Sign Language Prediction Using Machine Learning Algorithm

Ayaan Jamal

Delhi Public School
RK Puram, Delhi

Abstract- **As a child, I was always told that God has made all of us equal. However, my growing years opened my eyes to a not-so-equal world - with so many disadvantaged people in the world. One such category is persons with hearing and speech impairment. These individuals face barriers in their ability to communicate among themselves and with the rest of society. They learn to use sign language to be able to communicate. However, it is a harsh reality that most of us either do not feel the need or lack motivation to learn these languages. There is a need to improve the condition of these less privileged people to better integrate within the society. Through this project, I aim to build a novel algorithm(model) for sign language prediction using Convolutional Neural Network (Machine Learning). I used a dataset of images containing sign languages with more than 27455 images for training and 7172 for testing. We achieved an accuracy of 100% on the test set with this model. This will help better understand and establish communication with the speech and hearing impaired by understanding the meanings of various patterns of sign language. It will create an enabling framework to understand each other in a better way. This will lead to a brighter and more equal world.**

***Index Terms*: Speech Impaired; Hearing Impaired; Sign Language Prediction; Machine Learning; Artificial Intelligence**

1. Introduction

I am deeply passionate about helping people. Be it sharing knowledge with underprivileged children in government schools or distributing possessions to the needy, it is something that gives me the utmost happiness. A disadvantage like hearing or speech impaired is a huge challenge. It curbs one out of the society. This problem must be addressed. Every person has the right to live a full life without constraints. Since we do not have the time to learn sign languages nor sufficient motivation, it is important to devise ways in which we can communicate with these people.

Sign languages play a vital role in the lives of people who have hearing or speech challenges. Sign languages are distinct and fully developed languages that utilize visual-manual communication to convey meaning. They have their own grammar, vocabulary, and linguistic features, making them complete and complex linguistic systems. It's important to recognize that sign languages are not universal and are specific to different communities or regions. While there might be some similarities between sign languages due to shared features, they are generally not mutually intelligible across different languages. Figure 1 shows the list of all signs representing A-Z. Figure 2 presents some basic words in sign language.

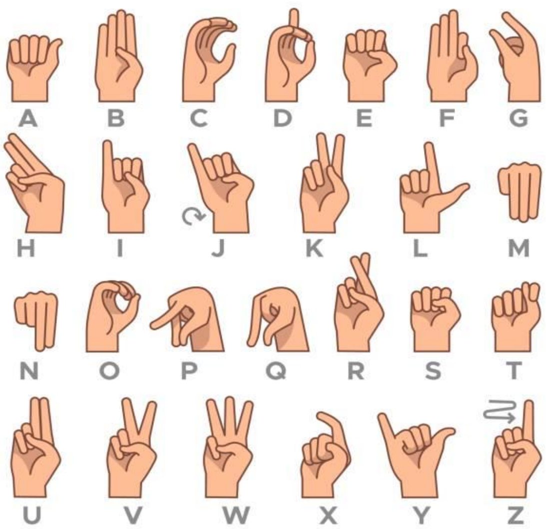


Figure 1 - Sign Language (A-Z) [1]

Figure 2 - Some Basic Words in Sign Language [2]

Artificial Intelligence (AI) is a valuable upcoming asset. It is built to function as a human if not even better and more precisely. Convolutional neural network (CNN) is a branch of AI under Deep Learning, which can learn to predict image-based tasks. It has been emerging as a very strong leader in many different fields. These include medicine, signature, handwriting, and image recognition. These models help to perform tasks that humans find difficult and time-consuming. Once fed with a lot of data, these networks can prove to be extremely helpful. Building an algorithm to learn and predict sign language can help this community of individuals with hearing and speech difficulties.

