the set of possible class labels, w_i is a weight assigned to each neighbor (which can be equal to 1 for a uniform weighting), I is an indicator function that outputs 1 when its argument is true and 0 otherwise.

3.5. Logistic Regression

Logistic Regression (LR) (14) is a binary classification algorithm that computes the likelihood that a given input belongs to a specific class. In the context of hand sign recognition, Logistic Regression can be taught to categorize photographs of hand signs as belonging to a particular alphabet or not.

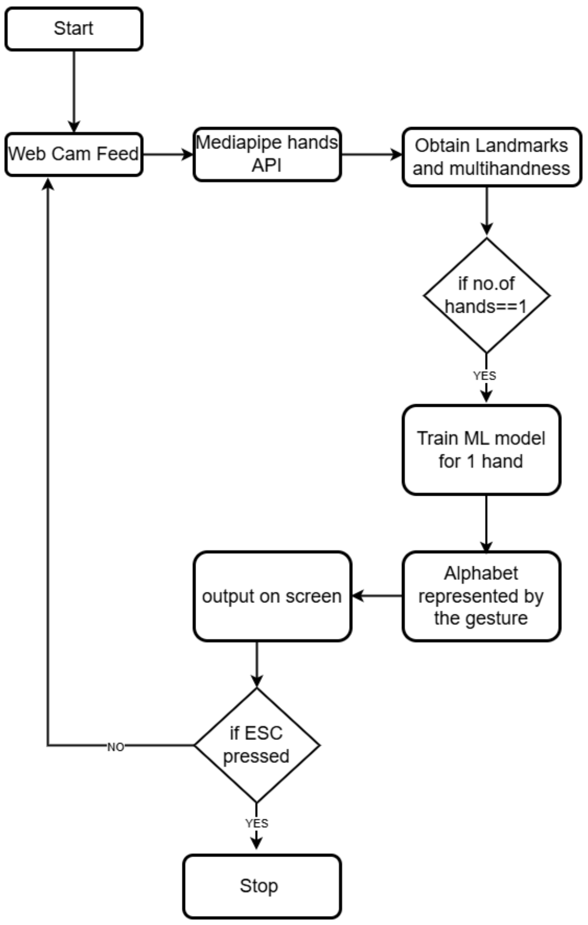


Figure 5: Process Flow Diagram

It operates by simulating the relationship between the input features (the hand sign image) and the binary target variable (the class label). Logistic regression is a simple and computationally efficient technique capable of achieving high accuracy for specific classification problems.

$$P(Y = 1|X;\beta) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + --- + \beta_p X_p)}} \qquad (3)$$

Where, β0 is the intercept or bias term, X1, X2,..., Xp are the features, and β1, β2,..., βp are the weights that are learned during training.

Support Vector Machine

The Support Vector Machine (SVM) (15) is one of the high-end classification algorithms to find out the best-tended hyper-level in the high-dimensional space. SVM transforms data points into a higher -dimensional characteristic space before a hyperplane with the maximum possible scope between the classes is identified. Since it uses the

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

kernel function to change the data for a new room, the technology works well on data that cannot be transferred linearly. Due to its precision and the ability to master large amounts of data, SVM is used extensively in various areas.

$$\langle w, x_i \rangle + b \geq 1, \quad \text{if } y_i = 1 \tag{4}$$

$$\langle w, x_i \rangle + b \leq -1, \quad \text{if } yi = -1 \tag{5}$$

$$\min_{\alpha} \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{n} \alpha_i \tag{6}$$

$$\text{Subject to } \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

Where, α_i are the Lagrange Multipliers, K (x_i , x_j) is the kernel function that measures.

Decision Tree

The classification of a decision tree (DT) (16) is a non -parametric technology in which data is divided into small categories until the conclusion. It is most frequently used in machine learning due to interpretability and simple decision -making. Algorithms can be used with both categorical and continuous input variables and therefore applied to almost any type of application. Overtime can occur if the tree is too complex and the performance suffers from large data sets.

