INDIAN SIGN LANGUAGE RECOGNITION SYSTEM TO HELP DEAF AND MUTE PEOPLE

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering / Technology degree in Computer Science and Engineering (Specialisation of degree)

By

Pitambara Awadhesh (38110406)

Reema Rose Toppo (38110458)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600119

MAY - 2022

ABSTRACT

Everything has changed due to the COVID19 pandemic. We went from offline to internet mode in no time. Some people found it easy to adjust to this way of life, but many others with disabilities never did and still find it difficult to explain their ideas in online meetings. To solve this problem, we have proposed a Convolutional Neural Network (CNN) based model for Indian sign language recognition. In our proposed method, a mute or deaf person can interact with the camera integrated into a computer and use gestures that will be recognized and converted to text for others to understand. For this, we have created a sample dataset and pre-processed it using a label binarizer. Afterward, feature extraction was done using two models, first for the palm region and the other for the fingers. Later on, this dataset was fed into a custom-made CNN model whose learning parameters were provided as 0.001 learning rate, 128 batch-size, and 10 epochs. The model performed well with 93% of accuracy while recognizing a hand gesture. Hence, our proposed model can be integrated into other online meeting sites for the target audience to use.

CONTENTS

CHAPTER 1 - INTRODUCTION

1.1 Basics

The goal of this project was to build a neural network able to classify which word of the Indian Sign Language (ISL) is being signed, given an image of a signing hand. This project is a first step towards building a possible sign language translator, which can take communications in sign language and translate them into written and oral language. Such a translator would greatly lower the barrier for many deaf and mute individuals to be able to better communicate with others in day-to-day interactions.

This goal is further motivated by the isolation that is felt within the deaf community. Loneliness and depression exist at higher rates among the deaf population, especially when they are immersed in a hearing world. Large barriers that profoundly affect life quality stem from the communication disconnect between the deaf and the hearing. Some examples are information deprivation, limitation of social connections, and difficulty integrating in society.

Most research implementations for this task have used depth maps generated by the depth camera and high-resolution images. The objective of this project was to see if neural networks are able to classify signed ISL letters using an input video sequence of a person taken with a personal device such as a laptop webcam. This is in alignment with the motivation as this would make a future implementation of a real-time ISL-to-oral/written language translator practical in everyday situations.

1.2 Python3

Python is an interpreted, high, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and

large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

1.3 Analysis and Visualization of Data

(a) NumPy

NumPy is a python library used for working with arrays. It also has functions for working in the domain of linear algebra, Fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open-source project and you can use it freely. NumPy stands for Numerical Python.

Why NumPy?

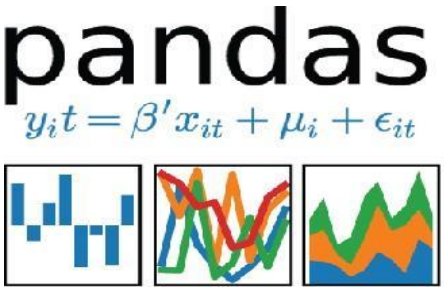
In Python, we have lists that serve the purpose of arrays, but they are slow to process. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.

Figure 1: NumPy

The array object in NumPy is called ndarray, in
provides a lot of supporting functions that make working with ndarray very easy. Arrays are very frequently used in data science, where speed and resources are very important.

(b) Pandas

Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the DataFrame. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables.



	BandName	WavelengthMax	WavelengthMin
0	CoastalAerosol	450	430
1	Blue	510	450
2	Green	590	530
3	Red	670	640
4	NearInfrared	880	850
5	ShortWaveInfrared_1	1650	1570
6	ShortWaveInfrared_2	2290	2110
7	Cirrus	1380	1360

Figure 2: Pandas: DataFrames

DataFrame object for data manipulation with integrated indexing.

- Tools for reading and writing data between in-memory data structures and different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of data sets.
- Label-based slicing, fancy indexing, and subsetting of large data sets.
- Data structure column insertion and deletion.
- Group by engine allowing split-apply-combine operations on data sets.
- Data set merging and joining.
- Hierarchical axis indexing to work with high-dimensional data in a lower-dimensional data structure.
- Time series-functionality: Date range generation and frequency conversion, moving window statistics, moving window linear regressions, date shifting, and lagging.
- Provides data filtration.

(c) MatPlotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib produces publication-quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, web application servers, and various graphical user interface toolkits.

