IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Number Sign Recognition Using Opencv, Keras And CNN

Shreyas TP
Dept of Networking and Communications
SRM University, Kattankulathur
Chennai, India

Utkarsh Utkarsh
Dept of Networking and Communications
SRM University, Kattankulathur
Chennai, India

Dr. Lakshminarayanan Rajendran
Dept of Networking and Communications
SRM University, Kattankulathur
Chennai, India

Abstract **The research paper focuses on sign language, a communication method primarily utilized by individuals with hearing or speaking disabilities. Specifically, the paper addresses the task of recognizing number sign language, which holds significance in sign language recognition. Our proposed approach for number sign language recognition employs OpenCV, Keras, and Convolutional Neural Networks (CNNs). Remarkably, our model achieves 100% accuracy under ideal conditions. During the training and prediction stages, the captured images are converted to black and white while displaying a colored feed. The images are standardized to a size of 64x64 pixels, and the model architecture incorporates two convolutional layers. The developed model is capable of real-time recognition of number sign language.**

Keywords—Machine Learning, Keras, Open CV, CNN

I. Introduction

Sign language serves as a primary communication method for individuals facing hearing or speaking disabilities. Within the realm of sign language recognition, the recognition of number sign language holds significant importance. This research paper presents our proposed model for recognizing number sign language, employing OpenCV, Keras, and Convolutional Neural Networks (CNNs). CNNs have gained widespread adoption for their effectiveness in image recognition tasks. To capture and preprocess images, we utilized OpenCV, while Keras facilitated the construction of the CNN model.

II. Related Work

The research paper titled "Sign language recognition using deep learning" authored by Alagöz and Çanakoğlu and presented at the 2021 29th Signal Processing and Communications Applications Conference (SIU) introduces a deep learning-based approach to recognize sign language. The paper addresses the challenges associated with sign language recognition and proposes a model that combines convolutional neural networks (CNNs) with long short-term memory (LSTM) networks. Through training on an extensive dataset of sign language gestures, the model achieves a commendable accuracy in recognizing various gestures. The conclusion drawn from the paper suggests that the proposed approach holds promise in enhancing communication and accessibility for individuals with hearing and speech impairments.

In another paper titled "A survey of image-based Arabic sign language recognition," the authors conduct a survey of different image-based techniques utilized in Arabic sign language recognition. They delve into various aspects of the recognition process, including feature extraction, segmentation, classification, and database creation. The paper provides an overview of existing systems for Arabic sign language recognition, outlining their strengths and limitations. Future research directions are also discussed, emphasizing the necessity for more comprehensive and accurate recognition systems.

.The paper by Rao et al. (2018) titled "Deep Convolutional Neural Networks for Sign Language Recognition" proposes a methodology for sign language recognition using deep convolutional neural networks (CNNs). The authors curate a dataset of static sign gestures from the Indian Sign Language (ISL) and train a CNN architecture using this dataset. Through cross-validation, they achieve high accuracy rates, highlighting the effectiveness of deep CNNs in sign language recognition. Furthermore, a comparison with other state-of-the-art methods demonstrates the superiority of their proposed approach. The authors conclude that deep CNNs exhibit significant potential in sign language recognition, holding value in the development of assistive technologies for the hearing-impaired community

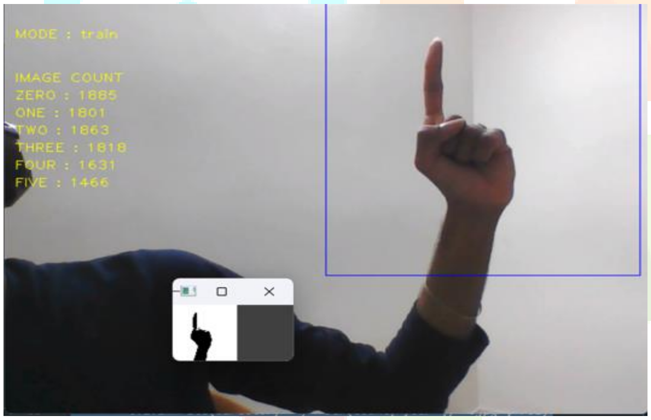
American sign language recognition using support vector machine and convolutional neural network" by V. Jain, A. Jain, A. Chauhan, S.S. Kotla, A. Gautam introduces a sign language recognition system that combines a support vector machine