Random Forest

Random Forest (RF) (17,18) is an ensemble algorithm for machine learning, which combines numerous decisions - making trees in order to increase the predictive accuracy and at the same time minimize the overhanging. Each tree is constructed using a random subset of the properties and samples of the training data. During the prediction, the algorithm uses the majority of all trees to make the final categorization selection. Random Forest is a popular and successful classification technology that is known that it treats high -dimensional data well and at the same time maintains good performance in the presence of loud properties.

RESULTS

In this experiment, we trained our classifiers in a data set that we had previously consisted of Mediapipe, which consisted of 26,000 lines and divided the data with a test size of 0.33, which led to 17,420 lines of train data and 8580 lines of test data. We applied 87,000 pre -processed pictures of 200x200 pixels. The training was carried out via an i5-11300H processor with an integrated graphics card and 16 gigabyte DDR4 RAM. The model was trained within a less than 300 seconds and exported to a local file by being serialized in a cucumber file. The classifier was tested with several criteria, including accuracy, precision and recall. The results of the evaluation give us the information on the performance of the classifier for the correct classification of characters. We also calculated the macro and the weighted average values for the metrics. The only difference between the two is that the macro average treats all classes equally, while the weighted average evaluates the relevance of each class. This enables us to determine the suitability of the classifier for the respective task and makes a well-founded decision on practical applicability. Overall, the results of this study offer useful insights into the performance of the classifier and can guide future research in this area.

Performance Evaluation with Random Forest algorithm

The Random Forest Classifier achieved an accuracy of 0.988 and an F1-score of 0.988. The result is given in the Table 1. These metrics suggest that the classifier performs very well in accurately classifying the hand gestures with a high level of precision and recall.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Journal of Information Systems Engineering and Management
2025, 10(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table1: Performance Evaluation Metrics Using Random Forest Algorithm

Random Forest			
Alphabets	**Precision**	**Recall**	**F1-score**
A	0.991	0.991	0.991
B	0.967	1.000	0.983
C	0.997	0.997	0.997
D	1.000	0.980	0.990
E	0.991	0.994	0.993
F	0.991	0.994	0.993
G	0.983	1.000	0.991
H	0.997	0.983	0.990
I	0.991	0.991	0.991
J	0.994	0.991	0.992
K	0.991	0.997	0.994
L	1.000	0.997	0.998
M	0.945	0.951	0.948
N	0.962	0.931	0.946
O	0.975	0.997	0.986
P	1.000	0.997	0.998
Q	0.988	0.997	0.993
R	0.971	0.965	0.968
S	0.982	0.979	0.980
T	0.994	0.991	0.992
U	0.944	0.944	0.944
V	0.981	0.975	0.978
W	1.000	0.988	0.994
X	0.985	0.994	0.989
Y	0.997	1.000	0.998
Z	1.000	0.997	0.999
macro avg	**0.99**	**0.99**	**0.99**
weighted avg	**0.99**	**0.99**	**0.99**
Accuracy		**0.9883449883449883**	
F1 Score		**0.988318103611549**	

Performance Evaluation with KNN algorithm

According to Table 2, the K-Nearest Neighbor (KNN) classifier had an F1-score of 0.978 and an accuracy of 0.979. These metrics indicate that the KNN classifier performs well in reliably classifying hand gestures, however the results are significantly lower than those achieved with the Random Forest Classifier due to the KNN algorithm's constraints.

Table2: Performance Evaluation Metrics Using Random Forest Algorithm

K-Nearest neighbors			
Alphabets	**Precision**	**Recall**	**F1-score**
A	0.997	0.979	0.988
B	0.964	1.000	0.982
C	0.991	0.994	0.993
D	0.997	0.977	0.987

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

E	0.994	0.976	0.985
F	0.991	0.982	0.986
G	0.980	1.000	0.990
H	1.000	0.986	0.993
I	0.997	0.985	0.991
J	0.991	0.997	0.994
K	0.968	0.997	0.982
L	1.000	0.994	0.997
M	0.926	0.926	0.926
N	0.941	0.921	0.931
O	0.960	0.997	0.978
P	0.993	0.987	0.990
Q	0.991	0.994	0.993
R	0.968	0.965	0.966
S	0.964	0.973	0.968
T	0.988	0.991	0.989
U	0.909	0.925	0.917
V	0.971	0.925	0.948
W	0.994	0.991	0.992
X	0.973	0.979	0.976
Y	0.997	1.000	0.998
Z	0.997	1.000	0.999
macro avg	**0.98**	**0.98**	**0.98**
weighted avg	**0.98**	**0.98**	**0.98**
Accuracy		**0.9786713286713287**	
F1 Score		**0.9784164207600803**	

Performance Evaluation with Logistic Regression.

