International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Hand Sign Language Translation System Using Machine Learning

Karampudi Dishank Jagadeeshnaidu

School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India

Abstract: ***Sign Language is a form of communication primarily used by individuals who are deaf and mute, employing hand movements and gestures. This study presents a proposed solution for recognizing hand gestures through the utilization of a Deep Learning Algorithm called Convolution Neural Network (CNN). The CNN is responsible for processing the images and making predictions about the gestures. The research focuses on the recognition of five hand gestures from the American Sign Language. The suggested system consists of various components, including pre - processing and feature extraction, model training and testing, and the conversion of sign language into text. To enhance the accuracy of recognition, different CNN architectures like VGG19 were employed, and pre - processing techniques such as greyscale and resizing were designed and evaluated using our dataset.***

Keywords: Sign Language, CNN, VGG19, Deep Learning, Hand Gestures

1. Introduction

The module receives an image as input and identifies the corresponding sign gesture depicted in the image or video. The process consists of four main steps:

Image Pre - processing: The initial step involves preparing the input image, which can be obtained directly or captured from a video. Various pre - processing techniques are applied to the captured image to enhance its suitability for further analysis.

CNN Model Input: The pre - processed image is then fed into a Convolution Neural Network (CNN) model. This model has been trained on a dataset of sign language images to recognize and classify different sign gestures.

Text Prediction: The CNN model analyzes the input image and generates a predicted text that represents the recognized sign gesture. This text is displayed as the output of the process.

To ensure accurate predictions, direct input of unprocessed images to the CNN model can lead to inaccuracies due to difficulties in background cancellation. To address this issue, separate image processing techniques are applied to the images prior to feeding them into the CNN model. These techniques help improve accuracy and include resizing the images to a standardized 45x45 size. The specific image processing techniques used with our dataset are detailed below.

In addition to the CNN model, a specific CNN architecture, VGG19, is utilized for training the model. Among the various architectures, VGG19 demonstrates superior accuracy compared to the basic CNN model.

2. Framework, Architecture or Module for the Proposed System (with explanation)

In the past few decades, Deep Learning has proved to be a very powerful tool because of its ability to handle large amounts of data. The interest to use hidden layers has surpassed traditional techniques, especially in pattern recognition. One of the most popular deep neural networks is Convolutional Neural Networks.

The AI system, known as AlexNet (named after its primary creator, Alex Krizhevsky), achieved a remarkable 85 percent accuracy and emerged as the winner of the 2012 ImageNet computer vision contest. In comparison, the runner - up managed a modest 74 percent accuracy on the same test.

At the core of AlexNet was Convolutional Neural Networks (CNNs), a specialized type of neural network that mimics human vision to some extent. Over time, CNNs have become an integral component of numerous Computer Vision applications, making them a fundamental aspect of any computer vision course. Let's delve into how CNNs function. CNNs were initially developed and utilized in the 1980s, primarily for recognizing handwritten digits. They found application in the postal sector for reading zip codes, pin codes, and similar tasks. It's important to note that deep learning models, including CNNs, require substantial amounts of training data and significant computing resources. This limitation posed a significant challenge for CNNs during that era, which restricted their usage to the postal sector, preventing them from making significant inroads into the realm of machine learning.

3. Limitations

Despite the power and resource complexity of CNNs, they provide in - depth results. At the root of it all, it is just recognizing patterns and details that are so minute and inconspicuous that it goes unnoticed to the human eye. But when it comes to understanding the contents of an image it fails.

The ImageNet Large Scale Visual Recognition Challenge is an annual competition in the field of computer vision. Participants compete in two tasks: object localization, where the goal is to detect objects from 200 different classes within an image, and image classification, where images need to be classified into one of 1000 predefined categories.

Licensed Under Creative Commons Attribution CC BY

VGG - 19 is a deep convolutional neural network proposed by Karen Simonyan and Andrew Zisserman from the Visual Geometry Group Lab at Oxford University. The model was introduced in their 2014 paper titled "VERY DEEP Convolutional Networks for Large - SCALE IMAGE RECOGNITION. " In the 2014 ILSVRC challenge, VGG - 19 achieved remarkable success by securing the first and second places in both object localization and image classification tasks.