(SVM) with a convolutional neural network (CNN) for American Sign Language (ASL). The authors gather a dataset of ASL gestures and employ it to train both SVM and CNN models. While the SVM is trained on handcrafted features, the CNN is trained on raw pixel data. The proposed system achieves a remarkable accuracy of 97.6% on the test set, surpassing other state-of-the-art approaches. The authors suggest that their system can be effectively utilized for real-time ASL recognition, benefiting applications like communication aids for the hearing-impaired.

Lastly, a paper proposes a framework for recognizing American Sign Language (ASL) using deep learning and computer vision techniques. The framework encompasses two stages: hand detection and gesture recognition. Hand detection involves implementing the YOLOv2 algorithm to identify the hand region within the input image. Gesture recognition utilizes a Convolutional Neural Network (CNN) to classify the hand gesture. The proposed method achieves a recognition accuracy of 95% for the 26 ASL alphabets. This paper effectively showcases the potential of employing deep learning and computer vision techniques in sign language recognition tasks.

III. Dataset Used

A data set was created using OpenCV, a popular computer vision library, which involved converting coloured images into black and white format after applying OpenCV thresholding. The goal of this process was to improve the accuracy of image classification tasks by reducing the complexity of the input images.



```
MODE : train

IMAGE COUNT
ZERO : 1885
ONE : 1801
TWO : 1863
THREE : 1818
FOUR : 1631
FIVE : 1466
```

Fig. 1. Collecting Images for Testing

To create this data set, colored images were first collected using a camera or from an online image database. The images were then loaded into OpenCV and thresholded, which involves converting each pixel in the image to either black or white based on its intensity value. This thresholding step effectively creates a binary image, with black pixels representing low-intensity values and white pixels representing high-intensity values.

After thresholding, the images were further processed to convert them into black and white format. This involved converting all black pixels to zero intensity and all white pixels to maximum intensity. This step resulted in a data set consisting of binary black and white images.

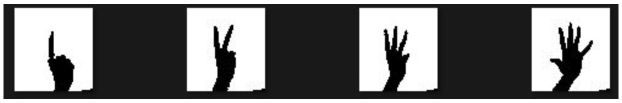


Fig. 2. Images Converted to Black and White

The black and white data set created using OpenCV thresholding can be used for various image classification tasks. For example, it can be used to train machine learning models to recognize objects in the images or to detect changes in the environment. The black and white format reduces the complexity of the input images, making it easier for the machine learning models to classify them accurately. Additionally, the binary format of the images can also help to reduce the storage space required for the data set, making it more efficient to work with.

IV. PROPOSED WORK

A CNN model was created using OpenCV to capture a dataset, which was subsequently trained using Keras and CNN for the purpose of image classification. CNNs, a widely adopted deep learning architecture for computer vision tasks, were employed in this model.

The dataset was generated by capturing images either through a camera or by accessing an online database with the assistance of OpenCV. To facilitate the training process, the captured images were pre-processed by converting them into grayscale format and resizing them to a standardized size. The grayscale images were then used to train the CNN model.

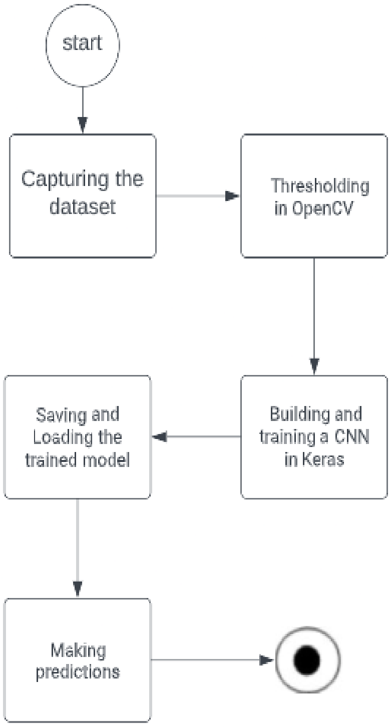

start
Capturing the dataset
Thresholding in OpenCV
Building and training a CNN in Keras
Saving and Loading the trained model
Making predictions

