



Implementation of Kannada Sign Language Recognition using Machine Learning

Srijith S Bhat¹, Omkar K Hegde², Ganesh N Bhat³, Sanjeevini S M⁴, Ashwini D S⁵

Student, Department of ECE, NIEIT, Mysuru, India¹⁻⁴

Assistant Professor, Department of ECE, NIEIT, Mysuru, India⁵

Abstract: Technology is going very fast day by day; digital recognitions are getting more popular and providing more scope to perform research in ML (Machine Learning) and AI (Artificial Intelligence). Recognition of sign languages has been done by many researchers. Few researchers also tried recognizing Kannada sign languages with static images. Still real time Kannada sign language recognition is one of the least touched techniques. This project is trying to use Machine Learning techniques to do this job of Kannada sign language recognition with reasonable accuracy.

Keywords: ML, Python, Kannada, Sign Language

I. INTRODUCTION

Humans understand one another by convincing their ideas, thoughts, and experiences to the people around them. There are a unit varied ways that to realize this and therefore the best one of the remainders is the gift of speech. Through speech everybody will terribly convincingly transfers their thoughts and perceive one another. It will be injustice, if we have a tendency to ignore people who area unit empty this valuable gift. The sole suggests that of communication on the market to the vocally disabled is that the use of sign language communication. The exploitation sign language communication they're restricted to their own world. This limitation prevents them from interacting with the outer world to share their feelings, artistic ideas and potentials. Only a few people that do not seem to be themselves deaf ever learn to sign language communication. This limitation will increase the isolation of deaf and dumb people from the common society. Technology is a method to get rid of this hindrance and profit these people many researchers have explored these prospects and have with success achieved finger spelling recognition with high levels of accuracy. However, progress within the recognition of sign language communication, as a full has varied limitations in today's applications. Varied systems were projected for the automated recognition of sign language communication.

II. TOOLS UTILISED

We used several softwares for the project implementation. Mainly we used PyCharm IDE as code editor for python program files. PyCharm Community version is free and used for smart and intelligent Python development, including code assistance, refactoring, visual debugging, and version control integration. Different types of python libraries are used to implement the machine learning methods in the project. OpenCV is the open-source computer vision python library used for camera operations to accelerate the use of machine perception in the commercial products. Keras and TensorFlow are the libraries used for image classification and to convert the images into weighted model. Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research. TensorFlow is a mostly used open-source platform for machine learning process. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state of the art in ML and developers easily build and deploy ML powered applications. Pandas is a Python library for data analysis. Pandas is built on top of two core Python libraries which are matplotlib and NumPy, matplotlib for data visualization and NumPy for mathematical operations. Pandas acts as a wrapper over these libraries, allowing you to access many of matplotlib's and NumPy's methods with less code.

III. IMPLEMENTATIONS

This work aims to solve the "Recognition of Kannada Sign Language", by using machine learning. Which aims to provide more accuracy and reliable results and efficient of handling the data. By using machine learning we can build large data set. Understanding the exact context of sign expression of dumb and deaf people is the challenging job in real life until unless it is properly specified. Kannada alphabet has 13 vowels and 34 consonants. The effort is more focused on the identification of sign expressions.



Creating Dataset

We have taken the standard dataset of the hand gestures of the Kannada characters from the All-India Institute of Speech and Hearing (AIISH), which is located in Manasagangotri (Mysore University Campus), Mysuru. By referring the standard datasets, we have created 15000 hand gesture images approximately. We have taken 34 Kannada letters from 'ಕೆ' to 'ಳ' which is generally called as Vyanjanaakshara in Kannada. Images have been captured using mobile cameras and web cameras. Approximately 500 images have been taken for each letter. The images are captured in a lite white background and the positions of the images are randomly rotated for small angles to ensure the proper prediction in the output.