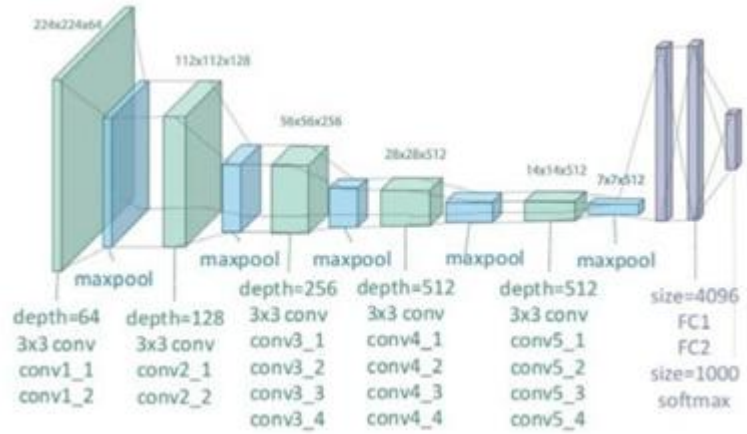

Figure 1: Illustration of the network architecture of VGG - 19 model

4. Architecture

The network takes an input image with dimensions of (224, 224, 3). The initial two layers consist of 64 channels with a filter size of 3x3 and same padding. Subsequently, there is a max pooling layer with a stride of (2, 2). Following this, there are two convolution layers with a filter size of 256 and a size of (3, 3), along with another max pooling layer with the same stride as the previous layer.

Continuing further, there are two convolution layers with a filter size of (3, 3) and 256 filters. This is succeeded by two sets, each containing three convolution layers and a max pooling layer.

Each of these layers has 512 filters with a size of (3, 3) and uses the same padding technique. The image is then passed through a stack of two convolution layers. Unlike AlexNet (which uses 11x11 filters) and ZF - Net (which uses 7x7 filters), these layers employ 3x3 filters. Additionally, 1x1 pixel filters are utilized in certain layers to adjust the number of input channels. Each convolution layer is followed by a 1 - pixel padding (same padding) to preserve the spatial features of the image.

Following the stack of convolution and max pooling layers, a feature map of size (7, 7, 512) is obtained. This output is then flattened into a feature vector of size (1, 25088). Subsequently, there are three fully connected layers: the first takes the feature vector as input and outputs a (1, 4096) vector, the second also outputs a (1, 4096) vector, and the third outputs a vector with 1000 channels corresponding to the 1000 classes in the ILSVRC challenge. The output of the third fully connected layer is passed through a SoftMax layer to normalize the classification vector. The top - 5 categories are evaluated based on the output of the classification vector.

ReLU activation function is used in all hidden layers. ReLU is chosen due to its computational efficiency, enabling faster learning, and its ability to mitigate the vanishing gradient problem.

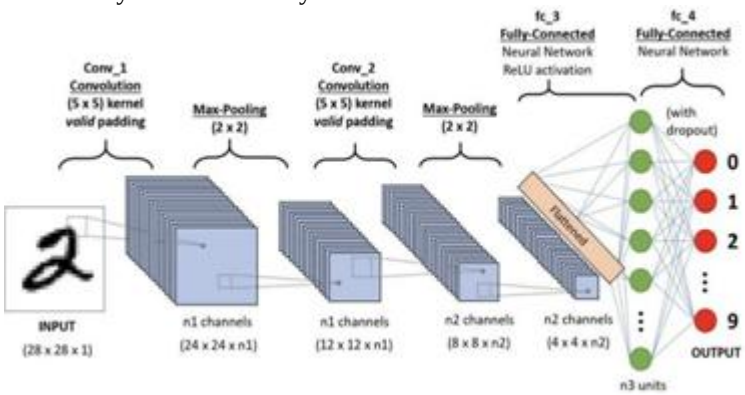

Figure 2: Background of CNN

5. Proposed System Model

Communication involves the transmission of information, but research indicates that approximately percent of the population in India faces challenges in communication. These individuals are unable to speak or hear, which greatly restricts their ability to interact. To express themselves, they