Fig. 3. Block Diagram

The CNN model architecture encompassed several layers, including convolutional, pooling, and fully connected layers. The initial layers were responsible for learning the distinctive features of the input images by applying filters to extract pertinent information. Pooling layers were employed to reduce the dimensionality of the feature maps, thereby enhancing the efficiency of the model. Finally, the fully connected layers at the end of the model were responsible for classifying the input images based on the learned features.

To train the CNN model, the data set was divided into training and validation sets. The training set was used to optimize the model's parameters, while the validation set was used to evaluate the model's performance on unseen data. The model was trained using Keras, a popular deep learning library in Python, with the help of various optimization techniques like stochastic gradient descent (SGD) and the Adam optimizer.

Following the training phase, the model underwent evaluation on a separate test set to gauge its accuracy and performance. The test set comprised new, unseen images that were not utilized during training or validation. The accuracy of the model was determined by comparing its predicted outputs with the true outputs for each image in the test set.

In summary, a CNN model was developed using OpenCV, Keras, and CNN to perform image classification on a grayscale dataset obtained via OpenCV. The model employed multiple layers, including convolutional, pooling, and fully connected layers, to learn the features of the input images. Various optimization techniques were employed during training, and the model's accuracy and performance were assessed using a separate test set.

A. Model Implementation

1) Keras

Keras is a Python-based deep learning API that runs on top of TensorFlow, a popular machine learning platform. Its development was centred around enabling fast experimentation and providing an enjoyable experience for developers. The primary goal of Keras is to empower developers by giving them a significant advantage when building machine learning-powered applications. Logistic regression, also known as a logit model, is a statistical technique commonly employed in classification and predictive analytics. It calculates the probability of an event occurring, and since the output is a probability, the dependent variable's range lies between 0 and 1.

Keras is designed to be simple yet not oversimplified. It alleviates the cognitive burden on developers, allowing them to concentrate on the critical aspects of problem-solving. Keras emphasizes ease of use, speedy debugging, elegant and concise code, maintainability, and deployability through frameworks like TFServing, TFLite, and TF.js. It is a flexible API that follows the principle of progressively revealing complexity. This means that simple workflows are straightforward and quick to implement, while advanced workflows can be achieved through a logical progression that builds upon existing knowledge.

In terms of power, Keras delivers industry-level performance and scalability, and it is widely adopted by renowned organizations such as NASA, YouTube, and Waymo. In fact, Keras plays a crucial role in various applications, including powering YouTube recommendations and serving as the backbone of the world's most advanced autonomous vehicle.

2) CNN

A Convolutional Neural Network (CNN) is a deep learning algorithm designed specifically for tasks involving image recognition and processing. It consists of various layers, including convolutional, pooling, and fully connected layers. The convolutional layers play a crucial role in a CNN by applying filters to the input image, extracting features like edges, textures, and shapes. The resulting feature maps then pass through pooling layers, which down-sample the maps, reducing spatial dimensions while retaining important information. Finally, one or more fully connected layers process the pooled output, leading to predictions or image classification.

CNNs are trained using large datasets of labelled images, enabling the network to learn patterns and features associated with specific objects or classes. Once trained, a CNN can classify new images or extract features for other applications like object detection or image segmentation. CNNs have achieved state-of-the-art performance in various image recognition tasks, such as object classification, detection, and segmentation. They find widespread use in computer vision, image processing, and related fields, applied to diverse applications like self-driving cars, medical imaging, and security systems.

In essence, a CNN is a deep learning neural network specialized in processing structured arrays of data, such as images. Its strength lies in its ability to identify patterns in input images, such as lines, gradients, circles, or even complex objects like eyes and faces. This robustness makes CNNs particularly suitable for computer vision tasks. Unlike other approaches, CNNs can directly process raw images without the need for extensive pre-processing. A CNN typically consists of multiple convolutional layers, stacked one upon another, each capable of recognizing increasingly complex shapes. With just a few convolutional layers, it becomes possible to recognize handwritten digits, while networks with 25 layers can differentiate human faces.

