American Sign Language (ASL) Recognition Using Convolutional Neural Network (CNN)

Gurleen Kaur[1], *Harshit Saini[2] and Mohd. Jubair Alam[3]

[1,2,3]Department of Computer Science and Engineering, Chandigarh University, Mohali , India

[1]gurleenkr0106@gmail.com, [2]sainiharshit554@gmail.com, [3]ansarijubair661@gmail.com

Abstract. Sign language is a natural language that emerged to fulfill the communication requirements of the Deaf community. It uses the visual-gestural modality, meaning that meaning is mostly communicated through the use of hands, facial expressions, and upper body motions. Sign language has changed and grown over time, just like all languages, to meet the ever-changing demands of its users. Current technological advancements offer fresh chances to enhance communication between Deaf individuals and non-sign language users, particularly in the domains of computer vision and artificial intelligence (AI). Language barrier reduction and improved clarity and efficiency in communication can be facilitated by these technologies, which allow for more accurate and rapid recognition of sign language motions. In order to develop models that can comprehend and recognize sign language, this research project looks into the application of AI and computer vision. To empower individuals with disabilities, the intention is to provide them with the means to express their ideas and opinions to a wider audience.

Keywords: ASL Recognition, Convolutional Neural Network, Machine Learning, Deep Learning, Artificial Intelligence, Language.

1 Introduction

Sign language is a special language primarily for the people with hearing and speaking disabilities. Sign language uses body language, facial expressions, and hand gestures for communication. American Sign Language (ASL) is one of the sign languages that is commonly used in the USA and some areas of Canada also. Growing interest in ASL has been shown in creating systems that can recognize ASL gestures in real time with the latest advancements and developments in deep learning and computer vision. These kinds of recognition systems can be used as assistive technology, in education, and in communication happening virtually. The detection system for sign language can reduce the communication gap between abled people and specially abled people.

N. Vyas et al. (eds.), *Proceedings of the International Conference on Advancements in Computing Technologies and Artificial Intelligence (COMPUTATIA 2025)*, Advances in Intelligent Systems Research 189,
https://doi.org/10.2991/978-94-6463-700-7_15

2 Literature Review

[1] The authors' study focuses on creating a deep learning-based system for recognizing ASL from video. They used convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to process both temporal and spatial elements of pictures. The authors' technique employs an RNN to analyze geographical features and a CNN to extract video frames in order to capture temporal correlations between frames. This hybrid approach, which includes spatial information for hand forms and motions, allows for reliable gesture detection on the ASL dataset. This work also investigates various issues in gesture identification, such as hand forms, occlusions, lighting fluctuations, and so on. [2] They presented a framework for detecting Indian Sign Language (ISL) gestures. Along with it, it emphasizes the particular difficulties ISL presents in contrast to other sign languages such as ASL or German Sign Language. ISL is less arbitrary than the earlier stated languages, since it incorporates the use of both hands in most signals, and these signs are generally mixed regularly, which makes it difficult to segment and identify. For a better rate of success in sign language recognition, the authors propose the use of a fingertip locating approach supported by Principal Component Analysis (PCA) for dimensionality reduction and feature extraction. This study also discovered that combining PCA with certain parameters such as the number of fingertips and distances from the hand centroid increases anti-noise capabilities when sign information is performed randomly. This research focuses on challenges such as hand overlap, motion blur, and different backgrounds in order to efficiently create a recognition system for the ILS.

[3] CNNs were used to investigate the detection of hand gestures in ASL. The authors evaluated different CNN architectures to determine the best model for deployment on mobile devices; models like as LeNet-5, AlexNet, VGG16, and MobileNet v2 were included in the study. The authors also discovered that deeper architectures, such as VGG16 and MobileNet v2, have the ability to learn feature representations and achieve higher accuracy since they have more layers and use complex methods, such as batch normalization and dropout. However, because these models are also more sophisticated, they are not particularly practical to employ in industries that demand real-time reactions, such as in mobile phones or embedded systems. [4] The authors created a new standard for recognizing ASL with the SLRNet-8 CNN model. This study aimed to improve the recognition of ASL numerals and alphabets by using four open-source datasets containing photographs of signs. Compared to traditional approaches, the suggested CNN model improves recognition accuracy to near-perfect levels on the datasets studied. To improve model resilience, the authors used rotation, scaling, shifting data augmentation techniques, cross-dataset testing, and 10-fold cross-validation. The study concluded that a proper CNN design and data augmentation are essential for achieving high output efficiency rates for ASL gesture detection. [5] The authors discussed the use of deep learning technology to detect illnesses in tomato plant leaves. The CNN model. The purpose of this study was to increase ASL recognition accuracy by analyzing four publicly available datasets containing images of ASL numerals and alphabets. When the proposed model was compared to existing methodologies, it was discovered that the CNN model significantly improves recognition accuracy on the used datasets,

