applied system innovation

MDPI

Article

Gesture-to-Text Translation Using SURF for Indian Sign Language

Kaustubh Mani Tripathi [1], Pooja Kamat [2,*], Shruti Patil [3], Ruchi Jayaswal [2], Swati Ahirrao [4] and Ketan Kotecha [3,*]

[1] Department of Computer Science & Information Technology, Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune 412115, Maharashtra, India
[2] Department of AI and Machine Learning, Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune 412115, Maharashtra, India
[3] Symbiosis Centre for Applied Artificial Intelligence (SCAAI), Symbiosis Institute of Technology, Symbiosis International (Deemed University), Pune 412115, Maharashtra, India
[4] Wipro Ltd., Pune 411045, Maharashtra, India
* Correspondence: pooja.kamat@sitpune.edu.in (P.K.); director@sitpune.edu.in (K.K.)

Abstract: This research paper focuses on developing an effective gesture-to-text translation system using state-of-the-art computer vision techniques. The existing research on sign language translation has yet to utilize skin masking, edge detection, and feature extraction techniques to their full potential. Therefore, this study employs the speeded-up robust features (SURF) model for feature extraction, which is resistant to variations such as rotation, perspective scaling, and occlusion. The proposed system utilizes a bag of visual words (BoVW) model for gesture-to-text conversion. The study uses a dataset of 42,000 photographs consisting of alphabets (A–Z) and numbers (1–9), divided into 35 classes with 1200 shots per class. The pre-processing phase includes skin masking, where the RGB color space is converted to the HSV color space, and Canny edge detection is used for sharp edge detection. The SURF elements are grouped and converted to a visual language using the K-means mini-batch clustering technique. The proposed system's performance is evaluated using several machine learning algorithms such as naïve Bayes, logistic regression, K nearest neighbors, support vector machine, and convolutional neural network. All the algorithms benefited from SURF, and the system's accuracy is promising, ranging from 79% to 92%. This research study not only presents the development of an effective gesture-to-text translation system but also highlights the importance of using skin masking, edge detection, and feature extraction techniques to their full potential in sign language translation. The proposed system aims to bridge the communication gap between individuals who cannot speak and those who cannot understand Indian Sign Language (ISL).

Keywords: classification models; gesture to text; skin masking; speeded-up robust features; SURF; feature extraction

Citation: Tripathi, K.M.; Kamat, P.; Patil, S.; Jayaswal, R.; Ahirrao, S.; Kotecha, K. Gesture-to-Text Translation Using SURF for Indian Sign Language. *Appl. Syst. Innov.* **2023**, *6*, 35. https://doi.org/10.3390/asi6020035

Academic Editor: Christos Douligeris

Received: 20 January 2023
Revised: 25 February 2023
Accepted: 27 February 2023
Published: 2 March 2023

1. Introduction

Sign language translation has become increasingly important for people with talking disabilities, with an estimated 1.57 billion individuals globally affected by hearing loss, representing around 20.3 percent of the world's population [1]. In India alone, 7 million deaf and mute people and 1.9 million people with speech disabilities rely on sign language as their primary means of communication, making it the world's 151st "spoken" language. Sixty-three percent are unemployed, and thirty percent have never attended school [2]. These individuals face significant challenges in their daily lives, such as limited access to education and employment opportunities, social exclusion, and difficulties in communication [3]. Due to their condition, they can only converse by using signs and gestures, which is further complicated by the diverse regional sign languages in India. Recent advances in sign language translation have vastly impacted the lives of these individuals. However,

challenges persist due to the evolution of the Indian Sign Language and the lack of generalization [4]. This literature addresses these challenges by proposing an ISL gesture-to-text translator using state-of-the-art machine learning techniques.

For this purpose, a camera is required to implement the ISL method. Gestures may be translated into ISL using ISL translation technology. The ISL translation framework receives images (from regular people) or continuous video clips, which the application then translates. Mute individuals adopt gestures to communicate with one another and convey themselves. Hand and other movements, facial emotions, and body postures make up sign language. However, since there are diverse groups of deaf and mute individuals all over the globe, their gestures will vary.

Various communities utilize the following sign languages to exchange ideas and converse. There are two forms of ISL: manual and non-manual signs. Manual gestures include one-handed and two-handed signs, in which the hands of the signer are used to express information.

This manuscript hopes to contribute to the ongoing research in this area by providing an ISL gesture to a text translator. For alphabet letters and numerals, a gesture recognition dataset was created. The attributes would be derived from segmented data by utilizing picture pre-processing, the bag of visual words model, which greatly benefits from the SURF (speeded-up robust features) feature extractor. These alphabetic characters are mapped to photos using histograms. Within the final stage, such qualities will be classified using supervised models.

This research employed some of the best state-of-the-art machine learning techniques to achieve the desired results. In particular, the study used speeded-up robust features (SURF) instead of a scale-invariant feature transform (SIFT) to make our model faster, more robust, and more reliable. It was then used by one of the most important machine-learning tools to create a visual vocabulary for BoVW. BoVW is a powerful method for feature extraction and classification in computer vision. Using SURF ensures that the vocabulary is invariant to the scale, orientation, and rotational changes of signs, which is crucial for sign language recognition, where different people may sign the same word differently.

The BoVW method is especially relevant for sign language recognition, as it can handle object appearance, pose, and illumination variations. In addition, the classification phase of the study was performed using support vector machine (SVM) naïve Bayes, logistic regression, K nearest neighbors (k-NN), as well as convolutional neural network (CNN). The utilization of these techniques led to the high accuracy achieved during this study.

This research paper presents an ISL translator that utilizes SURF for the gesture to text translation. The paper's organization is as follows: the study is divided into sections: Introduction, Prior Research, Research Methodology, Results and Discussion, and Conclusion. Section 2, Prior Research, aims to cover a range of studies and present a comparative analysis of their methodologies, techniques, and outcomes. By comparing prior research, the paper aims to build upon existing knowledge and identify gaps this study seeks to address. Section 3 of the research methodology provides a detailed description of the system setup and experimental procedures, including image collection, pre-processing, and feature extraction, where SURF and BoVW methods are discussed and incorporated. In Section 3.5, SVM, CNN, k-NN, naïve Bayes, and logistic regression are utilized. The subsection on image pre-processing discusses color space concepts, conversion of RGB to the HSV color space, and edge detection using Canny edge. Section 4, Results and Discussion, presents a comparative analysis and the final output. Finally, the conclusion section summarizes the key findings and contributions of the study and highlights the implications of the research; moreover, this section discusses possible future work.

2. Prior Research

Jadhav K. et al. [5] suggested a software system that included programming languages, machine learning, AI, and natural language processing (NLP). The researchers incorporated Python's PyAudio library for audio input. The recorded audio was then transformed into

speech using a speech-recognizing library such as the Google speech API (application program interface). The dependency parser helped to determine the underlying structure of a sentence and the relationships between its various components. This information was crucial for various NLP tasks, such as text classification, machine translation, and information extraction, as it helped better understand a sentence's meaning and context. The text was then divided using natural language processing to remove any confusion or ambiguity in the input sentence, resulting in a machine representation language in Python.

Kulkarni A. et al. [6] aimed to bridge the communication gap between individuals who were unable to speak and those who did not understand ISL. Their research was carried out in three categorical testing phases, including inflection handling checking, tense conversion checking, and stop word removal. The goal was to provide a useful and accessible solution for individuals who struggled with communication due to speech or hearing difficulties.

TESSA [7] is a speech-to-British sign language translator that was designed to assist deaf persons in communicating with postal clerks. This procedure employed a mathematical, linguistic method. A term searching library and a set of predetermined words were used in the interpretation. However, the interactivity between both participants was restricted since there were only a few statements to employ as templates. In conclusion, TESSA was a domain-specific solution.

Jayadeep G. et al. [8] came up with a combination of LSTM (long short-term memory) and CNN (convolutional neural network) for the dynamic identification of hand gesture methods for solitary words.

GrieveSmith's SignSynth effort [9], which used the ASCII-Stokoe technique to describe signals, was created in 1998 and 1999. This displayed an animated result obtained by translating ASCII-Stokoe into VRML (virtual reality modeling language). Utilizing proper transfer linguistic forms that fulfilled both the input and the output languages, any input phrase or conversation could be converted into a syntactically and semantically acceptable goal language inside this design.