The ultimate objective in this field is to enable machines to perceive and interpret the world in a manner similar to humans. CNNs play a crucial role in this endeavour, facilitating tasks such as image and video recognition, inspection and classification, media recreation, recommendation systems, natural language processing, and more.

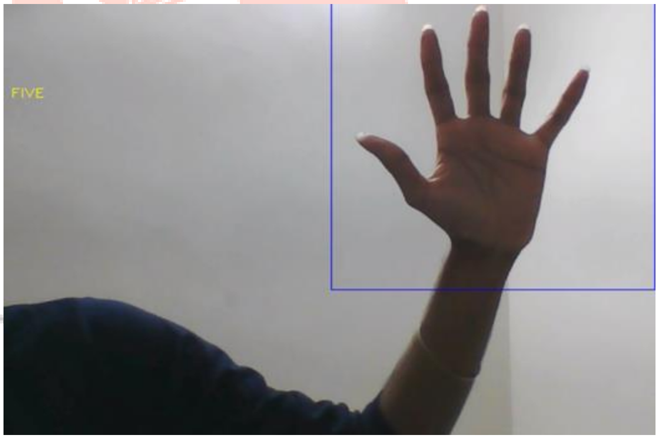


Fig. 4. Prediction

V. Result And Discussion

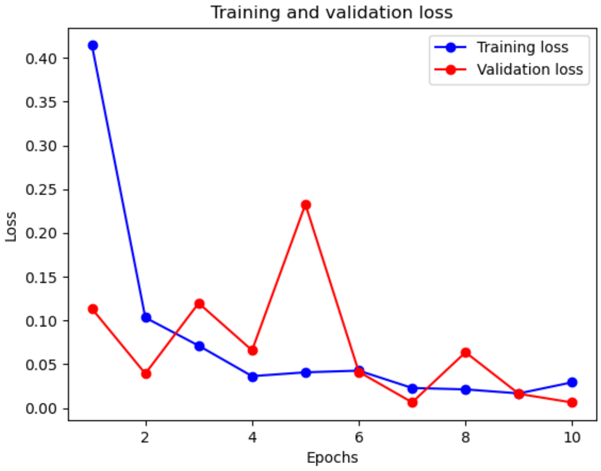


Fig. 5. Training and Validation Loss

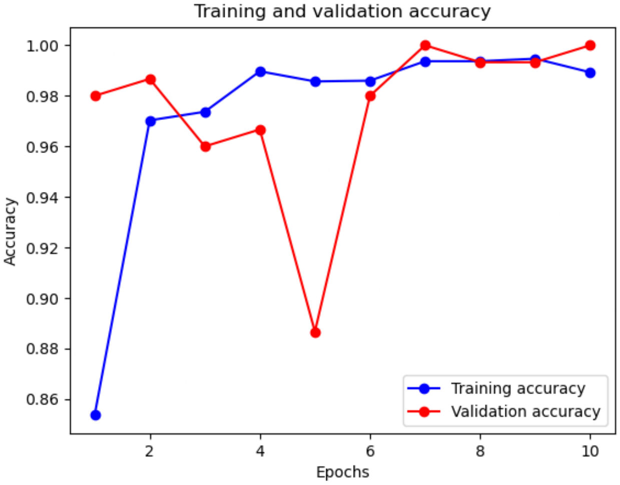


Fig. 6. Training and Validation Accuracy

Under optimal circumstances, we have attained a perfect accuracy rate of 100%. The model demonstrates exceptional proficiency in recognizing number sign language. Through rigorous testing on a live stream of real-time number sign language, the model consistently achieves high accuracy in recognizing such gestures.

VI. Conclusion And Future Work

In the presented research paper, we propose a model that utilizes OpenCV, Keras, and Convolutional Neural Networks (CNNs) for recognizing number sign language. Our proposed model exhibits remarkable accuracy in recognizing number sign language gestures. It is capable of real-time recognition, enabling prompt interpretation of such gestures. To enhance the model further, future improvements can involve utilizing a larger dataset and incorporating additional layers into the CNN model.