Table 3 shows that the accuracy and F1-score obtained with Logistic Regression are 0.983 and 0.982 respectively.These results are marginally lower than those produced with the Random Forest Classifier, implying that the Random Forest Model may be a better fit for the data than the Logistic Regression model.

Table3: Performance Evaluation Metrics Using Logistic Regression Algorithm

Logistic Regression			
Alphabets	**Precision**	**Recall**	**F1-score**
A	0.985	0.997	0.991
B	0.970	1.000	0.985
C	0.997	0.997	0.997
D	1.000	0.989	0.994
E	0.985	0.988	0.987
F	0.997	0.997	0.997
G	0.994	0.997	0.996
H	0.997	0.978	0.987
I	0.988	0.976	0.982

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

J	0.991	0.988	0.989
K	0.994	1.000	0.997
L	0.997	0.997	0.997
M	0.912	0.929	0.921
N	0.943	0.900	0.921
O	0.975	0.997	0.986
P	1.000	0.993	0.997
Q	0.994	0.994	0.994
R	0.974	0.962	0.968
S	0.979	0.964	0.971
T	0.994	0.979	0.986
U	0.954	0.969	0.961
V	0.994	0.997	0.995
W	1.000	0.985	0.992
X	0.936	0.982	0.959
Y	0.997	1.000	0.998
Z	1.000	0.994	0.997
macro avg	**0.98**	**0.98**	**0.98**
weighted avg	**0.98**	**0.98**	**0.98**
Accuracy		**0.9826340326340326**	
F1 Score		**0.9824982539953742**	

Random Forest is capable of handling nonlinear interactions between features and the target variable. Logistic regression presupposes a linear relationship between the independent variables and the dependent variable. In comparison, Random Forest can capture complex non-linear interactions by mixing numerous decision trees.

Performance Evaluation with Decision Tree Algorithm

As indicated in Table 4, the Decision Tree had an accuracy of 0.922 and an F1-score of 0.921, which was lower than the results obtained with the Random Forest Classifier. The Decision Tree is a standalone technique that uses a single tree to classify data. The Decision Tree can overfit training data and perform badly on new data, particularly when the dataset is huge and complex. Furthermore, the Decision Tree does not allow you to assess the importance of each feature in the forecast, which can aid in determining the most relevant features for the task at hand. This implies that the Decision Tree might not be the most appropriate method for this particular application.

Table 4: Performance Evaluation Metrics Using Decision Tree Algorithm

Decision Tree			
Alphabets	**Precision**	**Recall**	**F1-score**
A	0.955	0.960	0.957
B	0.906	0.959	0.932
C	0.978	0.918	0.947
D	0.932	0.937	0.934
E	0.896	0.901	0.898
F	0.953	0.964	0.958
G	0.962	0.977	0.969
H	0.971	0.925	0.948
I	0.953	0.930	0.942

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

J	0.893	0.920	0.906
K	0.949	0.955	0.952
L	0.951	0.984	0.967
M	0.848	0.874	0.861
N	0.890	0.831	0.859
O	0.866	0.919	0.892
P	0.944	0.935	0.939
Q	0.959	0.950	0.954
R	0.867	0.853	0.860
S	0.887	0.879	0.883
T	0.953	0.936	0.944
U	0.780	0.817	0.798
V	0.892	0.894	0.893
W	0.979	0.973	0.976
X	0.919	0.897	0.908
Y	0.951	0.933	0.942
Z	0.944	0.949	0.947
macro avg	**0.92**	**0.92**	**0.92**
weighted avg	**0.92**	**0.92**	**0.92**
Accuracy		**0.92226107226l0722**	
F1 Score		**0.9218188132646936**	

Performance Evaluation with SVM algorithm

According to Table 5, the Support Vector Machine (SVM) classifier had an F1- score of 0.9764 and an accuracy of 0.9766.