achieving nearly 100% accuracy. To boost the model's durability, the scientists additionally employed data augmentation techniques such as rotation, scaling, and shifting. The model's generalizability was validated across multiple datasets using cross-dataset testing and 10-fold cross-validation. The study concluded that to achieve high-efficiency rates for output in ASL gesture recognition, a well-structured CNN architecture and use of data augmentation are key. [6] The authors discussed employing deep learning technology to identify leaf illnesses in tomato plants. Based on this research, the authors focused their discussion on the following benefits of using deep learning technologies, such as OpenCV, CNNs, and image processing in the identification of plant ailments, as well as the recommendations for how to prevent subsequent infection. The suggested model datasets included approximately 15100 photos of healthy and diseased tomato leaves organized into ten classifications. The plant illness identification accuracy percentages ranged from 91.25% to 93.9%. This work also demonstrates that OpenCV can be used for image preprocessing and CNN for detection.

In conclusion, the authors identified the requirements for the successful application of deep learning in agriculture and discussed the possibility of using such models in web applications, embedded systems, mobile devices, and application-specific platforms such as drones. [7] The authors proposed using CNN to recognize motions in ASL hand gestures. This work investigates the following deep learning architectures to discover the ideal CNN model for real-time applications that require great computational efficiency at the expense of slightly reduced accuracy: LeNet-5, AlexNet, Vgg16, and MobileNet v2. This study also considers the downsides of using mobile and embedded devices to harness deeper networks required to capture the necessary information for proper gesture detection. This study examines the trade-off between computational efficiency and accuracy for real-time applications.

This study also takes into account the limitations of mobile and embedded devices while highlighting the significance of using deeper networks to capture complex features required for accurate gesture recognition. Using data augmentation methods, the authors can improve the recognizers' generalizability and, with their selected CNN architecture, attain a testing accuracy of 87.5%. This work contributes to the literature by offering insights about the factors that practitioners face when designing deployment strategies based on the conflicting factors of efficiency and model geometry, particularly within the resource-restricted setting. [8] The authors presented by comparing multiple techniques for identifying ASL, in which videos are then snipped in data for usage in the system, including Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and CNN. The authors concluded that CNNs produce better results than other models and are most effective for real-time applications. The proposed CNN model worked with a stochastic gradient descent optimizer and cross-entropy loss on a multilayered neural network. Its architecture is based on grayscale images. The key innovations of the model are heavy use of subsampling and attempting to learn sparse representations. Through interpreting the motions of ASL, this technology seeks to bring into reality efficient communication between signers and those who do not. [9] The authors of this research paper designed a CNN for recognizing human hand gestures. The authors put great emphasis on the model's resilience to changes in hand position and light circumstances. Before feeding the photos into the CNN, the model calibrates hand position and orientation to neutral postures and uses a Gaussian Mixture Model (GMM) for skin color segmentation. Their test findings show a high recognition accuracy of roughly 94.96%, confirming the applicability of their strategy for a variety of subjects and hand gestures. [10] Researchers conducted a thorough investigation of the deep learning model SLR in continuous sign recognition. To improve recognition accuracy, the paper emphasized the importance of adopting multimodal techniques in addition to hands; it should be able to distinguish body and facial emotions as well. In their research, the authors identified numerous challenges with the hand recognition model, including the range of hand morphologies and motion profiles, as well as the need for effective models with real-time processing and multilingual support. The survey underlines the importance of developing hybrid models that can manage the temporal dynamics of continuous sign language by combining CNNs with Recurrent Neural Networks (RNNs). [11] The authors suggest a CNN-based approach for detecting gestures in Indian Sign Language using mobile device selfie video data. Their dataset includes 200 frequently used ISL phrases that were collected from various perspectives. By employing stochastic pooling, it outperformed more traditional pooling strategies such as max and mean pooling. [12] Authors introduced an innovative LSTM-based model featuring a built-in Leap Motion sensor for Continuous Sign Language Recognition (CSLR). The model resets the LSTM's memory during sign transitions, and it incorporates convolutional layers for effective spatial feature extraction, demonstrating high performance in continuous input streams. This approach illustrated the applicability of deep learning for modeling sequential data, achieving an average accuracy of 72.3% for signed sentences and 89.5% for individual sign words. [13] Used an identification algorithm known as YOLO (You Only Look Once) to identify signs in real time from video frames. The authors developed Indonesian Sign Language (BISINDO), in which the system produced very good results in terms of precision and recall when dealing with static images. However, when dealing with dynamic video data, the precision and recall were low because the gesture was identified at a distance and in different lighting conditions. [14] The authors presented a research report on picture detection and recognition using CNNs. The authors stated that CNNs can be more efficiently applied to big datasets of 2D images. Aside from that, they discussed how to build picture detection and recognition using CNN. For the study, the two authors created a model that they evaluated on datasets like MNIST and CIFAR-10, yielding strategic success rates of 99.6% and 80.01%, respectively. The study also concluded that CNN is extremely useful in the domains of image recognition and detection, which are important in the development of applications such as hand gesture recognition. [15] The authors of this research paper focused their research on real-time hand gesture recognition using the deep learning framework OpenCV. The authors focused on fingertip detection using the convex hull algorithm and AdaBoost optimizer. The authors also addressed the challenges in the real-time detection of hand gestures. A vision-based system was created that avoids using additional equipment such as specialized gloves, making it more accessible. The author's system achieved an accuracy of 92%. [16] Authors created a system recognizing Indian license plates from live stream videos using a CNN. The research was focused on license plate recognition and how CNN recognizes complex patterns in different real-life conditions faced by the license plate recognition system. The authors compared several approaches, such as k-NN and Open-ALPR, and out of all, CNNs performed best in terms of accuracy and adaptability under different conditions such as varying lighting, skewed images, and different font styles. This research showcased the robustness of CNNs in complex visual pattern recognition.