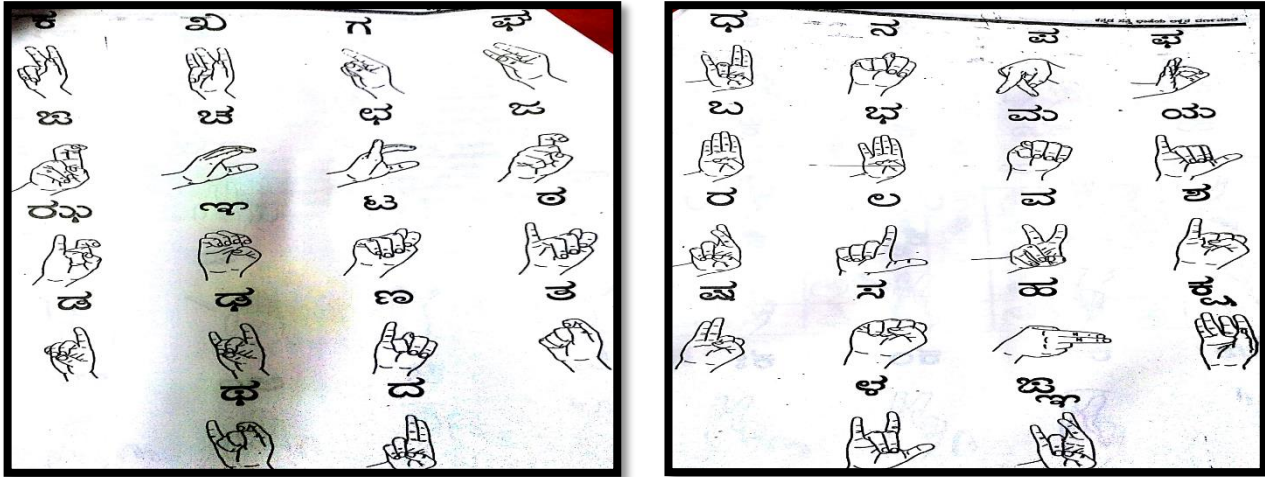


Fig 1. Standard Dataset

Training the model

The images are trained using Teachable Machine which makes use of transfer learning; A method that uses transferable knowledge from another model and refines it using data available for the task at hand. In the case of neural network models, the first approach uses a base model that does very well at some general tasks like classifying all types of images into a thousand different classes. We just strip out the last classifier layer, fit our own classification layer and train only this layer with new images. In this case, our base model simply functions as a 'fixed feature extractor' by creating a representation of an image that has captured generally relevant features. The second approach would go one step further and allow the weights of our base model to be altered during the training process. This is called 'fine-tuning' the model. So, the model will be downloaded in the form of "keras_model.h5" by making use of above process and Keras library. Teachable Machine uses TensorFlow.js, a library for machine learning in JavaScript, to train and run the models we make in our web browser. There's a pretrained neural network, and when users create their own classes, they can sort of picture that your classes are becoming the last layer or step of the neural net. Specifically, both the image and pose models are learning off of pretrained mobile net models, and the sound model is built on Speech Command Recognizer. The models which users make with Teachable Machine are real TensorFlow.js models that work anywhere JavaScript runs, so they play nice with tools like Glitch, P5.js, Node.js & more. Plus, export to different formats to use your models elsewhere, like Coral, Arduino & more.

Image Classification using CNN

Convolutional Neural Networks are the subdomain of Machine Learning and Deep Learning concepts. Algorithms under Deep Learning process information the same way the human brain does, but obviously on a very small scale, since our brain is too complex (our brain has around 86 billion neurons). Image classification involves the extraction of features from the image to observe some patterns in the dataset. Using an ANN for the purpose of image classification would end up being very costly in terms of computation since the trainable parameters become extremely large. It is a well-known method in computer vision applications. The teachable machine uses transfer learning; A method that uses transferable knowledge from another model and refines it using data available for the task at hand. In the case of neural network models, the first approach uses a base model that does very well at some general tasks like classifying all types of images into a thousand different classes. Stripping out the last classifier layer, fit our own classification layer and train only this layer with new images. In this case, our base model simply functions as a 'fixed feature extractor' by creating a representation of an image that has captured generally relevant features. This type of architecture is dominant to recognize objects from a picture or video. It is used in applications like image or video recognition, neural language processing, etc.



An image is composed of an array of pixels with height and width. A grayscale image has only one channel while the colour image has three channels (each one for Red, Green, and Blue). A channel is stacked over each other. In this tutorial, you will use a grayscale image with only one channel. Each pixel has a value from 0 to 255 to reflect the intensity of the colour. For instance, a pixel equals to 0 will show a white colour while pixel with a value close to 255 will be darker.

Image Comparison with the model

In the project “main.py” is the Python program used for image comparison. The Python libraries named TensorFlow, NumPy and Keras are used for the test image with the trained model. Once the model is loaded TensorFlow Similarity leverages Fast Approximate Nearest Neighbour search (ANN) to retrieve the closest matching items from the index in sub-linear time. This fast look up leverages the fact that TensorFlow Similarity learns a metric embedding space where the distance between embedded points is a function of a valid distance metric. These distance metrics satisfy the triangle inequality, making the space amenable to Approximate Nearest Neighbour search and leading to high retrieval accuracy.