Licensed Under Creative Commons Attribution CC BY

rely on Sign Language, a unique form of communication. In India, Indian Sign Language is utilized and standardized across the nation. It involves the use of hand gestures made with a single hand or both hands. Grammar rules, such as tense forms and articles like 'a', 'an', and 'the', are not considered. The sentence structure of Indian Sign Language follows a Subject - Object - Verb (SOV) pattern, which differs significantly from the Subject - Verb - Object (SVO) structure of the English language.

Functional Requirements: Functional Requirements:

1) Gesture Recognition: The system must accurately recognize and classify hand gestures using a Deep Learning Algorithm, specifically Convolutional Neural Network (CNN), applied to image processing, with a focus on Indian Sign Language.
2) Sign Language Recognition: The system should excel in recognizing and categorizing single and double - handed gestures specific to Indian Sign Language, achieving high accuracy rates.
3) Pre - processing and Feature Extraction: The system should include modules for pre - processing and extracting features from input images or video frames. Techniques like grayscale conversion and resizing should be employed to enhance accuracy in gesture recognition.
4) Training and Testing of Model: The system must facilitate training and testing of the CNN model using a dataset of hand gesture images. It should ensure effective prediction and classification of sign gestures.
5) Sign - to - Text Conversion: The system should provide a module to convert recognized sign gestures into corresponding text, enabling effective communication between sign language users and non - sign language users.
6) CNN Architecture and Comparison: The system should explore and compare different CNN architectures, such as VGG19, to determine the most accurate and efficient model for hand gesture recognition.
7) Image and Video Input Support: The system should support both image and video inputs for hand gesture recognition, offering flexibility in usage scenarios.
8) Background Cancellation: The system should address the challenge of background interference in hand gesture recognition by employing suitable image processing techniques to isolate hand gestures from the background.
9) Image Size Standardization: The system should resize input images to a standardized size, such as 45x45, prior to processing, ensuring consistency and improving recognition accuracy.
10) Computational Requirements: The system should specify the necessary hardware requirements, such as an Intel I5 processor, 4GB RAM, and 40GB hard disk, to effectively run the implemented system.

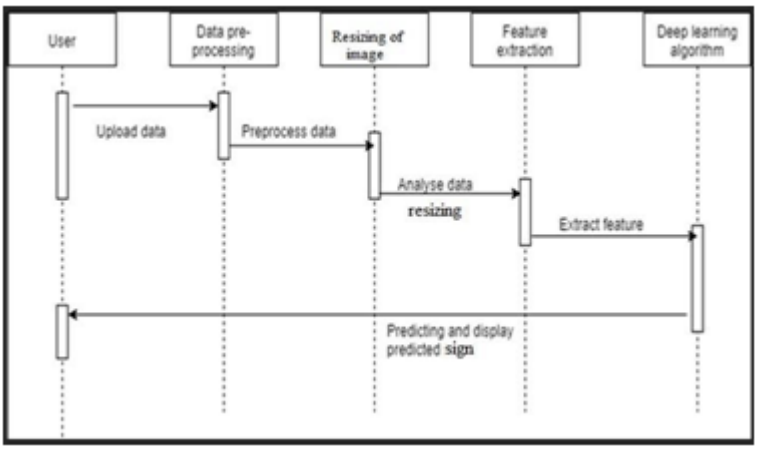

Figure 3: Sequential diagram

5. Experiments

Dataset

Data serves as the foundation for any machine learning project. The implementation phase of a project involves complex tasks such as data collection, selection, preprocessing, and transformation. Each of these phases can be broken down into several steps.