Table 5: Performance Evaluation Metrics Using SVM Algorithm

Support Vector Machines			
Alphabets	**Precision**	**Recall**	**F1-score**
A	0.997	0.997	0.997
B	0.958	1.000	0.979
C	0.997	0.997	0.997
D	0.994	0.954	0.974
E	0.985	0.997	0.991
F	0.991	0.997	0.994
G	0.994	1.000	0.997
H	1.000	0.997	0.999
I	0.976	0.991	0.983
J	0.988	0.979	0.983
K	0.997	0.964	0.980
L	1.000	0.997	0.998
M	0.922	0.914	0.918
N	0.935	0.912	0.924
O	0.951	0.997	0.973
P	0.997	0.997	0.997
Q	0.991	1.000	0.996

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

R	0.955	0.939	0.947
S	0.985	0.967	0.976
T	0.997	0.991	0.994
U	0.821	0.966	0.887
V	1.000	0.866	0.928
W	1.000	0.988	0.994
X	0.988	0.988	0.988
Y	0.997	1.000	0.998
Z	1.000	0.994	0.997
macro avg	0.98	0.98	0.98
weighted avg	0.98	0.98	0.98
Accuracy		**0.9766899766899767**	
F1 Score		**0.9764892445571648**	

SVM implies a linear relationship and is susceptible to outliers and noise in data. However, it is vital to note that the classifier's performance will vary based on the dataset and training parameters. Additional testing and refinement may be required to improve SVM's performance on this dataset.

Comparative Study

Table 6 compares the performance of the five different classifiers. When compared to all other classifiers, the random forest classifier has the highest accuracy and F1 score. This suggests that random forests are a promising method for hand sign recognition.

Figure 6 compares the precision ratings of the five classifiers employed in our investigation. The graph clearly shows that Random Forest outperforms all other classifiers, with a precision score of 0.991. The second-best performance is logistic regression, with a precision score of 0.983, followed by SVM, which has a score of 0.948. These results are slightly lower than the performance of the Random Forest Classifier, which was used to compare.

Table 6: Comparative Study of Different Classifiers

Metrics	Random Forest	KNN	Logistic Regression	Decision Tree	SVM
macro avg	0.9995	0.9885	0.9885	0.9225	0.9885
weighted avg	0.9995	0.9885	0.9885	0.9225	0.9885
Accuracy	0.9883	0.9786	0.9826	0.9222	0.9767
F1_score	0.9883	0.9784	0.9825	0.9218	0.9765

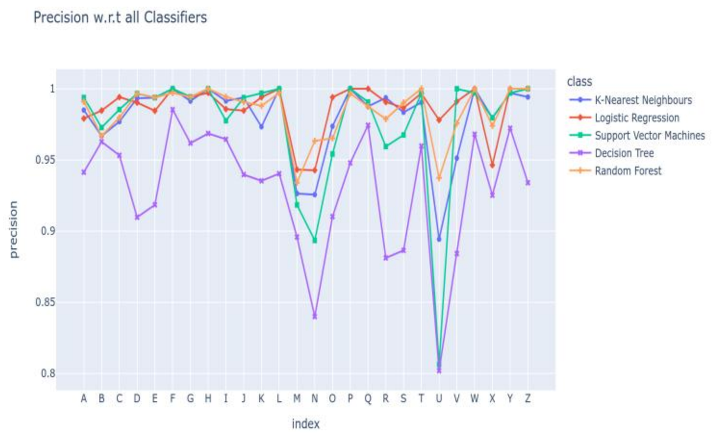


Figure 6: Precision Scores of All the Classifiers

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

KNN and Decision Tree exhibit the lowest precision scores, recorded at 0.923 and 0.903, respectively. Random Forest outperforms the other classifiers in terms of precision scores for the majority of alphabets. Furthermore, we see a considerable decrease in the precision of all classifiers for the alphabets m, n, and u. However, even for these difficult alphabets, Random Forest maintains a higher level

Figure 7 shows that the Random Forest Classifier continues to beat the other

classifiers in terms of recall scores. This suggests that Random Forest performs better at correctly identifying positive instances, or classifying hand signs.