Sruthi C.J. and Lijiya A. [10] curated a sign-recognizing model named "Signet", which was created using the CNN technique, supervised in nature, and succeeded in learning all 24 ISL static alphabets.

Malu S. Nair et al. [11] explored a function that took Malayalam data as the input for one or more phrases, resulting in a 3D character motion. The design incorporated a dynamic sign editor that transcribed indications into the HamNoSys (Hamburg Notation System for Sign Languages) schema to express signs, allowing users to add new terms to the collection. At the time, this service had 100 core communication terms. Its sign editor elements were used to insert these 100 words and the associated accompanying HamNoSys into the program.

P.V.V. Kishore et al. [12] suggested that sign language gestures expressed at different body areas be described as 3D motionlets, a subgroup of joint movements representing the signs. A 2-phase rapid approach extracted 3D inquiry signs out of a 3D gesture collection that had been dynamically sorted. Most human joints were divided into nonemotion joints (NMJ) and mobility joints (MJ) in Phase I. The relationship between NMJ and MJ was investigated to divide the dataset into four motion let categories. The Phase II approach investigated the relationships between the movement joints to express a sign's shape information as 3D motionlets. Three adjustable motionlet kernels were also included in the four-class sign dataset. The dataset was sorted and per the top-rated query sign using a simple kernel matching method.

Krori K. et al. [13] developed a minimal eigenvalue technique to train the model with dual-handed sign language. The picture material was collected with a Logitech webcam, and the analysis was carried out in MATLAB. Following the obtainment of Shi-Thomasi's excellent characteristics, the appropriate result was provided. Once contrasted to other alphabets, the test set of the alphabet A had the highest score (284) in the collection. Likewise, the letters F, S, L, Y, and B were counted and presented against record photos. The relevant result was shown as textual and afterward converted by correlating the feature

points of the testing data with the record in the database. Speech could be generated using the COM server in MATLAB.

Mahesh M. et al. [14] offered an Android application that transformed sign language into normal English, allowing deaf and impaired people to communicate using smartphones. The app took a picture with the device's camera, processed it, and found an appropriate gesture. Only those movements near the test sample were exposed to a Rotated BRIEF (binary robust independent element features) and oriented, fast-driven evaluation, which saved computational time. The app's users could also add new gestures to the same database.

Ebey Abraham et al. [15] designed a project to capture data on activities. They used a sensor glove with bendable sensors that detected the twisting of every finger and an inertial measurement unit to assess the hand's direction. The information was remotely sent and categorized as speech output. They explored and applied LSTM networks to classify gestural information, which have the capacity to study long-term correlations. The constructed model correctly classified 26 movements with a 98 percent accuracy rate, demonstrating the viability of utilizing LSTM-based sign language translation. A gadget would have to be able to distinguish both dynamic as well as static motions to interpret ISL.

Yellapu Madhuri et al. [16] suggested an app that was a software phone-based collaborative advanced application created with the LABVIEW software. It was also capable of recognizing letters (A–Z) and numeric one-handed gesture forms (0–9). The outcomes were extremely congruent, reliable, precise, and accurate. This project investigated the numerous problems with this novel strategy to sign language recognition, which also used appearance-based selected features straight from a webcast documented with a traditional webcam to recognize sign language alphanumeric characters on just one side, attempting to make the recognition system quite feasible. As a result, the geometric elements retrieved from the signer's dominant hand vastly enhanced the accuracy.

Purva C. Badhe et al. [17] proposed a method in which embedded videos were collected for testing purposes, processed beforehand, and features were extracted from them, resulting in a code vector. The researchers compared the resulting code vector with an existing codebook to detect movement. The most effective pre-processing, feature extraction, and vector measurement methods were determined by the integrated output algorithm, 2D FFT Fourier feature output, and four-vector codebook LBG. The program was designed for 10 users, with physical information that included 10 numbers, 26 character letters, and 10 different sentences.

Table 1 provides an overview of the different algorithms used by various authors in their research to achieve accurate results in sign language recognition. The dataset acquisition methods used by each author and their corresponding accuracy rates can also be seen. This table provides information about the state-of-the-art techniques used in sign language recognition and the level of accuracy that can be achieved using different algorithms and datasets.

After analyzing these works, the authors were motivated to create a model by comparing the performance of SURF and SIFT for image feature extraction. This comparison is important as there is very limited research where direct benefits of using one over the other are provided and implemented. Implementing the BoVW model to a multi-class image classification problem is important, as previous research has focused more on binary image classification. Extensive pre-processing must be used to improve the speed and accuracy, and robust techniques such as skin masking and Canny edge detection become important and crucial to be implemented for mute people's betterment. It becomes crucial to compare and analyze how SURF impacts classification models and determine whether there are any similarities or dissimilarities present to more closely observe how these models uniquely behave despite being considered similar.

Table 1. Literature Survey.

Author Name	Algorithm	Dataset Acquisition	Accuracy
Anderson et al. [18]	Hidden Markov model (light HMM, multi-stream HMM, and tied-density HMM.	Camera using DCF (digitizer configuration format) file.	Light HMM—83.6%, multi-stream HMM—86.7%, tied density HMM—91.3%
Md. Sanzidul Islam et al. [19]	Multi-layered convolutional neural network.	Eshara-lipi dataset in JPG format, containing 1000 images divided 100 per class, which is 128 × 128 pixels.	95% test accuracy
Ashish Sharma et al. [20]	Deep learning-based models: Pre-trained VGG16 model, natural language-based output network, and hierarchical network.	Images were clicked on Redmi Note 5 pro, 20 megapixel camera. Where all the images were resized to 144 × 144 pixels. Total 15,000 images were stored in the dataset with 5500 images per alphabet.	VGG16—94.52%, natural language-based output—92%, hierarchical model—98.52% for two-hand and 97% for one-hand
S. Sharma et al. [21]	Trbaggboost: ensemble-based transfer learning.	Multi-set and multi-modal database of signals recorded for 100 common signs.	Average classification accuracy was 80.44%. Classification accuracy improves to 97.04% as the number of new user data increases.
Garima Joshi et al. [22]	Naïve Bayes (NB), support vector machine (SVM), and simple logistic (SL).	Jochen Triesch's dataset called "Triesch's dataset".	NB—83%, SVM—94.5%, SL—89%

3. Research Methodology

To begin this research, raw images containing characters from A to Z and numbers from 1 to 9 were collected. The research methodology employed in this study combines both descriptive and experimental research. The first phase of the study involved a pre-processing step using the technique of skin masking to segment the hands in the images, which were then passed to Canny edge detection for sharp edge detection.

Following this, feature extraction was carried out using the SURF model, resistant to rotation, perspective scaling, occlusion, and variation. The SURF features were clustered using mini-batch k means clustering and fed into the bag of visual words model. The output of the BoVW model was then classified using labels and matched with the appropriate sign language words. The combination of descriptive and experimental research methodologies allowed for a comprehensive analysis of the effectiveness of the proposed system. The system's accuracy was evaluated using several classification algorithms, including naïve Bayes, logistic regression, K nearest neighbors, support vector machine, and convolutional neural network.

The purpose of this research is to analyze the existing study and conduct even more versatile research based on modern tools and techniques that can take in gestures from the camera and convert them to text which will help to reduce the communication barrier between mute people and the rest of society.

3.1. Gesture-to-Text Translation

The study's flow can be understood with the aid of Figure 1, but a more accurate and effective model must be developed to conduct a more thorough investigation. All these problems are addressed and properly evaluated in this work. It is also crucial to discover a less complex model capable of performing the task.