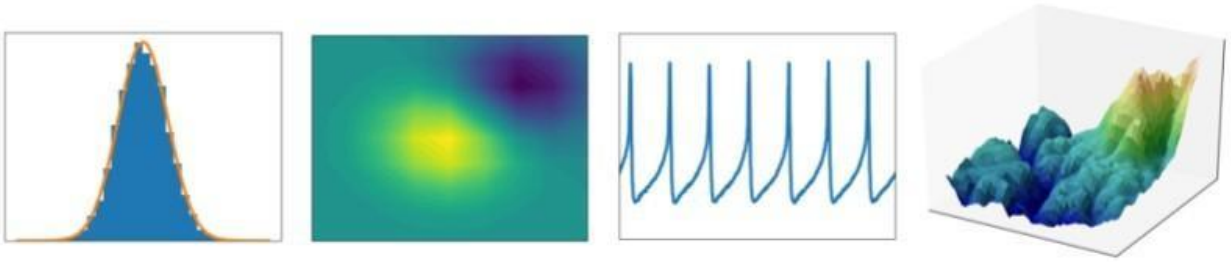


Figure 3: Plots using Matplotlib

• Convenient views onto the overall structure of complex data-set

- High-level abstractions for structuring multi-plot grids that let you easily build complex visualizations
- Concise control over matplotlib figure styling with several built-in themes
- Tools for choosing color palettes that faithfully reveal patterns in your data

1.4 Deep Learning

Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain—albeit far from matching its ability—allowing it to “learn” from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.

Deep learning drives much artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention.

Deep neural networks consist of multiple layers of interconnected nodes, each building upon the previous layer to refine and optimize the prediction or categorization. This progression of computations through the network is called forward propagation. The input and output layers of a deep neural network are called *visible* layers. The input layer is where the deep learning model ingests the data for processing, and the output layer is where the final prediction or classification is made.

To state the layer in a clear manner:

1. Input layer – The input layer has input features in a dataset that is known to us.
2. Hidden Layer – Hidden layer, just like we need to train the brain through hidden neurons.
3. Output layer – value that we want to classify.

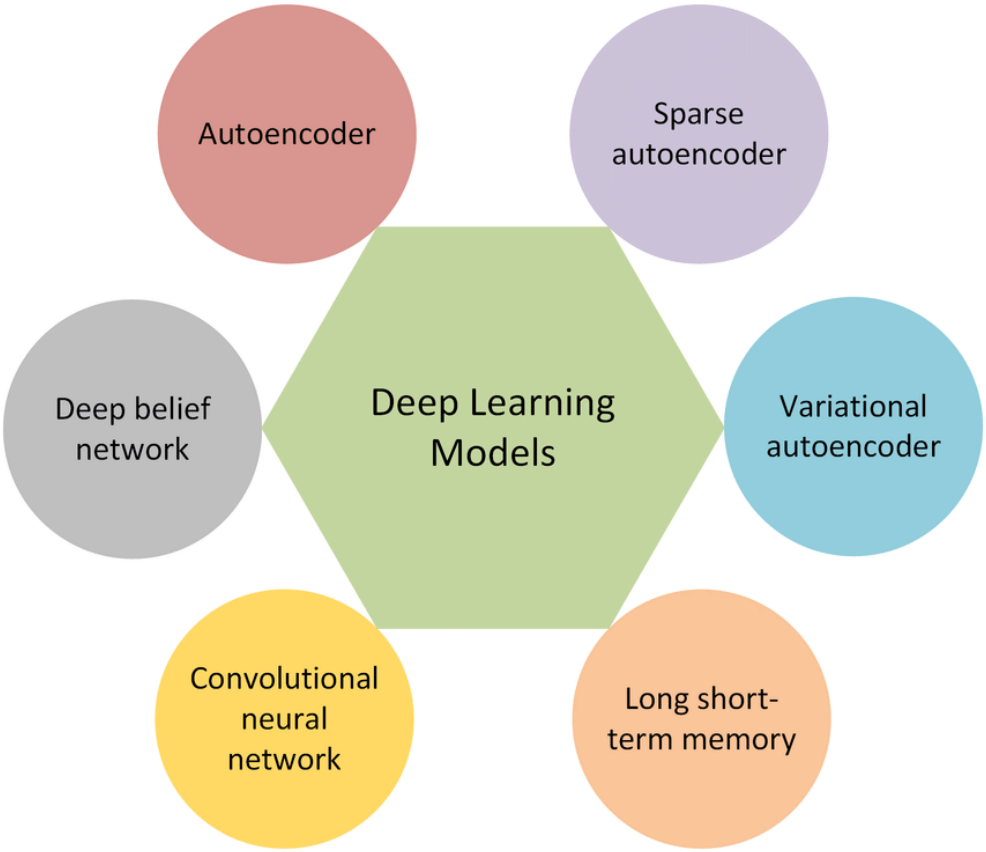


Fig 5: Types of Deep learning algorithms

There are many types of deep learning algorithms developed over the years but there are a few algorithms that are frequently used:

1. Artificial Neural Network:

An artificial Neural Network is the component of a computing system designed in such a way that the human brain analyses and makes a decision. Ann is the building block of deep learning and solves problems that seem impossible or very difficult to humans.

The first layer usually extracts basic features such as horizontal or diagonal edges. This output is passed on to the next layer which detects more complex features such as corners or combinational edges. As we move deeper into the network it can identify even more complex features such as objects, faces, etc.

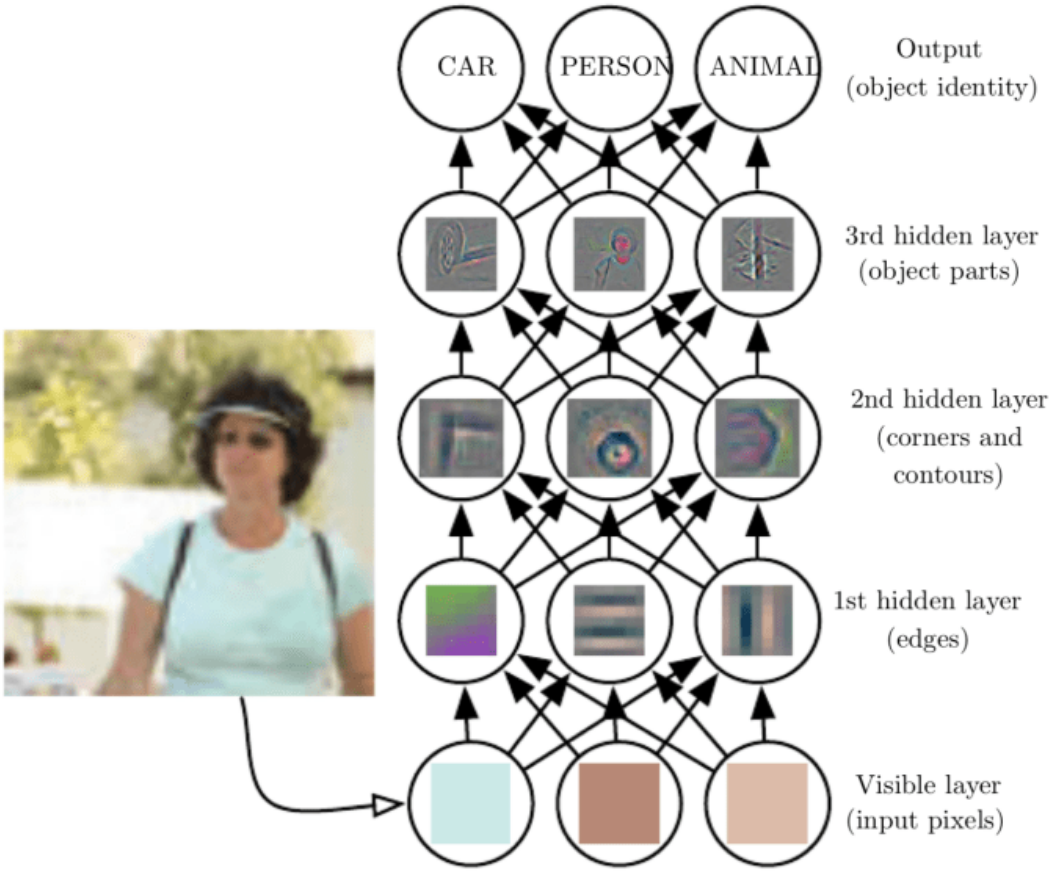


Fig 7: Working of a CNN

3. Recurrent Neural Networks (RNNs)

RNN is a type of supervised deep learning where the output from the previous step is fed as input to the current step. RNN deep learning algorithm is best suited for sequential data. RNN is most preferably used in image captioning, time-series analysis, natural-language processing, handwriting recognition, and machine translation.