classifiers in terms of recall scores. This suggests that Random Forest performs better at correctly identifying positive instances, or classifying hand signs. Similar to the precision scores, we can notice a large decline in recall scores for the alphabets m, n, and u for all classifiers, however Random Forest still has a good recall score for all alphabets. This shows that Random Forest is stronger and more dependable at classifying even the most difficult alphabets of precision than the other classifiers

The classification accuracy of an algorithm is well described by an F1 score, which combines the values of both accuracy and recirculation measures. Therefore, there is a balanced view of the way an algorithm works at the class level. Figure 8 shows that Randallswald defeats all other algorithms and results in an F1 score of 0.99. For decision -making tree, SVM, logistical regression, KNN were 0.88, 0.98 or 0.98. This implies that random forest is the most prominent of these five algorithms in the classification of ASL. In comparison of the F1 scores, the most suitable ASL algorithm must be random forest.

We used the Mediapipe Library, a CNN-based python library, to start the hand model, which was then loaded with our trained random forest classifier. Real-time film material was recorded with OpenCV and fed into the hand model for prediction. In order to evaluate its performance, the model was tested in various lighting situations and with various hand signals. Our method provided extremely precise results because the algorithm almost quickly suspected the algorithm in real -time videos with a very precise precision. Figure 8 shows some of the projected consequences of our model.

Our strategy for the combination of deep learning and machine learning with media pipe proved to be very effective in the field of image categorization, especially hand signaling. The use of media pipe, a powerful and versatile library, offers robust image processing functions, while the random forest approach provided higher classification accuracy than other algorithms for machine learning.