Data Collection: During this phase, a data analyst takes charge and leads the way in implementing machine learning. Their role is to identify ways and sources to collect relevant and comprehensive data, interpret it, and analyze the results using statistical techniques. The type of data depends on the specific prediction objective. The amount of data required varies for each machine learning problem, and the selection of attributes depends on their predictive value. It is recommended to collect as much data as possible, as it is challenging to determine in advance which portion will yield the most accurate results. Therefore, collecting and storing all types of data, including internal and open, structured and unstructured, is crucial. Various tools can be utilized for data collection, such as web analytic tools like Mixpanel, Hotjar, CrazyEgg, and widely known tools like Google Analytics. Additionally, publicly available datasets from platforms like Kaggle, Github, and AWS can complement internal data.

Data Preprocessing: The purpose of data preprocessing is to convert raw data into a format suitable for machine learning. Structured and clean data enables data scientists to achieve more precise results with applied machine learning models. This technique involves data formatting, cleaning, and sampling.

Licensed Under Creative Commons Attribution CC BY

Data Formatting: Data formatting becomes crucial when data is acquired from different sources and recorded by different individuals. Standardizing record formats is the initial task, ensuring that variables representing attributes are recorded consistently. This applies to variables such as product titles, prices, date formats, and addresses. Data consistency is also important for numeric range attributes.

Data Cleaning: Data cleaning involves procedures to remove noise and resolve inconsistencies in the data. Data scientists use techniques like imputation to fill in missing values, such as substituting missing values with attribute means. Outliers, which are observations that significantly deviate from the rest of the distribution, are detected and either corrected or removed if they indicate erroneous data. This stage also includes removing incomplete and irrelevant data objects. In some cases, sensitive attributes may need to be anonymized or excluded, particularly when working with healthcare or banking data.
Image Capture: The initial step in sign recognition involves capturing hand gestures using a camera. Proper interfacing with a high - definition web camera is critical for this method.

Data Augmentation: To overcome overfitting in the training stage of Convolutional Neural Networks (CNNs), data augmentation techniques are employed. By expanding the dataset through augmentation, the model can learn more irrelevant patterns, avoiding overfitting and achieving higher performance. Techniques such as rotation transformations, horizontal and vertical flips, and intensity disturbance (e. g., brightness, sharpness, contrast disturbances) are used for data augmentation.

Data Sampling: When dealing with large datasets, data sampling can be applied to reduce computational complexity. Data sampling involves selecting a smaller but representative subset of data for building and running models, resulting in faster processing without sacrificing accuracy. Image Preprocessing: Image processing can be categorized as analog image processing and digital image processing. Digital image processing, being a subfield of digital signal processing, offers advantages over analog processing. It allows for a wider range of algorithms to be applied, aiming to improve image data by suppressing unwanted distortions and enhancing important features for AI - Computer Vision models.

Image Reading and Resizing: The image dataset is read and stored in variables, and functions are created to display images. Resizing images is performed to establish a consistent size for all images, facilitating AI algorithm processing.

Data Splitting: A machine learning dataset should be divided into three subsets: training, test, and validation sets. The training set is used to train the model and determine its optimal parameters. The test set is used to evaluate the trained model's ability to generalize to new unseen data, avoiding overfitting by using different subsets for training and testing.

Model Performance
During this phase, the data scientist engages in training multiple models to determine the one that provides the most accurate predictions.

Model Training: At this stage, the limited number of images are used to train the model. The fast. ai library offers various architectures that facilitate transfer learning, making it convenient to create a convolutional neural network (CNN) model. Pre - trained models are utilized, which are suitable for a wide range of applications and datasets. Specifically, the ResNet architecture is employed due to its combination of speed and accuracy. The number in "resnet18" represents the neural network's layer count. A metric, such as error rate, is passed to evaluate the model's prediction quality using the validation set from the data loader. The fine - tuning process is similar to the fit () method in other machine learning libraries. To train the model, the desired number of epochs is specified, indicating the number of training iterations for each image.