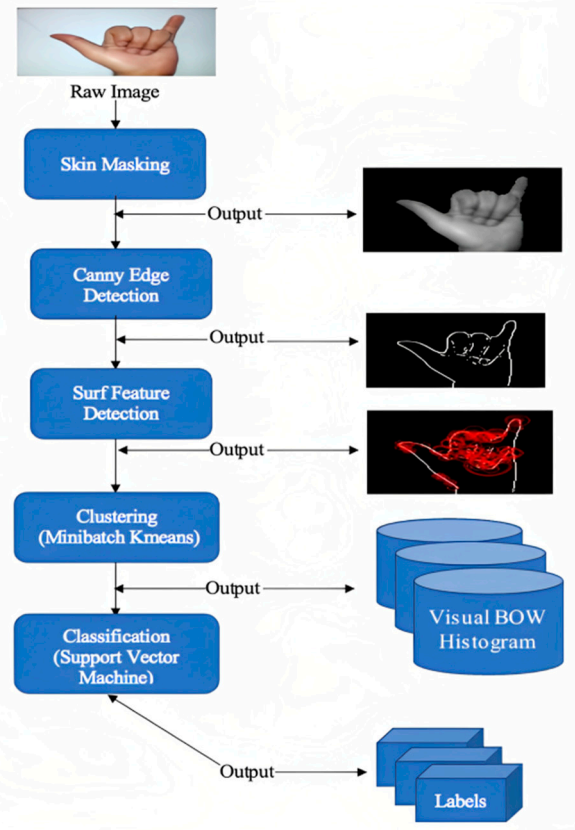

Figure 1. Flowchart of gesture-to-text translation.

Indian gesture recognition movements captured in still photos are detected by the technique utilized in this study. The structure is split into several processes, as shown in Figure 1: picture collection, image pre-processing (segmentation), feature extraction, and classification. The BoVW paradigm was used to identify photographs. The concept of BoVW was influenced by NLP. In image processing, the BoVW concept is a histogram-based representation of independent features. A picture could be used as a script to portray any movement utilizing the BoVW paradigm. Similarly, it is critical to describe each "word" in visuals. The following processes are usually included in this process: feature description and codebook generation (visual words). Histograms can be made for all the photos using these codebooks. SVM and other techniques such as CNN, k-NN, logistic regression, and the naïve Bayes model can classify photos.

3.2. Experimental Setup

This study is being carried out on a Python platform using Python 3.9. The SVM, CNN, logistic regression, naïve Bayes, and k-NN algorithms are applied and put into practice to evaluate the mode's efficacy. The integrated development environment VS code was used throughout the entire procedure (IDE). We combined all these datasets into one that was pre-labeled, and it contained 42,000 photos that were considered from various research articles. Factors such as clarity, resemblance, the background used in the image, and file size were considered while evaluating image datasets. The device on which the study was run had a 2.2 GHz processor, 201 GB of storage on the disc, and 8 GB of RAM. Our research began when these specifications were set. The study used various research articles and books to be conducted, along with technical expertise and theoretical knowledge of image processing. Different parameters for picture conversion, such as RGB to HSV, were also an important element of the study.

The system overview is detailed in Figure 2, which is divided into four stages. Stage 1 is the picture collection stage, where 42,000 photos of the alphabet (A–Z) and numbers (1–9)

are used. The next step is Stage 2, also known as image pre-processing, where an image is converted from RGB to HSV. Given that we are dealing with color-based object recognition, the HSV picture is utilized to isolate a certain area (i.e., a hand-making gesture). This strategy aids in background minimization so that our model may concentrate on the object crucial to the research. The next phase is Canny edge detection, a multi-part procedure that begins with smoothing the image to minimize noise, locating the image's gradient to highlight edges, and using non-maximum suppression to obtain thin edges. Following this, we proceed to the third stage, feature extraction, which begins with the SURF feature extractor, which, in contrast to SIFT, is resistant to perspective scaling, rotation, variation, and occlusion. After passing through K, mini batch clustering, visual words, and finally, visual vectors are produced. The final stage employs SVM (support vector machine) to classify and forecast an output after this.

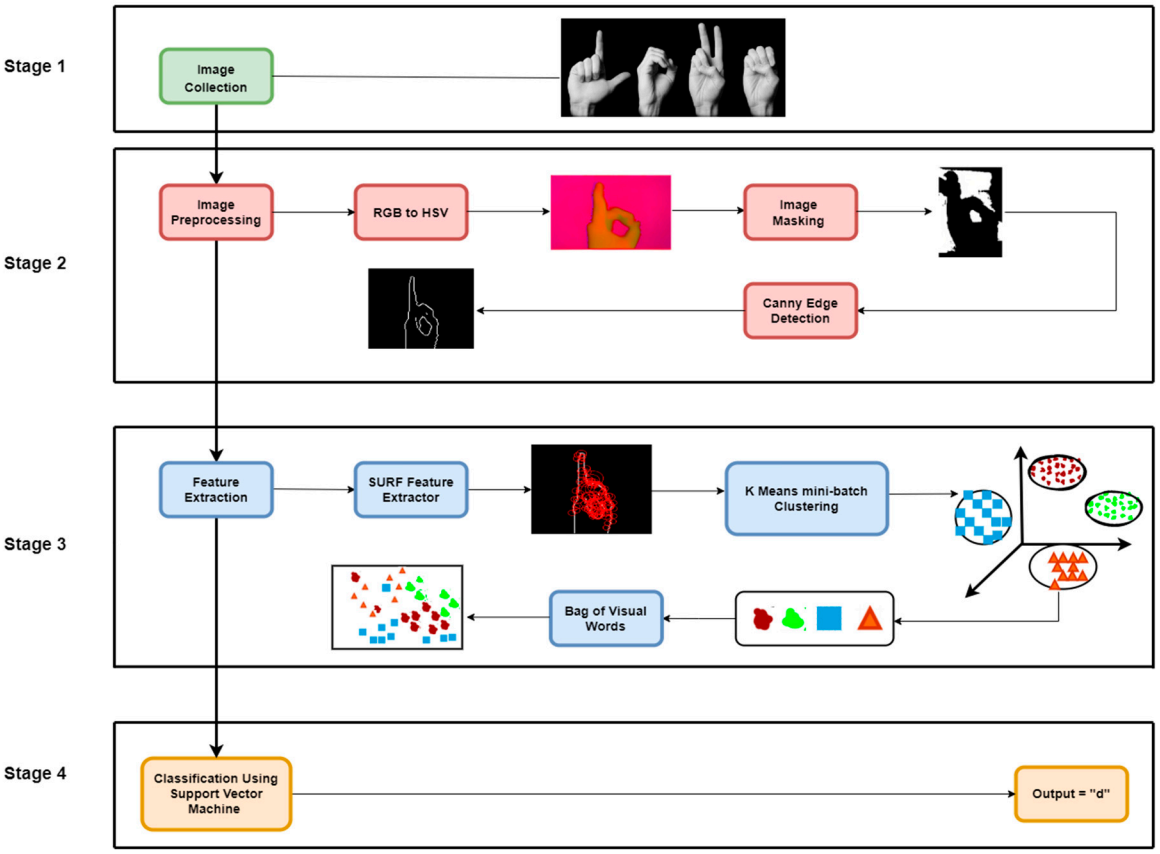

Figure 2. Overview of the system.

A proper data set for ISL currently needs more study in this subject. As a result, an ISL data collection containing 35 classes and 1200 photos per class was developed. With 42,000 images, the alphabets (A–Z) and numerals (1–9) are divided into 35 classes. In this research, the dataset used for gesture recognition consisted of 26 alphabets (A–Z) and 9 numbers (1–9), resulting in 35 classes. Each class corresponds to a specific alphabet or number. For example, class 'A' represents the gesture for the letter A, while class '1' represents the gesture for the number 1. Combining the alphabets and numbers into a single dataset with 35 classes enables the gesture recognition system to recognize both letters and numbers, which are commonly used in everyday communication. Combining the alphabet and numbers into a single dataset also helps to simplify the overall dataset and reduce the computational complexity of the system. Having each alphabet/number in a separate class allows the gesture recognition system to differentiate between the different gestures and accurately classify them. By training the system with many examples for each class, the system can learn to recognize gestures with high accuracy.

Webcam video is used to capture all the photographs for each class. Every video frame is preserved as a picture. Gestures were taken using a black background to decrease noise. Figure 3 depicts all motions associated with courses.

Figure 3. Collection of signs for alphabets and numbers [23].

3.3. Image Preprocessing

All photos are pre-processed in this phased way and will be utilized to extract feature image segmentation (skin masking), skin detection, and edge detection, which are the three approaches used in this phase, the order of which can be observed in (Figure 4).