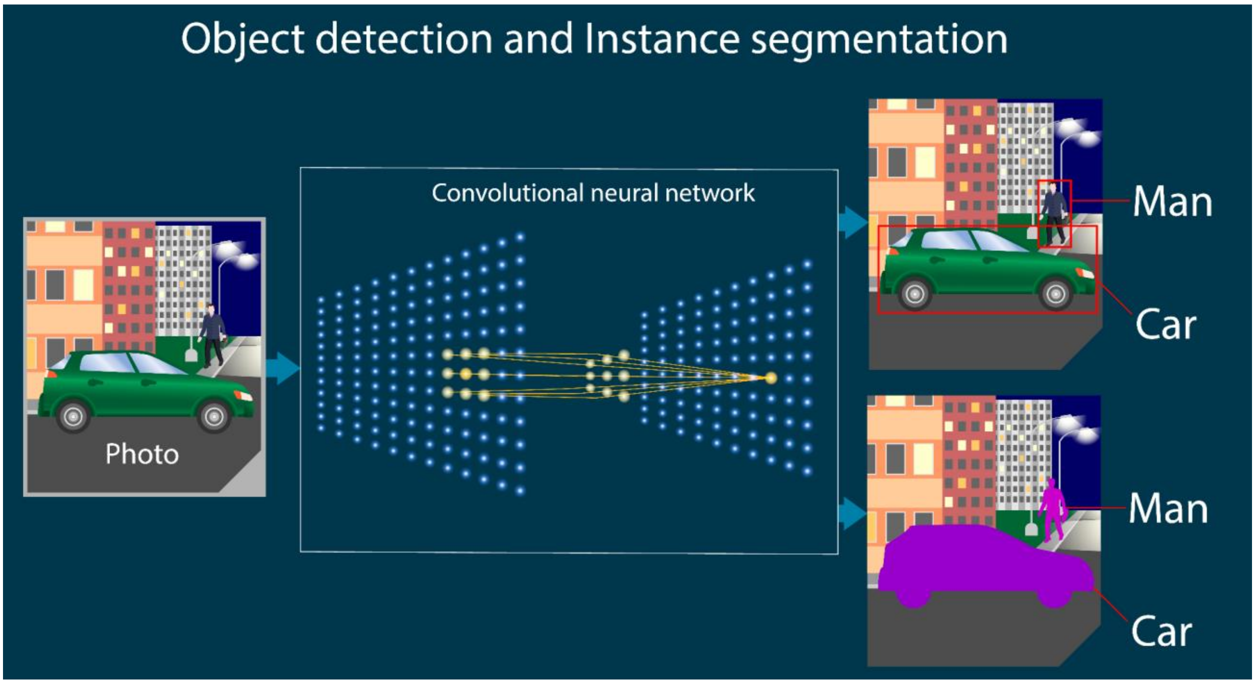


Figure 3 - Real Life Example of how Convolution Neural Networks read and decipher images [3]

Sign Language Prediction is a field that has already been indulged and researched in. There have been many successful models in the past. For sign language prediction, a series of steps need to be followed. People who are unable to speak learn sign language to communicate. Our aim is to build an algorithm that will be able to read the sign languages and successfully and efficiently convert the signals into words. We will use a large dataset which will help to provide a huge amount of data to the model. Thus, the algorithm will be able to predict many different signs. I also aim to improve the existing models by building a novel CNN algorithm that is more accurate and user-friendly. Taking forward this valuable initiative is important in order to provide the disadvantaged with a platform to better express their views, communicate, and better integrate with society.

2. Literature Review

2.1 Convolutional Neural Network

Convolutional Neural Network (CNN) is a branch of Artificial Intelligence (AI) and Machine Learning (ML) that is commonly used for image processing tasks. CNN involves methods to perform functions that would take a lot of time for humans. Some examples include face recognition, handwriting recognition [4], and the subject of this research paper, sign language prediction. In recent years, CNN has been used in many medical science applications, such as radiology and cancer detection, and emotion and facial recognition [4]. A lot of work is also being put into hand gesture recognition in this vast world of computer vision [5]. In the case of skin cancer detection, an algorithm scans a mole on a person's skin and conducts all the necessary processes to define the mole as benign or malignant. This enables a person, sitting at home, to know if he/she has skin cancer. CNN models are extremely valuable and are set to lead in many different aspects of our lives.