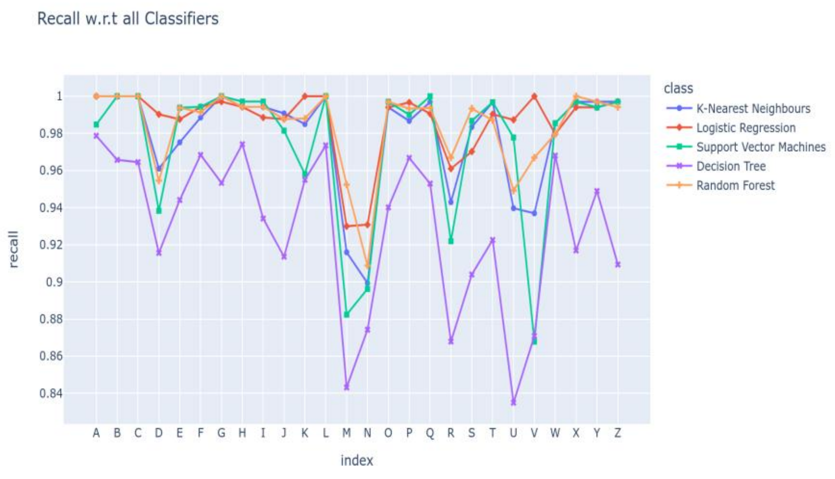


Figure 7: Recall Scores of all the Classifiers

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

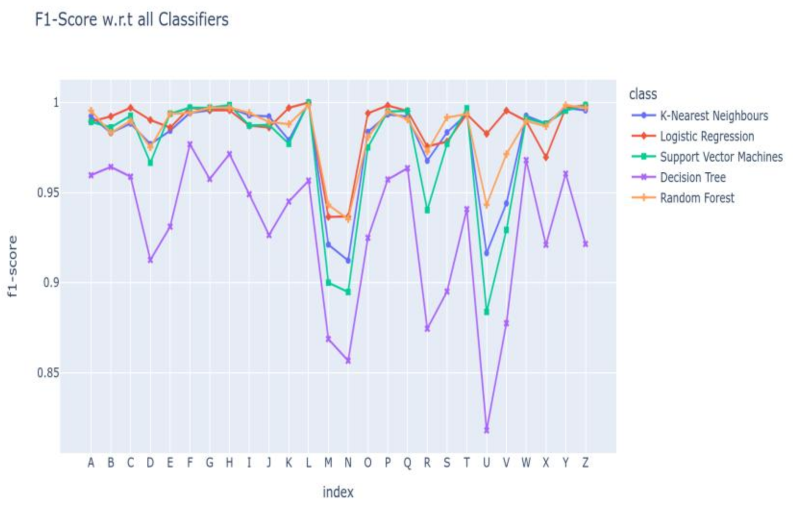


Figure 8: F1 Scores of all the Classifiers

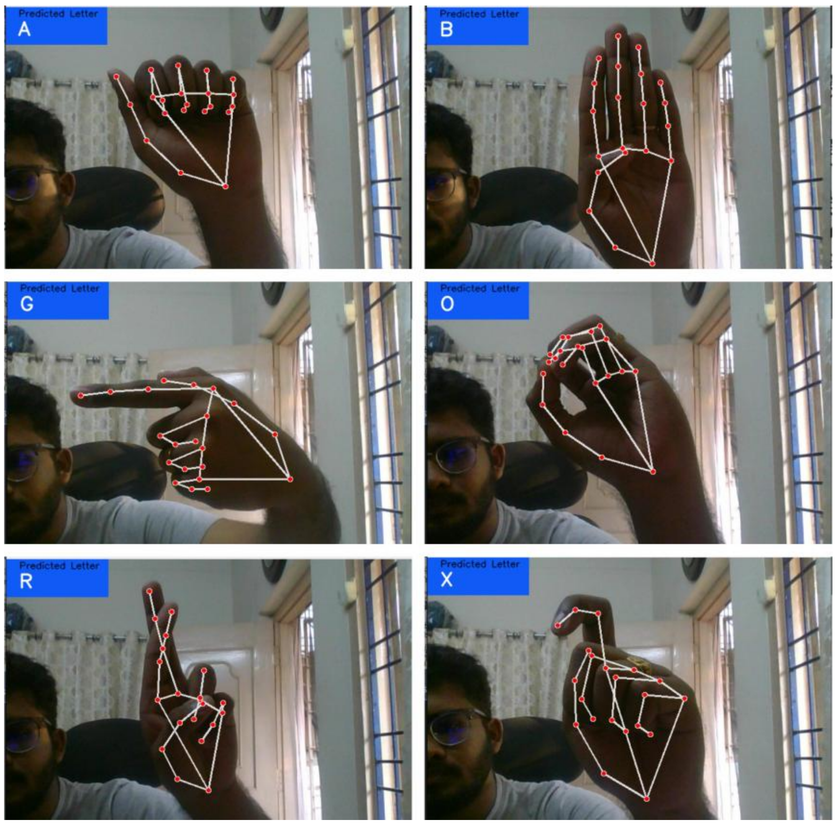


Figure 9: Random Forest Model Classifying Hand Signs and Translating in Real Time

To summarize, our study proposes a successful model for real-time hand sign recognition based on the MediaPipe library and the Random Forest algorithm. The model produced highly accurate and exact predictions, demonstrating our approach's applicability for picture classification. Our findings make a substantial contribution to the development of more efficient and robust models for real-time hand sign recognition, with potential applications in accessibility, education, and human-computer interaction.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

CONCLUSION

A study on the efficacy of Mediapipe and machine learning algorithms for real-time American Sign Language (ASL) recognition produced impressive results. The study compared five classification models, including Logistic Regression, K-Nearest Neighbors, Support Vector Machines, Decision Trees, and Random Forest, for detecting the alphabet of each hand gesture. The Random Forest Algorithm surpassed the other algorithms, with an astonishing 98.83% accuracy rate. This study gives useful insights for enhancing machine learning models to improve performance and accuracy in automated systems that communicate with American hand signs, such as sign language interpreters, smart home systems, and human-robot interaction. The findings highlight the possibility of merging Mediapipe with machine learning techniques for ASL detection, although restrictions such as the dataset's confinement to 26 English alphabet classes must be addressed. The study compared five classification models, including Logistic Regression, K-Nearest Neighbors, Support Vector Machines, Decision Trees, and Random Forest, for detecting the alphabet of each hand gesture. The Random Forest Algorithm surpassed the other algorithms, with an astonishing 98.83% accuracy rate.