3 Proposed Model

In this section, it outlines the methodology used to develop an ASL recognition system using CNNs. To view the pictorial representation of the proposed model, Fig. 1 can be referred to. [7] The process involves several key stages:

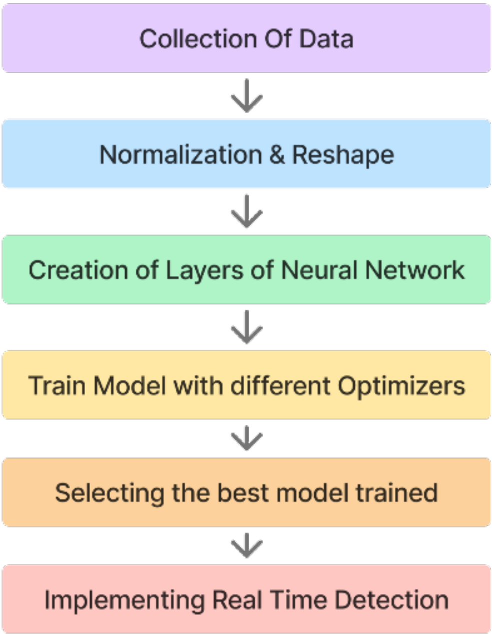

Fig. 1. Flow chart of Proposed Model

3.1 Collection Of Data

The first stage for developing a high accuracy ASL recognition system is by collection of large and diverse dataset of ASL hand signs. [10] You can get the dataset by 2 methods:

- **Custom Dataset**: Make your own dataset to deliver superior results for a particular use case.
- **General Dataset**: If creating the dataset takes too much time for you, you can obtain a general dataset from websites like Kaggle that have been created by experts. We

opted for the latter for our dataset since building a custom dataset can be time-consuming, while adopting a general dataset from Kaggle will significantly cut down on the amount of time needed for model training without significantly affecting the model's accuracy. [4]

The dataset is displayed as a CSV file that contains the pixel values for images with a resolution of 28 by 28 pixels. There are 27,455 photos in our training dataset and 7,172 images in our test dataset. [13] The dataset's images were captured in a variety of settings, taking into account varying lighting conditions. To view the pictorial chart of signs in ASL, Fig. 2 can be referred to.