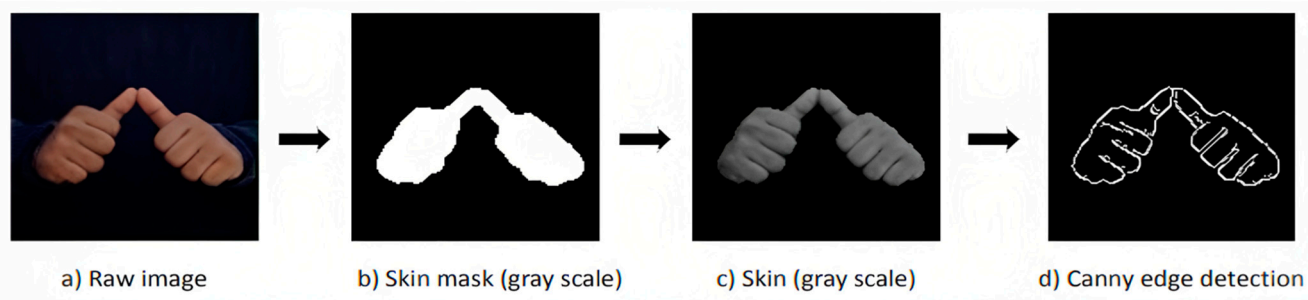

Figure 4. Steps involved in image preprocessing.

3.3.1. Concept of Color Space and Conversion of RGB to HSV Color Space

The method of distinguishing between the skin and non-skin pixels is known as the color of skin detection. It is the first stage in identifying possible territories. This is the first stage in locating areas in a picture that may contain human limbs and features. The following parameters must be considered throughout the skin-color-detecting process: (1) The equipment used to capture the photograph and various cameras produce different results for the same image; (2) within the image, the differentiation or segregation of skin and non-skin pixels; (3) the color space was employed to detect or segment the data; (4) color distortion caused by object motion decreases image resolution; (5) skin tones differ between individuals; (6) silhouettes and lighting play an essential part in changing the image's hue; and (7) does the image's brightness vary dramatically?

During skin detection, a variety of color spaces are accessible. RGB, normalized color space, luminance-based color space (HSI, HSV, and HSL), and hue-based color system (HSI, HSV, and HSL) are all RGB-based color spaces (YCbCr, YIQ, and YUV). The normalization process is incorporated for RGB conversion into normalized RGB [24].

$$\mathrm{r} = \frac{R}{R+G+B} \tag{1}$$

$$\mathrm{g} = \frac{G}{R+G+B} \tag{2}$$

$$\mathrm{b} = \frac{B}{R+G+B} \tag{3}$$

$$\mathrm{r} + \mathrm{g} + \mathrm{b} = 1 \tag{4}$$

$$\mathrm{b} = 1 - \mathrm{r} - \mathrm{g} \tag{5}$$

Normalizing RGB is a quick and easy technique to get the job done when it comes to reducing shadow and lighting distortions. However, when irregular luminescence is present, hue (HSV) and luminance (YCbCr)-based techniques are used, as these approaches identify color and intensity data even in the presence of uneven illumination.

Now, the calculation of m and M using the relations given the three parameters, R, G, and B (all within 0 to 255) [25], is as follows:

$$\mathrm{M} = \max\{\mathrm{R}, \mathrm{G}, \mathrm{B}\} \tag{6}$$

$$\mathrm{m} = \min\{\mathrm{R}, \mathrm{G}, \mathrm{B}\} \tag{7}$$

$$\mathrm{V} = \frac{M}{255} \tag{8}$$

$$\mathrm{S} = \begin{cases} 1 - \frac{m}{M}; & M > 0 \\ 0 \quad\quad ; & M = 0 \end{cases} \tag{9}$$

$$\mathrm{H} = \begin{cases} cos^{-1}\left[\frac{R - \frac{1}{2}G - \frac{1}{2}B}{\sqrt{R^2+G^2+B^2-RG-RB-GB}}\right]; G \geq B \\ 360 - cos^{-1}\left[\frac{R - \frac{1}{2}G - \frac{1}{2}B}{\sqrt{R^2+G^2+B^2-RG-RB-GB}}\right]; B \geq G \end{cases} \tag{10}$$

The following explains the technique to detect human skin color in color images.

1. The user's image is created by dividing the webcam video into image frames.
2. Through conversion, the RGB color space of a source image is converted to the HSV color space. A hue, saturation, and value (HSV) image comprises three separate images.
3. A histogram is constructed for every one of the three aspects, and the threshold value for every element is calculated from the histogram.
4. For skin pixels, the masking concept is applied.
5. The masked picture is given a criterion (threshold).
6. The threshold picture is flattened and refined.
7. At last, just the flesh pixels remain viewable.

The skin mask is constructed by converting the raw picture into the HSV color space (Figure 4). Skin pixels were defined by pixels there in the (H, S, V) range of (0,40,30) to (43,255,254). A skin mask could be used to and isolate the hand in the image. Lastly, the Canny edge method [25] can detect and identify abrupt interruptions in a picture, allowing the picture's edges to be recognized.

3.3.2. Edge Detection Using Canny Edge

The most important information in a photograph is image edge data, which can indicate the relative position inside the target region, target frame, and other pertinent details. Among the most important processes in image processing is edge discovery, and

the outcomes of the discovery directly impact picture analysis. In 1986, Canny introduced three performance metrics for edge detection operators and found the most effective to be the Canny edge detection operator [26]. The conversion of the image from the BGR format to the Canny edge image happens in the hierarchal format in the order explained in Figure 5. The output of the Canny edge detection is shown in Figure 6, where it can be observed that the boundary of the hand gesture is extracted in the Canny edge output.

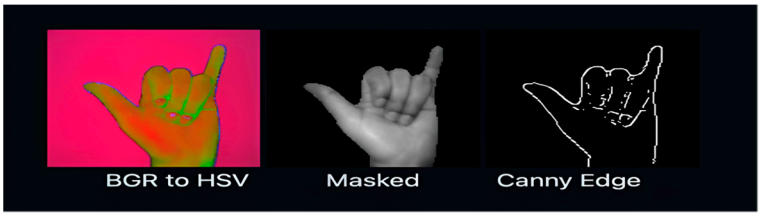

Figure 5. BGR to Canny edge conversion.

Figure 6. Character "D" after Canny edge detection.

The algorithm for the Canny edge detection [26].

- Smooth the picture with the Gauss function:

$$G(x,\ y) = \frac{exp\left[-\left(x^2+y^2\right)/2\sigma^2\right]}{2\pi\sigma^2} \tag{11}$$

where σ denotes a gauss filter parameter.

- Differential operators are used to determine the amplitude gradient and orientation.

$$Ex[i, j] = (I[I + 1, j] - I[i, j] + I[I + 1, j + 1] - I[i, j + 1])/2 \tag{12}$$

$$Ey[i, j] = (I[i, j + 1] - I[i, j] + I[I + 1, j + 1] - I[I + 1, j])/2 \tag{13}$$

$$G_x = \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix} \tag{14}$$

$$G_y = \begin{pmatrix} -1 & 1 \\ -1 & 1 \end{pmatrix} \tag{15}$$

It is possible to quantify the size and direction of the slope. Its image gradient is of the following magnitude:

$$\left|\left|M(i,\ j)\right|\right| = \sqrt{E_x[i,\ j]^2 + E_y[i,\ j]^2} \tag{16}$$

The azimuth of the image gradient is:

$$\theta(i,\ j) = \tan^{-1}\left(\frac{E_y[i, j]}{E_x[i,\ j]}\right) \tag{17}$$

- Select a non-maximal number reduction approach (NMS) to obtain potential edge points.

- Double threshold these potential candidate edge points.
- Join the remaining potential edge points that are linked to edge points to obtain the picture edge detection findings.

3.4. Feature Extraction

The next phase after image collection is the extraction of features. This phase involves three stages: image feature identification, clustering, and codebook preparation for the BoVW model. Initially, the scale-invariant feature transform (SIFT) method was used to detect significant image features [27].