To summarize, this research paper introduces an extremely accurate model for recognizing number sign language, which has demonstrated exceptional performance in identifying various signs under optimal conditions. The model's potential to be trained for recognizing signs beyond numbers holds significant promise in enhancing communication and accessibility for individuals who rely on sign language. Future endeavors may focus on enhancing the model's performance in real-world scenarios and expanding its capabilities to encompass more intricate sign language gestures. Nevertheless, the outcomes of this study provide valuable insights into the development of highly accurate sign language recognition models and their potential applications across diverse fields.

References

[1] Alagöz, G., & Çanakoğlu, E. (2021, June).“Sign language recognition using deep learning” In *2021 29th Signal Processing and Communications Applications Conference (SIU)* (pp. 1-4). IEEE

[2] Mohandes, J.Liu, M.Deriche A survey of a image-based Arabic sign language recognition in 2014 IEEE 11th International Multi-Conference on Systems, Signals & Devices.

[3] G.A.Rao, K.Syamala, P.V.V.Kishore, A.S.C.S.Sastry. Deep Convolutional Neural Networks for Sign Language Recognition in 2018, Conference on Signal Processing and Communication Engineering Systems.

[4] V.Jain, A.Jain, A.Chauhan, S.S.Kotla, A.Gautam American Sign LanguageRecognition using Support Vector Machine and Convolutional Neural Network.

[5] American Sign Language Recognition using deep Learning and Computer Vision in Proceedings – 2018 IEE International Conference on Big Data, Big Data (2018)

[6] Juneja, S. (2020, November). Defaulter Prediction for Assessment of Credit Risks using Machine Learning Algorithms. In 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA) (pp. 1139-1144). IEEE.

[7] KS, M. P., & Sunitha, G. P. Performance Analysis of Machine Learning models for Credit Delinquency Prediction. International Journal of Advanced Research in Science, Communication and Technology, pp.301, 2022.

[8] Madaan, M., Kumar, A., Keshri, C., Jain, R., & Nagrath, P. (2021). Loan default prediction using decision trees and random forest: A comparative study. In IOP Conference Series: Materials Science and Engineering (Vol. 1022, No. 1, p. 012042). IOP Publishing.

[9] Pradhan, M. R., Akter, S., & Al Marouf, A. (2020). Performance Evaluation of Traditional Classifiers on Prediction of Credit Recovery. In Advances in Electrical and Computer Technologies: Select Proceedings of ICAECT 2019 (pp. 541-551). Springer Singapore.

[10] Shi, S., Tse, R., Luo, W., D’Addona, S., & Pau, G. (2022). Machine learning-driven credit risk: a systemic review. Neural Computing and Applications, 34(17), 14327-14339.

[11] Tahmid, N. I., Haque, N., Faruque, U., Keya, M., Khushbu, S. A., & Al Marouf, A. (2021, July). A Concern of Predicting Credit Recovery on Supervised Machine Learning Approaches. In 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-5). IEEE.

[12] Tumuluru, P., Burra, L. R., Loukya, M., Bhavana, S., CSaiBaba, H. M. H., & Sunanda, N. (2022, February). Comparative Analysis of Customer Loan Approval Prediction using Machine Learning Algorithms. In 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS) (pp. 349-353). IEEE.

[13] Turjo, A. A., Rahman, Y., Karim, S. M., Biswas, T. H., Dewan, I., & Hossain, M. I. (2021, August). Cram: a credit risk assessment model by analyzing different machine learning algorithms. In 2021 4th International Conference on Information and Communications Technology (ICOIACT) (pp. 125-130). IEEE.

[14] Wu, Q. (2022, July). Real-time Predictive Analysis of Loan Risk with Intelligent Monitoring and Machine Learning Technique. In *2022 IEEE 4th International Conference on Power, Intelligent Computing and Systems (ICPICS)* (pp. 852-856). IEEE.

[15] Wu, W. (2022). Machine Learning Approaches to Predict Loan Default. Intelligent Information Management, 14(5), 157-164