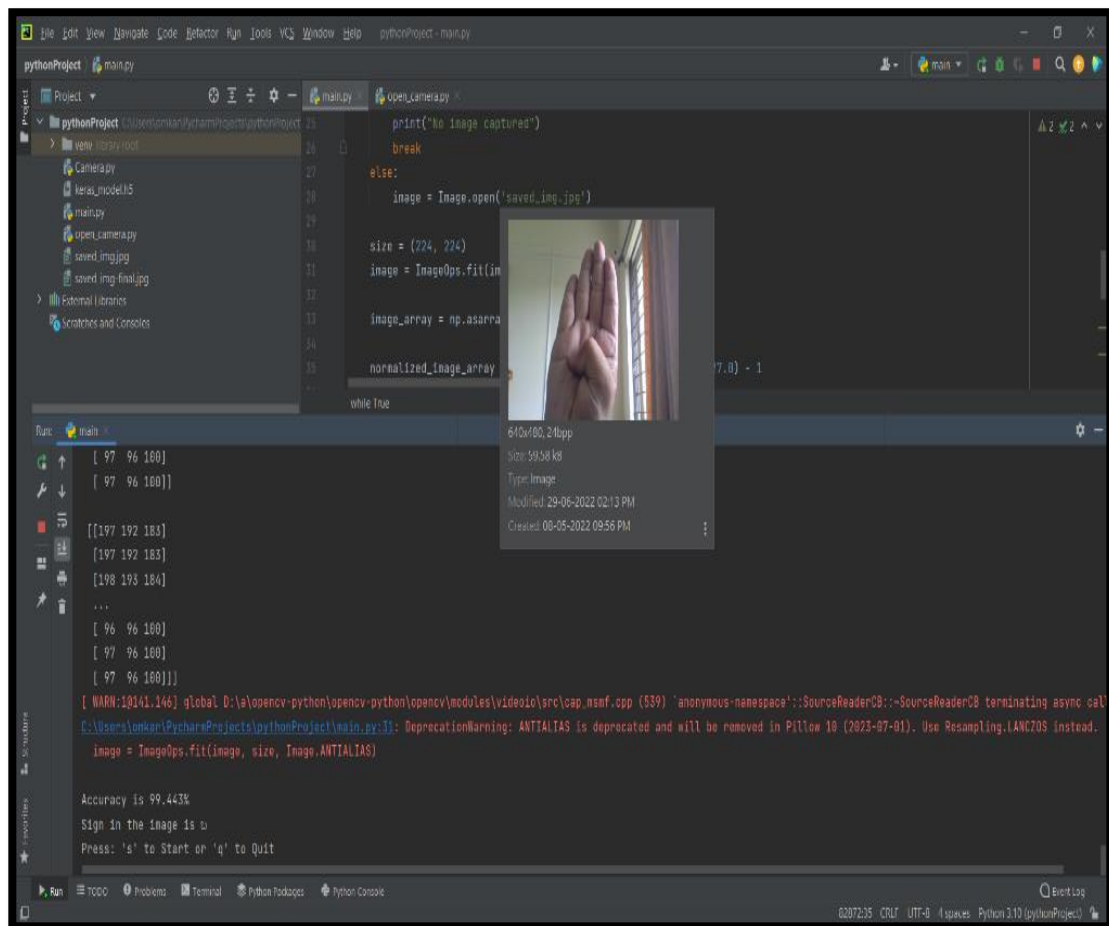


Fig 2. Implementation in PyCharm IDE

Camera Operations

We have used OpenCV Python library for camera operations in a Python program called “open_camera.py”. The program prompts the system to open the camera window named ‘capture’ and save the captured image in the specified path, the image is nothing but test image. For each and every time the captured image will be overwritten. OpenCV is a video and image processing library and it is used for image and video analysis, like facial detection, license plate reading, photo editing, advanced robotic vision, and many more a video can be read either by using the feed from a camera connected to a computer or by reading a video file. The first step towards reading a video file is to create a Video capture object. Its argument can be either the device index or the name of the video file to be read. The python program shows to press ‘s’ to capture the image and ‘q’ to quit. On the basis of command, we entered it will run the project and after capturing the image again we have to press ‘s’ to save the image and compare the test image with the trained images which is in the ‘keras.h5’ model trained using teachable machine.