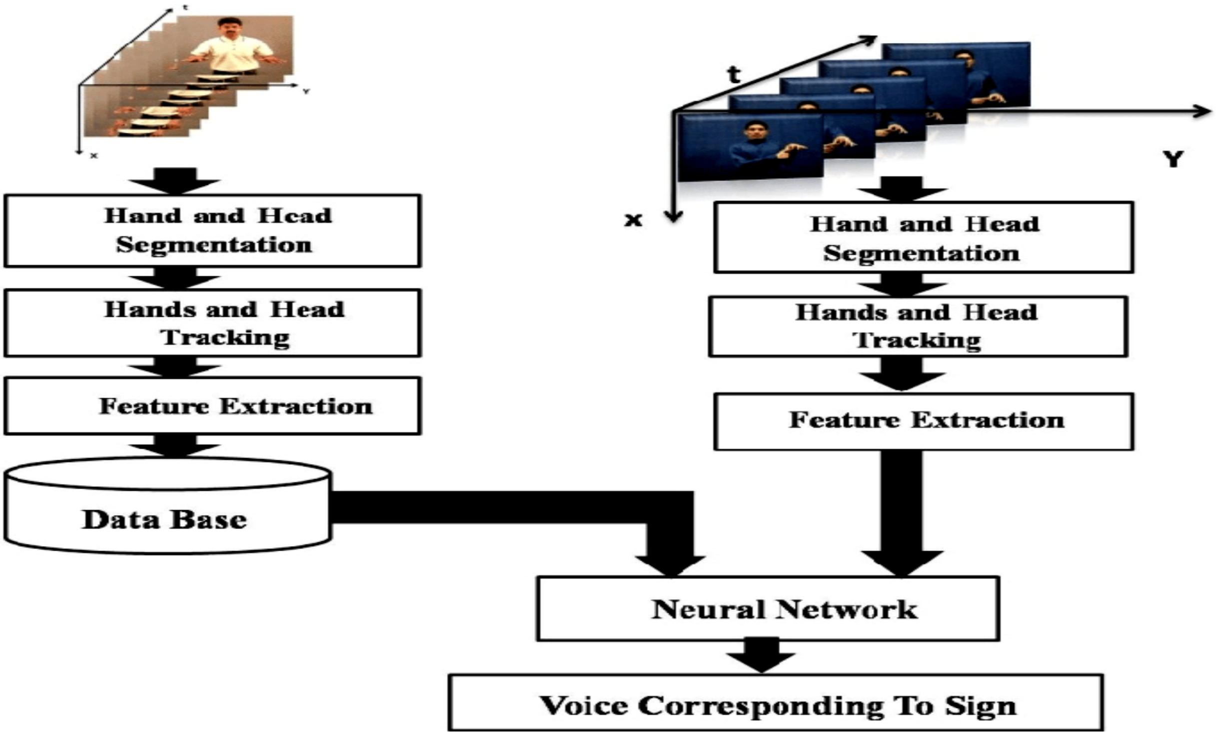


Figure 4 - Methods involved in Convolutional Neural Network [6]

CNN architecture has three layers: convolutional layers, pooling layers, and fully connected layers. Convolutional layers largely deal with feature extraction, pooling layers with dimensionality reduction whereas the fully connected layers convert the obtained features into the output. In CNN, the algorithms are designed to automatically learn the spatial hierarchies of characteristics. This is performed through a method called backpropagation [4]. Feature extraction is an important step in every conventional pattern recognition task. There are two cases for feature extraction. These include extracting features from (i) static images and (ii) dynamic images. Static images make it easy to perform this task and also have a lower computational cost. However, for dynamic images, it is required to combine a sequence of gestures and movements to extract the features [7]. Images are processed through kernels, which are also referred to as 'feature extractors'. A lot of data needs to be provided to the algorithm/model to make it as accurate as possible. This is because a large amount of data embodies all different possibilities of images occurring, through rigorous training and testing. CNN is still a work in progress and has flaws such as overfitting (when the model obtains unnecessary information during training and does not perform very well on a new dataset). It is, however, tipped to be the leader of the world in the upcoming years.