Application of Deep Learning Modules for Object Detection:
CNN Classifier: In this project, object detection is accomplished using the CNN (Convolutional Neural Network) technique. The system is trained and tested with images of objects to determine if a person's face is covered or not. CNN is a widely used neural network type for image recognition and classification, relying on supervised learning. The CNN classifier consists of four layers: Convolutional, Pooling, Rectified Linear Unit (ReLU), and Fully Connected layers.

a) Convolutional layer: This layer extracts features from the input image by convolving with neurons, producing a feature map that serves as the input for the subsequent convolutional layer.
b) Pooling layer: The pooling layer reduces the dimensionality of the feature map while preserving important features. Typically, this layer is placed between two convolutional layers.
c) ReLU layer: ReLU (Rectified Linear Unit) is a non - linear operation that replaces negative values in the feature map with zeros. It is performed element - wise.
d) Fully Connected layer: The Fully Connected layer ensures that each filter in the previous layer is connected to each filter in the next layer. This layer is responsible for classifying the input image into various classes based on the training dataset.

The model construction phase depends on the chosen machine learning algorithm, such as Convolutional Neural Networks in this project. After constructing the model, the next step is model training, where the model is trained using the training data and their expected outputs. Once the model is trained, model testing is conducted using a separate set of data that the model has not encountered before to verify its true accuracy. After completing the model training phase, the saved model can be deployed in real - world scenarios, leading to the model evaluation phase.

6. Conclusion and Future work

The project is a simple demonstration of how CNN can be

Licensed Under Creative Commons Attribution CC BY

used to solve computer vision problems with an extremely high degree of accuracy. A finger spelling sign language translator is obtained The project can be extended to other sign languages by building the corresponding dataset and training the CNN. Sign languages are spoken more in context rather than as finger spelling languages, thus, the project is able to solve a subset of the Sign Language translation problem. The main objective has been achieved, that is, the need for an interpreter has been eliminated. There are a few finer points that need to be considered when we are running the project. The sizes needs to be monitored so that we don't get distorted grayscales in the frames. If this issue is encountered, we need to either reset the histogram or look for places with suitable lighting conditions. This project can be enhanced in a few ways in the future, it could be built as a web or a mobile application for the users to conveniently access the project, also, the existing project only works for ASL, it can be extended to work for other native sign languages with enough dataset and training.

References

[1] CNN - RNN: A Unified Framework for Multi - label Image Classification by Jiang Wang Yi Yang Junhua Mao Zhiheng Huang Chang Huang Wei Xu Baidu Research University of California at Los Angles Facebook Speech Horizon Robotics.
[2] Research on Computer - Vision based Object detection and Classification by Juan WuBo PengZhenxiang HuangJietao Xie.
[3] Study Of Object Detection Based On Faster R - CNN by BIN LIU, Wencang ZHAO and Qiaoqiao SUN.
[4] A Comparative evaluation of the GPU vs. the CPU for Parallelization of Evaluation Algorithms through multiple independent runs by Anna Syberfeldt and Tom Ekblom.
[5] Deep convolutional neural networks for brain image analysis on magnetic resonance imaging: a review by Jose Bernal, Kaisar Kushibar, Daniel S. Asfaw, Sergi Valverde, Arnau Oliver, Robert Mart´ı, Xavier Llad.
[6] Ray: A Distributed Framework For Emerging AI Applications by Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I. Jordan, Ion Stoica.
[7] Refining Faster - RCNN for Accurate Object Detection" by Myung - Cheol Roh; Ju - young Lee.
[8] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection.2005.
[9] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1867–1874, 2014.
[10] Nikhil R Pal, Chien - Yao Chuang, Li - Wei Ko, Chih - Feng Chao, Tzyy - Ping Jung, Sheng - Fu Liang, and Chin - Teng Lin. Eeg - based subjectand session - independent drowsiness detection: an unsupervised approach. EURASIP Journal on Advances in Signal Processing, 2008: 192, 2008.
[11] Bhargava Reddy, Ye - Hoon Kim, Sojung Yun, Chanwon Seo, and Junik Jang. Real - time driver drowsiness detection for embedded system using model compression of deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 121–128, 2017.
[12] Tereza Soukupova and Jan Cech. Eye blink detection using facial landmarks. In 21st Computer Vision Winter Workshop, Rimske Toplice, Slovenia, 2016.

Licensed Under Creative Commons Attribution CC BY