However, we later switched to using resilient features provided by the speeded-up robust features (SURF) technique [28]. SURF is a groundbreaking feature extraction technique that offers improved speed, which is essential for designing real-time gesture recognition applications. The technique achieves its speed advantage by using a box filter approximation, which approximates the Laplacian of Gaussian (LoG) used in SIFT. SURF is also more robust than SIFT for image scaling and rotation. It uses a scale-space pyramid and a rotation-invariant descriptor to handle variations in scale and rotation, respectively. In contrast, SIFT also uses a scale-space pyramid, but its descriptors are not inherently rotation-invariant and require additional computation to achieve rotation invariance. For these reasons, we implemented SURF in our study.

Figure 7 shows what the image's SURF features look like. The next stage is to build a visual language by grouping all the identical SURF elements. It is improbable that humans can locate all relevant feature descriptors. The K-means clustering strategy may be effective in this scenario.

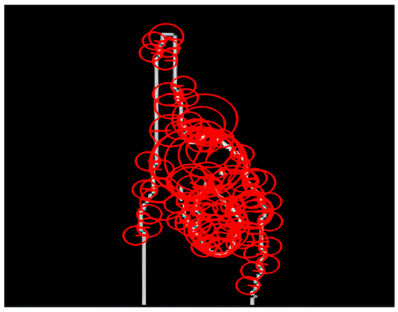

Figure 7. Character "D" after SURF implementation.

The K-means algorithm is a regularly used unsupervised clustering algorithm [29]. Its purpose is to divide n features into k clusters and use the cluster mean to forecast a new feature for each cluster (centroid). K-means clustering takes a long time and much memory because much work is done with SURF features from 42,000 photographs. As a result, micro-batch K-means, comparable with K-means but require less memory and are faster, were deployed [30].

After micro batch K-means has learned all SURF characteristics, it puts all related features into a bag with k clusters (visual words). A K value of 280 was used because there are 35 classes to categorize. A series of actions are taken to produce the desired outcome: a visual word, a feature vector representation of an image, and each vector dimension correspond to a visual word in the image.

In picture classification and retrieval activities, these vectors are employed. To evaluate and categorize photos, these ideas and methods are utilized in combination in the manner shown in (Figure 8). This is carried out by extracting significant aspects from the images, putting related features together, and then representing the images with a set of visual words. The recurrence of every visual word linked with the image in overall visual words is used to compute the histogram.

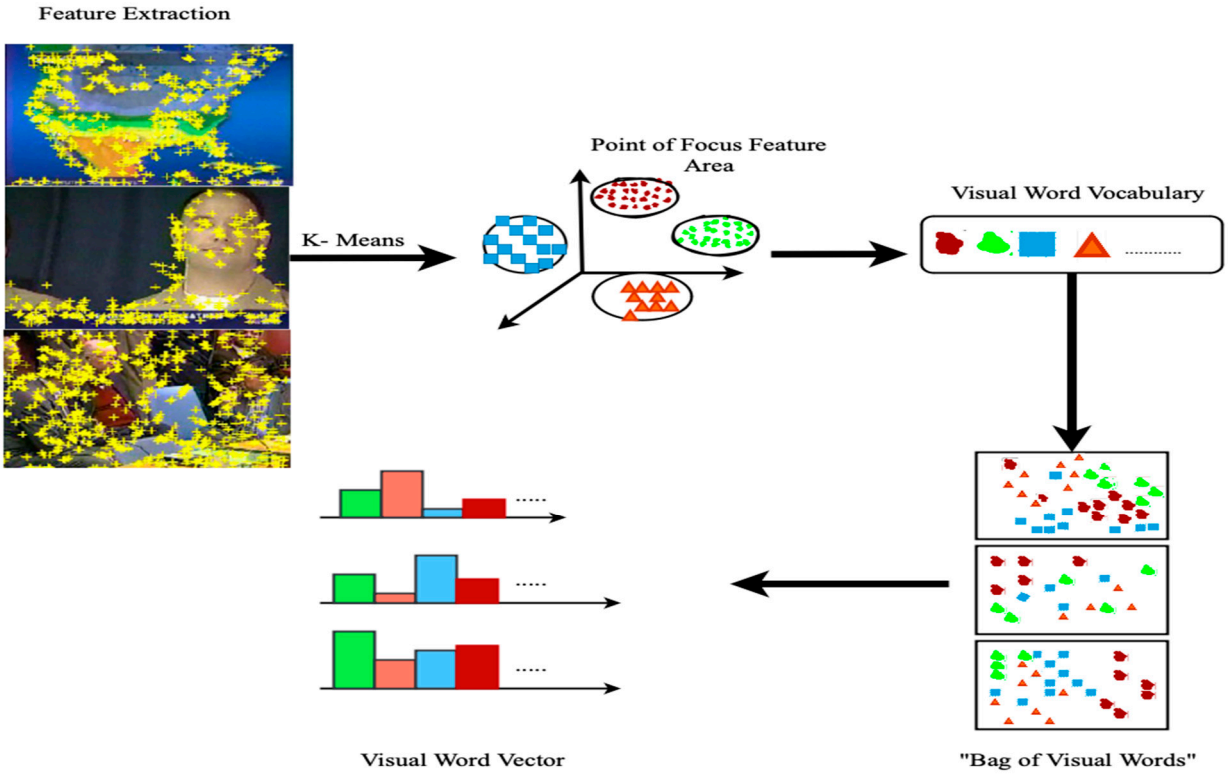

Figure 8. Combination of steps used for extracting features and ultimately converting it to a visual word vector [31].

3.5. Classification

The categorizing method begins after all statistics for the entire data set have been created. It is vital to split the dataset for training and testing before going on to classification. As a result, the entire dataset was split into 70:30 classes, with 960 photos used for training as well as 240 pictures in use for testing. After the training set is available, the following step is to input it into the machine learning model. The data were trained and forecasted using SVM, CNN, k-NN, logistic regression, and naïve Bayes. Despite being considered a similar model, each has unique strengths and limitations, as SVM can effectively handle high-dimensional data and complex classification problems.

Additionally, SVM has a solid theoretical foundation and is less prone to overfitting than other algorithms. However, one potential limitation of SVM is that it can be computationally expensive; SVM also relies heavily on the choice of the kernel function, and selecting an inappropriate kernel can lead to poor performance. During our research, we observed that linear kernels showed the best results. Naïve Bayes is known for its simplicity and speed, making it a popular choice for text classification and other simple classification tasks. However, it strongly assumes that all features are independent, which can limit its accuracy for more complex problems. Logistic regression is relatively simple and fast but can handle more complex relationships between features than naïve Bayes. However, it may struggle with high-dimensional datasets or non-linear relationships between features. k-NN is non-parametric, meaning it does not make any assumptions about the underlying distribution of the data. This can make it more flexible and accurate than some parametric methods but can also make it computationally expensive for large datasets. CNN is a powerful tool for image recognition tasks and has achieved state-of-the-art performance on various datasets. However, it is also complex and computationally expensive and can be difficult to train effectively without large amounts of data.

It also became extremely important to understand the importance of explainable AI, which is crucial in decision making [32]. Real-time recognition was also created, allowing users to predict real-time motions while watching a video feed.

In Figure 9, the x-axis represents the predicted label, which denotes the class or category our model assigned to that instance. In contrast, the y-axis represents the true label, which refers to the actual category of an instance in the dataset. The count or frequency of instances falling into each combination of true and expected labels is contained in the cells of the confusion matrix. In our example, the letters A–Z and the numerals 1–9 will create a 36 × 36-dimension confusion matrix. Each cell will store the number of times the true label is either a number or a letter and the predicted label is a different number or letter. The confusion matrix column with the value 240 indicates 240 cases in which both the true and predicted labels correspond to the given combination.