Acknowledgement : NA

Funding Information: No funding

Conflict of Interest: The authors do not have any conflict of interest

Author's contribution (Not Compulsory)

REFERENCES

1. Mitchell RE, Young TA, Bachleda B, Karchmer MA. How many people use ASL in the United States? Why estimates need updating. Sign Language Studies. 2006;6(3):306-35.
2. N. Rajasekhar, M. G. Yadav, C. Vedantam, K. Pellakuru and C. Navapete, "Sign Language Recognition using Machine Learning Algorithm," 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 2023, pp. 303-306,doi: 10.1109/ICSCSS57650.2023.10169820.
3. I.A. Adeyanju, O.O. Bello, M.A. Adegboye,
4. Machine learning methods for sign language recognition: A critical review and analysis, Intelligent Systems with Applications, Volume 12, 2021, 200056, ISSN 2667-3053,https://doi.org/10.1016/j.iswa.2021.200056
5. K. Bantupalli and Y. Xie, "American Sign Language Recognition using Deep Learning and Computer Vision," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 4896-4899, doi: 10.1109/BigData.2018.8622141.
6. W. K. Wong, Filbert H. Juwono, "Multi-Features Capacitive Hand Gesture Recognition Sensor: A Machine Learning Approach", IEEE SENSORS JOURNAL, VOL. 21, NO. 6, MARCH 15, 2021.
7. N. Gopinath, J. Anuja, S. Anusha, V. Monisha , "A Survey on Hand Gesture Recog- nition Using Machine Learning", International Research Journal of Engineering and Technology (IRJET), 2020.
8. Omkar Vedak, Prasad Zavre, Abhijeet Todkar, Manoj Patil, "Sign Language Interpreter using Image Processing and Machine Learning", International Research Journal of En- gineering and Technology (IRJET), 2019.
9. Nandy, A.; Pr7asad, J.; Mondal, S.; Chakraborty, P.; Nandi, G. "Recognition of Isolated Indian Sign Language Gesture in Real Time". Commun. Comput. Inf. Sci. 2010, 70, 102–107.
10. Chen, J.K. "Sign Language Recognition with Unsupervised Feature Learning"; CS229 Project Final Report; Stanford University: Stanford, CA, USA, 2011.
11. Liao, Y.; Xiong, P.; Min, W.; Min, W.; Lu, J. "Dynamic Sign Language Recognition Based on Video Sequence with BLSTM-3D Residual Networks". IEEE Access 2019, 7, 38044–38054.
12. Wazalwar, S.S.; Shrawankar, U. "Interpretation of sign language into English using NLP techniques". J. Inf. Optim. Sci. 2017, 38, 895–910.
13. Hastie T, Tibshirani R, Friedman J. Springer series in statistics the elements of statistical learning data mining, inference, and prediction second edition [Internet]. 2009. Available from: https://www.sas.upenn.edu/~fdiebold/NoHesitations/BookAdvanced.pdf

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

14. Aryanie D, Heryadi Y. American sign language-based finger-spelling recognition using k-Nearest Neighbors classifier. In 2015 3rd International Conference on Information and Communication Technology (ICoICT) 2015 May 27 (pp. 533-536). IEEE.
15. Ng, A. (n.d.). Machine Learning Yearning. [Online]. Chapter 5.
16. Cortes C, Vapnik V. Support-vector networks. Machine Learning [Internet]. 1995 Sep;20 (3):273–97. Available from: https://link.springer.com/article/10.1007/BF00994018
17. Quinlan JR. Induction of decision trees. Machine Learning [Internet]. 1986 Mar;1(1):81–106. Available from: http://www.hunch.net/~coms-4771/quinlan.pdf
18. Breiman L. Random Forests. Machine Learning [Internet]. 2001;45(1):5–32. Available from: https://link.springer.com/article/10.1023/a:1010933404324
19. Bajaj Y, Malhotra P. American sign language identification using hand trackpoint analysis. International Conference on Innovative Computing and Communications: Proceedings of ICICC 2021, Volume 1 2022 (pp. 159-171). Springer Singapore.
20. H. Orovwode, I. D. Oduntan and J. Abubakar, "Development of a Sign Language Recognition System Using Machine Learning," 2023 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD), Durban, South Africa, 2023, pp. 1-8, doi: 10.1109/icABCD59051.2023.10220456.
21. Alhafdee AH, Abbas H, Shahadi HI. Sign language recognition and hand gestures review. Kerbala Journal for Engineering Sciences. 2022 Dec 1;2(4):209-34.

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.