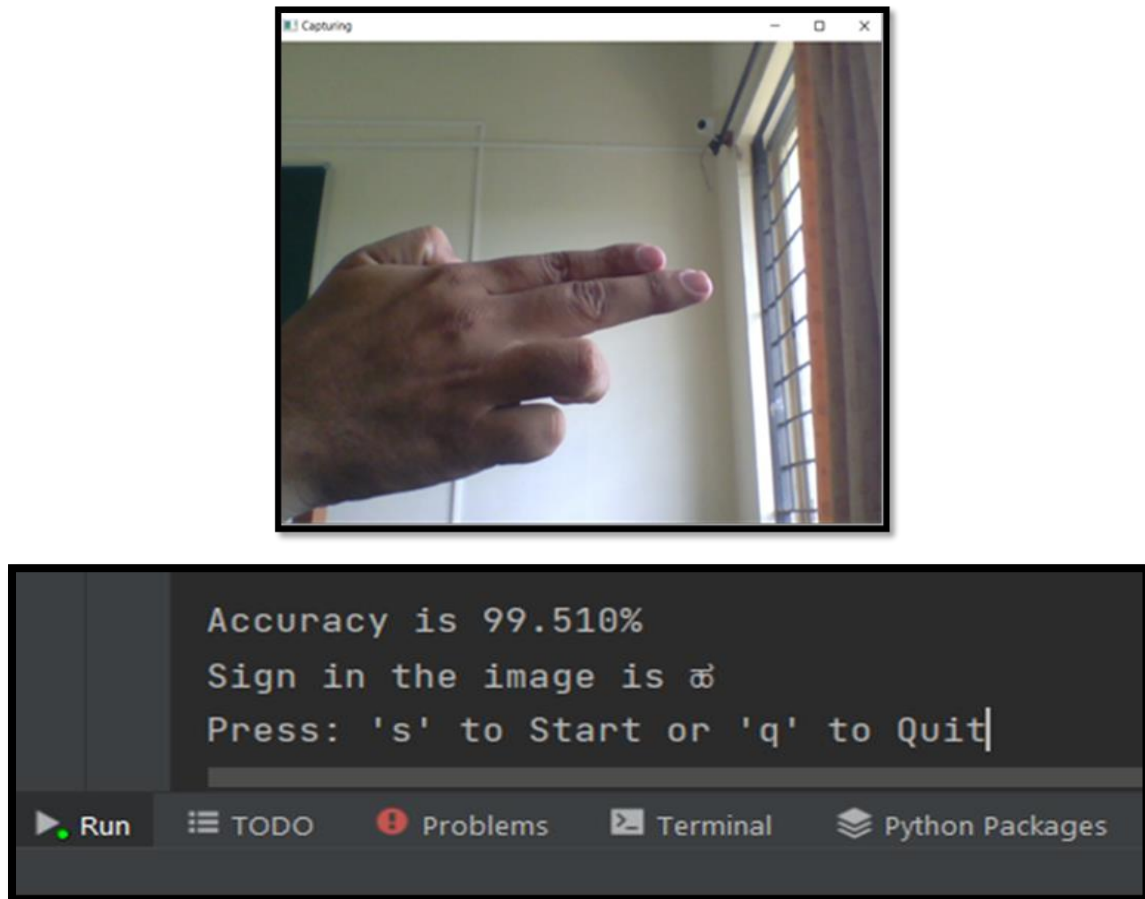


Fig 3. Sample image with output and accuracy

Accuracy

In machine learning, accuracy is one of the most important performance evaluation metrics for a classification model. The mathematical formula for calculating the accuracy of a machine learning model is $1 - (\text{Number of misclassified samples} / \text{Total number of samples})$. In this model we have tried to achieve maximum accuracy using own datasets. “Prediction” refers to the output of an algorithm after it has been trained on a historical dataset and applied to new data when forecasting the likelihood of a particular outcome. The algorithm will generate probable values for an unknown variable for each record in the new data, allowing the model builder to identify what that value will most likely be. Machine learning model predictions allow businesses to make highly accurate guesses as to the likely outcomes of a question based on historical data, which can be about all kinds of things – customer churn likelihood, possible fraudulent activity, and more. These provide the business with insights that result in tangible business value. For example, if a model predicts a customer is likely to churn, the business can target them with specific communications and outreach that will prevent the loss of that customer. We calculated accuracy by dividing the number of correct predictions (the corresponding diagonal in the matrix) by the total number of samples by using NumPy library. The final output is displayed with its accuracy.

IV. FUTURE SCOPE

The main scope of this model is to expand the range to frequently used Kannada words and sentences, which will take the model to the greater heights. The video processing techniques and more complex algorithms are required to expand this machine learning model to words and sentences. As the model’s aim is to fill the communication gap, in present days main way to communicate is through digital platforms i.e., social medias like WhatsApp, Twitter, etc., So as to fill the gap the improved version of this machine learning model can be included in any and all social media networks. Thus, helping mainly during video calls.