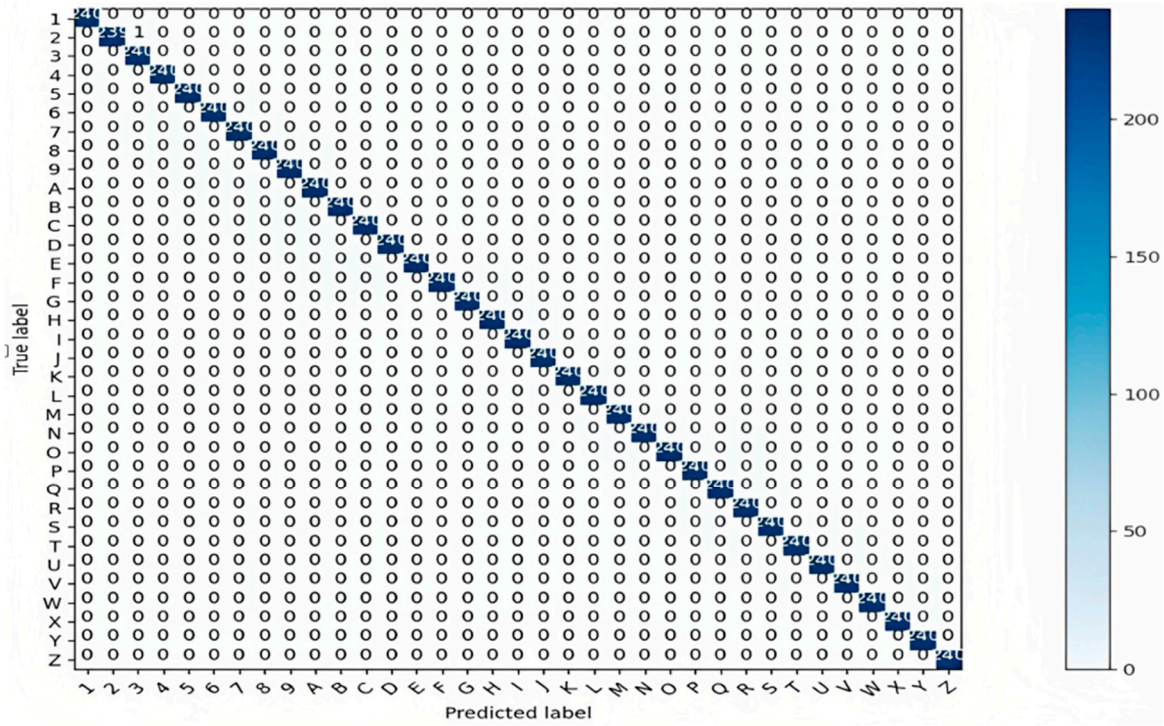

Figure 9. Confusion matrix.

4. Results and Discussion

Our study evaluated the performance of several classification algorithms (naïve Bayes, logistic regression, K nearest neighbors, support vector machine, and convolutional neural network) in recognizing hand gestures for Indian Sign Language translation. Figures 10 and 11 can be observed to compare the accuracy achieved by each algorithm, with SVM achieving the highest accuracy but requiring more computation time and naïve Bayes having lower accuracy but being faster.

This study also evaluated the effectiveness of various pre-processing techniques, such as skin masking, Canny edge detection, and feature extraction with SURF. We found that these techniques contributed significantly to the accuracy of the classification models, improving the system's overall performance. It became very fruitful for us to switch from SIFT to SURF as it significantly sped up processing. It can also be well observed in Table 2 that our pre-processing techniques performed better in improving the quality of the input data by reducing noise, enhancing contrast, and normalizing the illumination of the images. Using the correct pre-processing tools and techniques is essential to ensure that the input data are consistent and reliable.

If we analyze Figures 10 and 11, implementing the SURF feature extraction technique improved the accuracy of all the classification models. Naive Bayes, logistic regression, K nearest neighbors, and support vector machine showed a significant increase in accuracy when using SURF, with K nearest neighbors showing an increase of 27%. Convolutional neural networks, on the other hand, did not show any improvement in accuracy with the

use of SURF. The naïve Bayes model showed the lowest accuracy without SURF, with only 39%, while K nearest neighbors showed the highest accuracy without SURF, 62%.

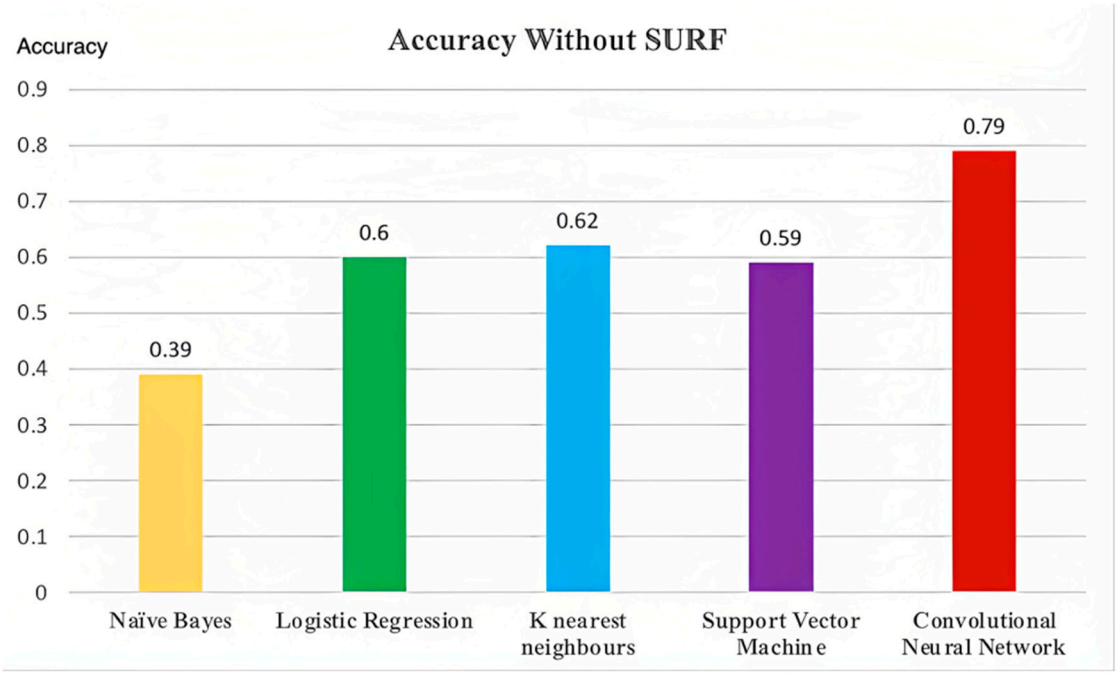

Figure 10. Accuracy of models without using SURF.

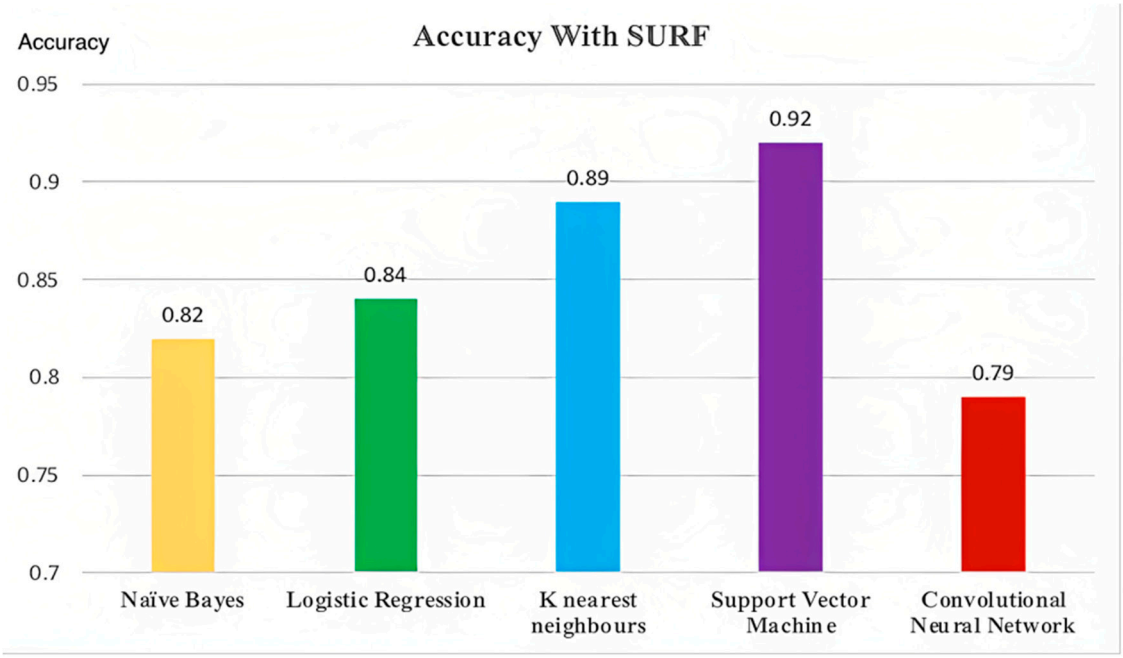

Figure 11. Accuracy of models with using SURF.