Fig. 2. Images of the types of Alphabets Signs taken in Consideration

3.2 Normalization and Reshape

Normalization [3] is used to standardize pixel values, usually scaling them to a range of 0 to 1. This guarantees that the input data is consistent and speeds up the model training process. This standardization of all the pixel values in the range from [0,1] is important for a CNN because:

- **Faster Convergence**: Normalizing input data guarantees that all features have the same range, allowing optimization methods such as gradient descent to converge faster.
- **Mitigating Exploding/Vanishing Gradients**: Normalization eliminates severe gradients during backpropagation, lowering the likelihood of exploding or disappearing gradient issues.
- **Improved Generalization**: By scaling inputs evenly, the model avoids overfitting to individual characteristics, resulting in improved performance on unknown data.

- **Equalized Learning Rate**: Normalization guarantees that all input characteristics contribute equally and prevents features with higher values from dominating the learning process.
- **Reduced Sensitivity to Initial Weight**: Normalization stabilizes the training process by lowering reliance on starting weight values. Consistency Across Inputs By standardizing variances between samples (such as illumination and scale), the model may avoid bias from uneven input scales and instead concentrate on learning pertinent patterns.

3.3 Train Model with different Optimizers

A typical CNN architecture consists of several layers for feature extraction and classification. The core components of the CNN model include:

- **Convolutional Layers**: The convolutional layers [11], which are used for spatial features and texture extraction, employ filters to identify patterns in pictures, including edges, curves, and forms associated with hand motions.
- **Pooling Layers**: The pooling layers in CNN perform pooling operations that involve sliding a two-dimensional filter over each channel of the feature map. Mainly, a pool is used for reducing the dimensionality of the feature maps and summarizing the features present in a region of the feature map generated by a convolution layer.
- **Full-linked layers** are a neural network in which each neuron applies a linear transformation to the input vector through a weight matrix. This layer plays a crucial role in learning high-level representations, as it captures the complex relationships between the features and the output classes.
- **Activation Function**: To add non-linearity, ReLU (Rectified Linear Unit) activation is frequently applied after each convolutional layer. The last layer for multi-class classification uses SoftMax activation.

Optimizers are essential for training CNNs since they update model weights depending on loss function gradients. Multiple kinds of optimizers may be utilized for our use case, and we constructed models using two of them: RMSProp and Adam. When evaluating the model, it was discovered that both optimizers, Adam and RMSProp, produced identical model correctness. So, we chose Adam because it combines the benefits of RMSprop (adaptive learning rates) and momentum (smoothing updates), resulting in smoother and faster convergence. It is also one of the most commonly utilized optimizers.

3.4 Implementing Real Time Detection

Real-time prediction models are critical for applications that demand quick analysis and feedback, such as gesture recognition, autonomous systems, and security surveillance. In the context of ASL recognition, real-time prediction models enable continuous identification of hand motions, allowing for rapid communication. These models process live data streams, usually from video feeds, and utilize machine learning methods like

CNNs to anticipate outcomes based on input. Technologies and libraries used for real-time detection are:

- **OpenCV**: OpenCV [5] was chosen because it makes the process of recording, processing, and presenting video frames easier and is capable of handling real-time video feeds. For image-based predictions, it makes integration with other frameworks like TensorFlow simple.
- **TensorFlow/Keras**: TensorFlow [5] was chosen because of its extensive deep-learning capabilities. Keras (a TensorFlow API) makes it easier to import and use pre-trained CNN models.
- **NumPy**: used for effective numerical operations, including normalization, reshaping, and preprocessing of images.
- **Webcam**: Live video input is captured using a common camera, which makes the device easily attainable and simple to configure.

The real-time gesture recognition system follows a defined control flow, which includes recording video frames, preprocessing the picture, predicting motions using the trained model, and presenting the results on screen.

- **Video Capture**: OpenCV records a live video stream from the camera.
- **Region of Interest (ROI):** A specific area of the video frame is selected to isolate the hand movements.
- **Preprocessing:** The chosen ROI is converted to greyscale, scaled, normalized, and reshaped to fit the CNN model's input size.
- **Model Prediction:** The preprocessed frame is submitted to the CNN model, which predicts the ASL gesture.
- **Display Results:** The predicted gesture is shown on the live video feed.
- **Exit Condition:** The system will operate until the user presses the 'x' key to exit.

To view the Control flow of real-time detection of ASL Recognition Fig. 3 can be referred.