V. CONCLUSION

Maximum research work exists in the survey for Kannada Handwritten numeral, text recognition. However, there are very few works on Kannada sign language recognition, also there is no standard solution to recognize Kannada sign language with reasonable accuracy. In existing papers, different approaches have been used in recognition process, however each approach provides solution only for few sign language alphabet sets. This project trying to propose best possible method for recognition of sign language versions of Kannada vowels and Kannada consonants using Machine learning technique. There is a scope for improving the methodology by using better features, which will be explored to improve the recognition rate and test with other classifiers.

REFERENCES

- [1]. Algorithm to Identify Kannada Vowels using Minimum Features Extraction Method by K.S. Prasanna Kumar, Associate Professor, Department of Computer Science-MCA, Acharya Institute of Technology, Bangalore, India.
- [2]. Handwritten Assamese Character Recognition using Texture and Diagonal Orientation features with Artificial Neural Network by Rubul Kumar Bania, Assistant Professor and RuhUllah khan, Former PG Student, Department of Computer Applications, North-Eastern Hill University, Meghalaya, India.
- [3]. KANNADA-MNIST: a new handwritten digit's dataset for the Kannada language by Vinay Uday Prabhu, dig.minst@gmail.com
- [4]. New Methodology for Translation of Static Sign Symbol to Words in Kannada Language by Ramesh M Kagalkar Research scholar VTU Belgaum, Nagaraj H N Professor IITE Ahmadabad.
- [5]. Sign Language Translator Using Machine Learning by Vishwas S, Hemanth Gowda M, Vivek Chandra H N, Tannvi, Students, Department of Computer Science & Engineering, Vidyavardhaka College of Engineering, Mysuru.
- [6]. Deep Learning for Sign Language Recognition: Current Techniques, Benchmarks and Open Issues by Muhammad Al-Qurishi (Member, IEEE)
- [7]. A Deep Learning based Indian Sign Language Recognition System by Sruthi C. J and Lijiya A IEEE Members 12382, Saudi Arabia.
- [8]. A Survey on Sign Language Recognition Systems by Shruty M. Tomar (M Tech), Dr. Narendra M. Patel (Professor), Dr. Darshak G Thakore (Head)
- [9]. Sign Language Recognition Using Convolutional Networks by Lionel Pigou(B), Sander Dieleman, Pieter-Jan Kindermans and Benjamin Schrauwen ELIS, Ghent University, Ghent, Belgium
- [10]. Conversion of sign language to text and speech using machine learning techniques by Victoria A. Adewale Crawford University, Dr. Adejoke O. Olamiti Department of Neuroimaging Sciences, Centre for Clinical Brain Sciences, University of Edinburgh, Edinburgh UK.
- [11]. Lipi Gnani - A Versatile OCR for Documents in any Language Printed in Kannada Script by H. R. SHIVA KUMAR and A. G. RAMAKRISHNAN,
- [12]. Deep Learning Network Architecture based Kannada Handwritten Character Recognition by N. Shobha Rani, Subramani A C, Akshay Kumar P, Pushpa B Department of Computer Science, Amrita School of Arts & Sciences, Mysore.
- [13]. Handwritten Character Recognition of Kannada Language Using Convolutional Neural Networks and Transfer Learning by H Parikshith
- [14]. Handwritten Numeral Recognition of Kannada Script by S.V. Rajashekararadhya, P. Vanaja Ranjan Department of Electrical and Electronics Engineering CEG, Anna University, Chennai, India
- [15]. Sign Language Recognition by Anna Deza and Danial Hasan.
- [16]. Curvilinear tracing approach for recognition of Kannada sign language by Ramesh M. Kagalkar and Shyamrao V. Gumaste Visvesvaraya Technological University (VTU), Belgaum, Karnataka, India
- [17]. Gesture Recognition and Machine Learning Applied to Sign Language Translation by Luis A. Estrada Jiménez, Marco. E. Benalcázar and Nelson Sotomayor
- [18]. Sign Language Recognition and Translation: A Multidisciplinary Approach from the Field of Artificial Intelligence by Becky Sue Parton University of North Texas
- [19]. Study of vision-based hand gesture recognition using Indian sign language by Archana Ghotkar Pune Institute of Computer Technology
- [20]. Indian sign language interpreter using image processing and machine learning by Shubhendu Apoorv, Sudharshan Kumar Bhowmick and R Sakthi Prabha Department of ECE, Sathyabama Institute of Science and Technology, Chennai, India