However, after implementing the SURF technique, the naïve Bayes model showed a drastic improvement in accuracy to 82%, which is even higher than the accuracy achieved by K nearest neighbors without SURF. This suggests that the naïve Bayes model highly depends on feature extraction and can greatly benefit from using techniques such as SURF. The logistic regression and support vector machine models also showed a significant improvement in accuracy with SURF, with an increase of 24% and 33%, respectively. This indicates that these models are also highly dependent on the feature extraction process, and

techniques such as SURF can significantly enhance their performance. K nearest neighbors showed the highest accuracy without using SURF but still showed a significant increase in accuracy after implementing the feature extraction technique. This indicates that K nearest neighbors is less dependent on feature extraction than the other models but can still benefit from it. Convolutional neural networks, on the other hand, did not show any improvement in accuracy with the use of SURF. This is because CNNs are designed to extract features automatically from the input data and may not require additional feature extraction techniques such as SURF.

Table 2. In contrast to alternative strategies outlined in other referred manuscripts.

Dataset Consisted of	Skin Detection	Feature Extraction	Accuracy	Reference No.
Static signs and alphabets	HSV	SIFT	Light HMM—83.6%, multi stream HMM—86.7%, and tied-density HMM—91.3%	[18]
Numbers 0–9	Not required	Softmax activation to use logistic regression on feature extraction which was connected before the final layer	95% test accuracy	[19]
Triesch's dataset	Skin detection was conducted during preprocessing using gamma variation	Hog feature extraction without segmentation	94.5% best performing accuracy during SVM implementation	[22]
A–Z Alphabets and 1–9 numbers	HSV, Canny edge detection	SURF	Word accuracy—88%, and Alphabet and number accuracy—96%	Proposed study

It can be observed from Figure 12 that the recognition rate of SVM using SURF is significantly better and consistent than for CNN without using SURF, which shows better results than CNN with using SURF, as observed earlier in Figures 10 and 11. It is important to generate a consistent model as it can help develop more efficient and accurate sign language recognition systems, thereby improving communication for people with hearing or speech impairments.

The system will check the gesture from the database where the user has already designed and taught several gestures after the threshold has been set. In this example, just the taught motions appear in the findings, which were trained using Google Collaboratory. The output file is related to the display gesture software in the form of an H5 extension. The best result came from the support vector machine. Finally, before proceeding to the next phase, the classification dataset was divided into a 70:30 ratio, and classification was applied. Overall, gesture recognition accomplished the system and produced satisfactory results, with a word accuracy of 88 percent for the alphabet and number correctness of 96 percent, which is shown in Table 3. Figures 13 and 14 show the correct real-time output for the number "1" and character "G"; here, we used SVM as the classification model.

Table 3. Accuracy of words and numbers.

Word accuracy	88%
Number accuracy	96%

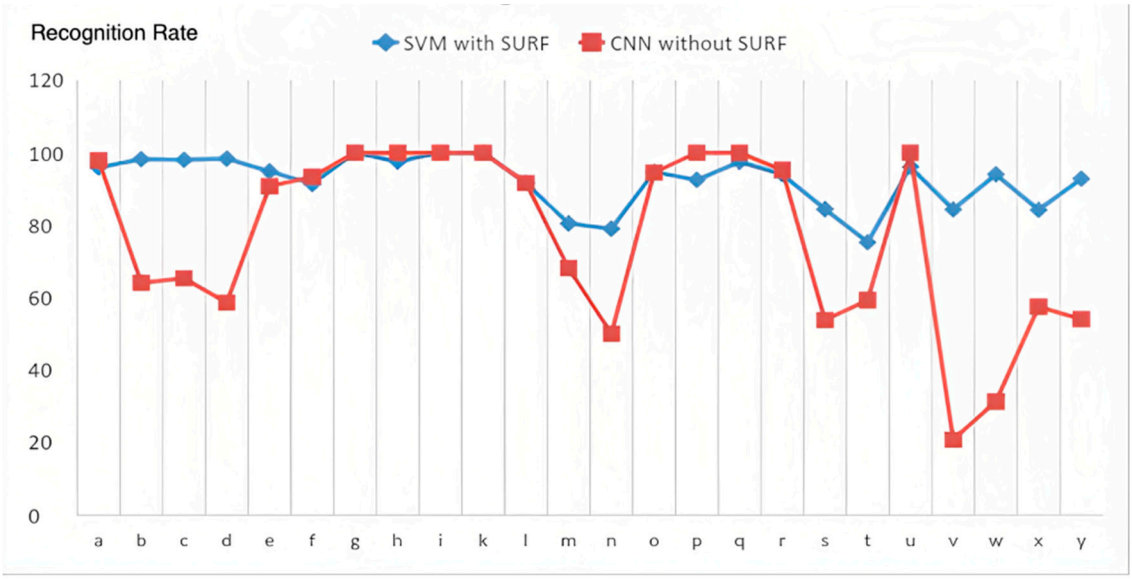

Figure 12. Recognition rate difference for alphabets using SVM with SURF and CNN without SURF.

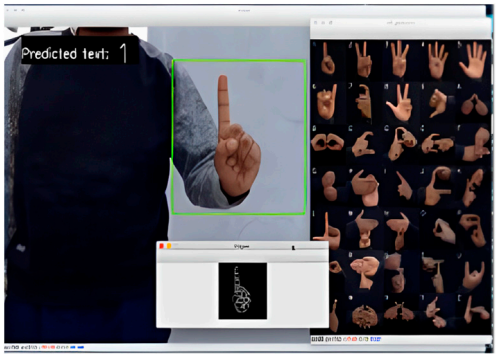

Figure 13. Output screen of gesture-to-text conversion successfully predicting 1 correctly.

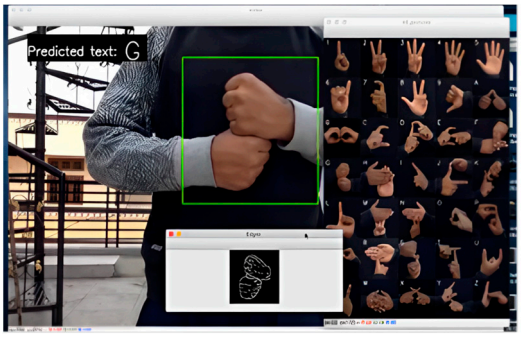

Figure 14. Output screen of gesture-to-text conversion successfully predicting G correctly.

5. Conclusions

In conclusion, our study evaluated the performance of various classification algorithms and pre-processing techniques for recognizing hand gestures in Indian Sign Language. Our findings suggest that the SVM algorithm achieved the highest accuracy but required more computation time, while the naïve Bayes model was faster but had lower accuracy. Implementing the SURF feature extraction technique significantly improved the accuracy of all classification models—naïve Bayes, logistic regression, K nearest neighbors, and support vector machine, except for the CNN, which did not show any improvement because they are designed to extract features automatically from the input data, and SURF, which is an external feature extraction technique that is not of any help here. Our study also showed

that pre-processing techniques such as skin masking, Canny edge detection, and feature extraction with SURF contributed significantly to the accuracy of the classification models by improving the quality of the input data.

Regarding recognition rate consistency, the SVM algorithm with SURF outperformed the CNN model without SURF. Generating a consistent model is crucial for developing more efficient and accurate sign language recognition systems, which can greatly improve communication for people with hearing or speech impairments. Our gesture recognition system achieved word accuracy of 88 percent for alphabets and number correctness of 96 percent using the SVM classification model, which produced satisfactory results in real time. Overall, the findings of our study can guide the development of more effective sign language recognition systems that can facilitate communication for individuals with hearing or speech impairments.

Finally, this study was an endeavor to contribute to making these amazing people's lives better by reducing the challenges they experience. Despite the study's early success, much more development may still be achieved in this area. We suggest exploring other feature extraction techniques and algorithms to further improve the accuracy and efficiency of the gesture recognition system. Additionally, we recommend investigating the system's robustness to different lighting conditions and camera angles to ensure its reliability in real-world applications.