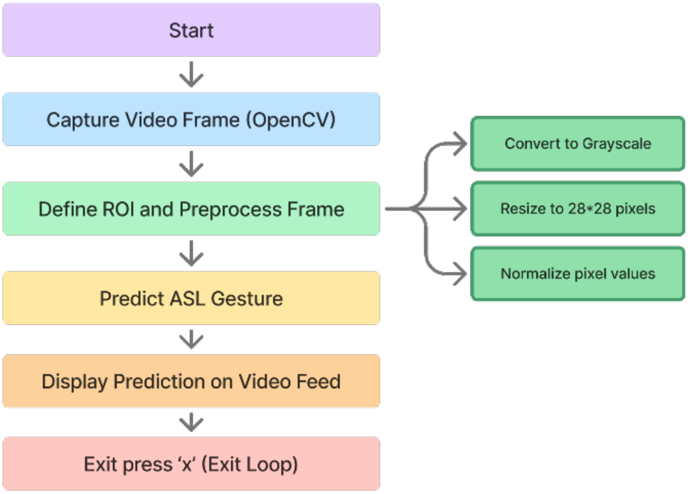

Fig. 3. Control flow of the real-time detection of ASL Recognition

4 Results

A real-time system for recognizing ASL using CNNs has shown encouraging results in terms of correctly detecting hand motions in an ongoing video stream. The system processed live frames smoothly, gave almost immediate predictions, and tracked pictures effectively and accurately. This mix of prediction accuracy and real-time performance demonstrates the practical applicability of deep learning models for improving nonverbal communication, particularly for people with hearing or speech difficulties. For the result obtained, accuracy, loss, and working implementation, Table 1, Fig. 4, and Fig. 5 can be referred to.

Table 1. Evaluation Metrics

Evaluation Metrics	Value
Accuracy	94.80%
Precision	0.95
Recall 0.94	0.94
F1-Score	0.94

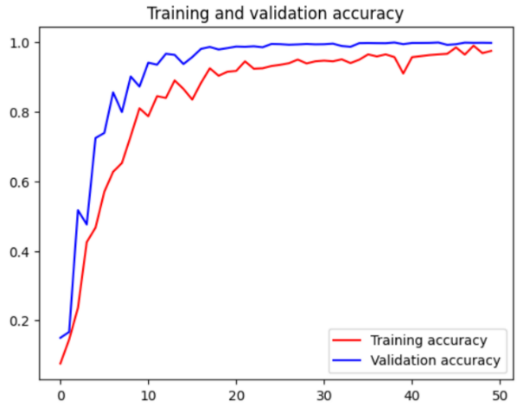

Fig. 4. Accuracy Graph of the CNN Model for ASL Recognition

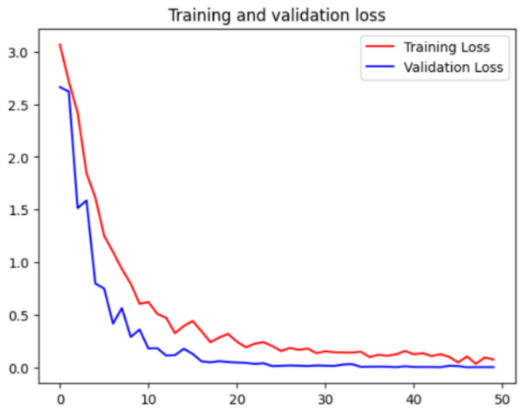

Fig. 5. Loss Graph of the CNN Model for ASL Recognition

5 Future Scope

During our study, we successfully created and deployed an ASL recognition model. However, we discovered many restrictions that hampered the model's overall performance. First, the prediction accuracy was often extremely sensitive to little variations in hand motions, resulting in inconsistent outcomes. Furthermore, the present system is only intended to recognize single-hand motions from a single user, limiting its potential to accept multi-hand gestures or manage input from several users at the same time. Furthermore, the model struggled to follow motions in dynamic or crowded backgrounds, limiting its practical application in complicated contexts.

The system still has a lot of potential for growth in the future, even with current challenges—especially if these obstacles are removed. Improvements in robustness and effectiveness might significantly expand the use of this technology, particularly for those with special needs who utilize sign language, leading to innovations such as:

- **Communication Application for Non-Sign Language Users**: Create a program aimed at people who do not know sign language. Using this software, people will be able to successfully communicate across language barriers by tracking and interpreting each other's hand motions via the camera.
- **Language Recognition Implementation with Smart Devices:** Devices such as smart watches, AR (Augmented Reality), and VR (Virtual Reality) can assist in the instant translation of sign language into speech or text, such as smart spectacles or AR headsets. This invention aims to improve inclusion by facilitating effective communication between sign language users and those who do not.
- **AI-Powered Virtual Assistants**: A bot powered by artificial intelligence that can recognize and respond to sign language commands. This type of accessibility support facilitates and simplifies interactions between members of the Deaf community and technologies such as websites, systems, and operating systems.
- **Educational Integration:** Incorporating sign language recognition into classroom settings so that teachers can use sign language to educate or communicate directly with hearing-impaired students. This integration will improve academic access for specially abled children in classrooms.