Author Contributions: Conceptualization, P.K. and S.P.; methodology, P.K. and K.M.T.; software, K.M.T., R.J., P.K. and S.P.; validation, K.K. and S.A.; formal analysis, R.J. and P.K.; investigation, S.P. and K.K.; resources, P.K. and K.K.; data curation, S.A. and R.J.; writing—original draft preparation, K.M.T., R.J., P.K. and S.P.; writing—review and editing, K.M.T., P.K. and R.J.; visualization, S.P.; supervision, S.A.; project administration, P.K.; funding acquisition, K.K. and P.K. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded under the Research Support Fund of Symbiosis International (Deemed) University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Verma, M.P.; Shimi, S.L.; Chatterji, S. Design of smart gloves. *Int. J. Eng. Res. Technol.* **2014**, *3*, 210–214.
2. Sahoo, A.K. Indian sign language recognition using machine learning techniques. *Macromol. Symp.* **2021**, *397*, 2000241. [CrossRef]
3. Tyagi, A.; Bansal, S. Feature extraction technique for vision-based indian sign language recognition system: A review. *Comput. Methods Data Eng.* **2020**, *1*, 39–53.
4. Katoch, S.; Singh, V.; Tiwary, U.S. Indian Sign Language recognition system using SURF with SVM and CNN. *Array* **2022**, *14*, 100141. [CrossRef]
5. Jadhav, K.; Gangdhar, S.; Ghanekar, V. Speech to Isl (Indian Sign Language) Translator. *Int. Res. J. Eng. Technol.* **2021**, *8*, 3696–3698.
6. Kulkarni, A.; Kariyal, A.V.; Dhanush, V.; Singh, P.N. Speech to Indian Sign Language Translator. In *3rd International Conference on Integrated Intelligent Computing Communication & Security (ICIIC 2021)*; Atlantis Press: Zhengzhou, China, 2021; pp. 278–285.
7. Deora, D.; Bajaj, N. Indian sign language recognition. In Proceedings of the 2012 1st International Conference on Emerging Technology Trends in Electronics, Communication & Networking, Gujarat, India, 19–21 December 2012; pp. 1–5.
8. Jayadeep, G.; Vishnupriya, N.V.; Venugopal, V.; Vishnu, S.; Geetha, M. Mudra: Convolutional neural network based Indian sign language translator for banks. In Proceedings of the 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 13–15 May 2020; pp. 1228–1232.
9. Punekar, A.C.; Nambiar, S.A.; Laliya, R.S.; Salian, S.R. A Translator for Indian Sign Language to Text and Speech. *Int. J. Res. Appl. Sci. Eng. Technol.* **2020**, *8*, 1640–1646. [CrossRef]
10. Sruthi, C.J.; Lijiya, A. Signet: A deep learning based indian sign language recognition system. In Proceedings of the 2019 International Conference on Communication and Signal Processing (ICCSP), Tamil Nadu, India, 4–6 April 2019; pp. 596–600.
11. Nair, M.S.; Nimitha, A.P.; Idicula, S.M. Conversion of Malayalam text to Indian sign language using synthetic animation. In Proceedings of the 2016 International Conference on Next Generation Intelligent Systems (ICNGIS), Kottayam, India, 1–3 September 2016; pp. 1–4.

12. Kishore, P.V.V.; Kumar, D.A.; Sastry, A.C.S.; Kumar, E.K. Motionlets matching with adaptive kernels for 3-d indian sign language recognition. *IEEE Sens. J.* **2018**, *18*, 3327–3337. [CrossRef]
13. Dutta, K.K.; GS, A.K. Double handed Indian Sign Language to speech and text. In Proceedings of the 2015 Third International Conference on Image Information Processing (ICIIP), Waknaghat, India, 21–24 December 2015; pp. 374–377.
14. Mahesh, M.; Jayaprakash, A.; Geetha, M. Sign language translator for mobile platforms. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 13–16 September 2017; pp. 1176–1181.
15. Abraham, E.; Nayak, A.; Iqbal, A. Real-time translation of Indian sign language using LSTM. In Proceedings of the 2019 Global Conference for Advancement in Technology (GCAT), Bangaluru, India, 18–20 October 2019; pp. 1–5.
16. Madhuri, Y.; Anitha, G.; Anburajan, M. Vision-based sign language translation device. In Proceedings of the 2013 International Conference on Information Communication and Embedded Systems (ICICES), Chennai, India, 21–22 February 2013; pp. 565–568.
17. Badhe, P.C.; Kulkarni, V. Indian sign language translator using gesture recognition algorithm. In Proceedings of the 2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS), Odisha, India, 2–3 November 2015; pp. 195–200.
18. Anderson, R.; Wiryana, F.; Ariesta, M.C.; Kusuma, G.P. Sign language recognition application systems for deaf-mute people: A review based on input-process-output. *Procedia Comput. Sci.* **2017**, *116*, 441–448.
19. Islam, S.; Mousumi, S.S.S.; Rabby, A.S.A.; Hossain, S.A.; Abujar, S. A potent model to recognize bangla sign language digits using convolutional neural network. *Procedia Comput. Sci.* **2018**, *143*, 611–618. [CrossRef]
20. Sharma, A.; Sharma, N.; Saxena, Y.; Singh, A.; Sadhya, D. Benchmarking deep neural network approaches for Indian Sign Language recognition. *Neural Comput. Appl.* **2021**, *33*, 6685–6696. [CrossRef]
21. Sharma, S.; Gupta, R.; Kumar, A. Trbaggboost: An ensemble-based transfer learning method applied to Indian Sign Language recognition. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *13*, 3527–3537. [CrossRef]
22. Joshi, G.; Gaur, A. Interpretation of indian sign language using optimal hog feature vector. In *International Conference on Advances in Computing and Data Sciences*; Springer: Singapore, 2018; pp. 65–73.
23. Shivashankara, S.; Srinath, S. American sign language recognition system: An optimal approach. *Int. J. Image Graph. Signal Process.* **2018**, *11*, 18.
24. Dargham, J.A.; Chekima, A. Lips detection in the normalised RGB colour scheme. In Proceedings of the 2006 2nd International Conference on Information & Communication Technologies, Damascus, Syria, 24–28 April 2006; Volume 1, pp. 1546–1551.
25. Chernov, V.; Alander, J.; Bochko, V. Integer-based accurate conversion between RGB and HSV color spaces. *Comput. Electr. Eng.* **2015**, *46*, 328–337. [CrossRef]
26. Song, R.; Zhang, Z.; Liu, H. Edge connection based Canny edge detection algorithm. *Pattern Recognit. Image Anal.* **2017**, *27*, 740–747. [CrossRef]
27. Azhar, R.; Tuwohingide, D.; Kamudi, D.; Suciati, N. Batik image classification using SIFT feature extraction, bag of features and support vector machine. *Procedia Comput. Sci.* **2015**, *72*, 24–30. [CrossRef]
28. Mistry, D.; Banerjee, A. Comparison of feature detection and matching approaches: SIFT and SURF. *GRD J.-Glob. Res. Dev. J. Eng.* **2017**, *2*, 7–13.
29. Wei, P.; Zhou, Z.; Li, L.; Jiang, J. Research on face feature extraction based on K-mean algorithm. *EURASIP J. Image Video Process.* **2018**, *2018*, 83. [CrossRef]
30. Maharjan, A. Nepali Document Clustering using K-Means, Mini-Batch K-Means, and DBSCAN. Ph.D. Thesis, Department of Computer Science and Information Technology, Tribhuvan University, Kathmandu, Nepal, 2018.
31. Rong, W.; Li, Z.; Zhang, W.; Sun, L. An improved CANNY edge detection algorithm. In Proceedings of the 2014 IEEE International Conference on Mechatronics and Automation, Tianjin, China, 3–6 August 2014; pp. 577–582.
32. Rathod, M.; Dalvi, C.; Kaur, K.; Patil, S.; Gite, S.; Kamat, P.; Kotecha, K.; Abraham, A.; Gabralla, L.A. Kids' Emotion Recognition Using Various Deep-Learning Models with Explainable AI. *Sensors* **2022**, *22*, 8066. [CrossRef] [PubMed]