References

1. K. Bantupalli and Y. Xie, "Proceedings of the 2018 IEEE International Conference on Big Data," in *IEEE Big Data 2018*, Seattle, pp. 1–13, 2018.
2. D. Deora and N. Bajaj, "Emerging Technology Trends in Electronics, Communication, and Networking," in *IEEE International Conference 2012*, pp. 1–13, 2012.
3. S. Chavan, X. Yu, and J. Saniie, "Convolutional Neural Network Hand Gesture Recognition for American Sign Language," in *IEEE International Conference on Electro Information Technology*, pp. 188–192, 2021, doi: 10.1109/EIT51626.2021.9491897.
4. Md. M. Rahman, Md. S. Islam, Md. H. Rahman, R. Sassi, M. Rivolta, and Aktaruzzaman, "Sustainable Technologies for Industry 4.0," in *IEEE International Conference 2019*, Dhaka, pp. 1–13, 2019.
5. T. Verma, S. Khattar, H. Joshi, H. Saini, R. Kumar, and V. K. Choudhary, "Leaf Disease Detection Using OpenCV and Deep Learning," in *CCICT 2024*, pp. 15–22, 2024, doi: 10.1109/ccict62777.2024.00013.
6. R. Poojary and A. Pai, "Electrical and Computing Technologies and Applications," in *IEEE International Conference 2019*, pp. 1–13, 2019.
7. U. Hari Priya, S. Krishna Prasad, M. M. Jacob, R. R. Krishna, and P. R. Vinod, "American Sign Language Recognition Using CNN," *Int. J. of Research in Engineering, Science and Management*, vol. 3, no. 7, pp. 1–13, 2020.

8. H.-I. Lin, M.-H. Hsu, and W.-K. Chen, "Automation Science and Engineering," in *IEEE International Conference 2014*, pp. 1–13, 2014.
9. R. Rastgoo, K. Kiani, and S. Escalera, "Sign Language Recognition: A Deep Survey," *Expert Systems with Applications*, vol. 164, pp. 1–13, 2021, doi: 10.1016/j.eswa.2020.113794.
10. G. A. Rao, K. Syamala, P. V. V. Kishore, and A. S. C. S. Sastry, "Deep Convolutional Neural Networks for Sign Language Recognition," in *IEEE International Conference 2018*, pp. 1–13, 2018.
11. A. Mittal, P. Kumar, P. P. Roy, R. Balasubramanian, and B. B. Chaudhuri, "A Modified LSTM Model for Continuous Sign Language Recognition Using Leap Motion," *IEEE Sensors Journal*, vol. 19, no. 16, pp. 7056–7063, 2019, doi: 10.1109/JSEN.2019.2909837.
12. S. Daniels, N. Suciati, and C. Fathichah, "Indonesian Sign Language Recognition Using YOLO Method," in *IOP Conference Series: Materials Science and Engineering*, vol. 1077, no. 1, pp. 1–13, 2021, doi: 10.1088/1757-899x/1077/1/012029.
13. R. Chauhan, K. Ghanshala, and R. C. Joshi, "Convolutional Neural Network (CNN) for Image Detection and Recognition," in *Secure Cyber Computing and Communication 2018*, Jalandhar, 2018, pp. 1–13.
14. R. Gurav and P. Kadbe, "Real-time Finger Tracking and Contour Detection for Gesture Recognition Using OpenCV," in *IEEE International Conference on Industrial Instrumentation and Control 2015*, 2015, pp. 1–13.
15. S. Prabhu, S. Kalambur, and D. Sitaram, "Recognition of Indian License Plate Number from Live Stream Videos," in *IEEE International Conference on Advances in Computing, Communications and Informatics 2017*, 2017, pp. 1–13.
16. H. Yang and X. Han, "Face Recognition Attendance System Based on Real-time Processing," *IEEE Access*, vol. 8, pp. 159143–159150, 2020, doi: 10.1109/ACCESS.2020.3007205.