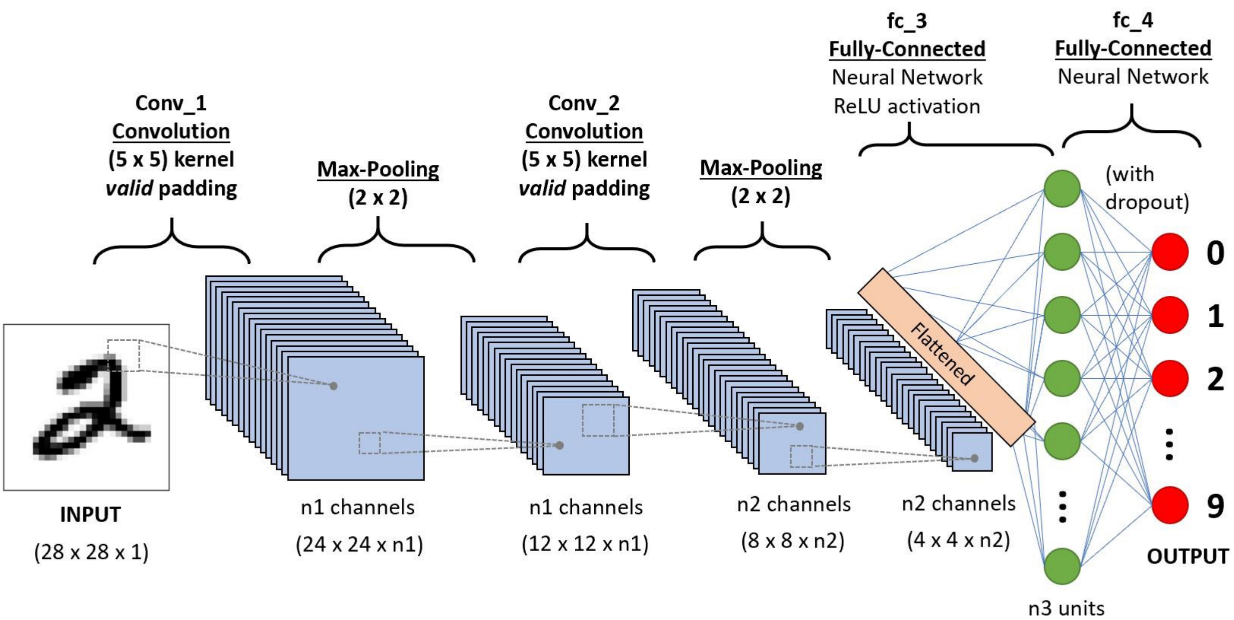


Figure 5 - Convolutional Neural Network - Architecture [8]

2.2 Sign Language Prediction using CNN

Many people in the world are at an extreme disadvantage. One category of such people includes the speech and hearing impaired. These people have limited means to communicate. We need to step up and provide solutions so that they can better integrate into society. In the past, we have witnessed inventions for Sign Language prediction, thus making their lives better. To carry on this initiative, in this study, we plan to predict Sign Language using a Machine Learning Algorithm. Subsequently, we aim to construct a novel algorithm and model (using CNN) in order to make Sign Language Prediction more accurate. 'Sign Language Prediction using Machine Learning Algorithm' is a topic that intrigued me as soon as I ventured upon it. I have previously been associated with helping the less privileged. I believe that this idea has huge potential to help such people all around the world.

Md. Moklesar Rahman et al. have used specific methods to perform the task of Sign Language Prediction. They used the Massey University Gesture dataset which included standard ASL hand gestures comprising 2425 images.[9]. Another method used by Nipun Jindal et al. states that the process of sign language prediction includes several vital steps: Image Acquisition, Preprocessing, segmentation, feature extraction, and outputs. They mention drawbacks due to the lack of 100% accuracy (which will be the case for most models).[10] A similar procedure is used by Karma Wangchuk et al. They say that the prediction takes place through steps that consist of a system overview, data acquisition, image pre-processing, feature extraction, and finally image recognition.[11]. Another study has been conducted using SVM (Support Vector Machine Learning Algorithm) by Vanita Jain et al. It has achieved an accuracy of 81.49% [5]. Building upon these, my goal, through this research project, is to create a novel sign language prediction model which can be used in real-time and make the lives of those in need as convenient as possible.

3. Methodology

3.1. Dataset:

For the CNN model to learn, I used a dataset of sign language images. 34,627 images were divided into two sets of 27,455 images for training and 7,172 images for testing after a division ratio of 0.8:0.2 between both sets. There were 24 different outcomes or classes of sign language in this dataset. The figures have not been diluted with much cleaning. However, they have been processed to be clear enough to identify the 24 outcomes with rigorous training and testing. These images have been sourced from a learning platform called Kaggle.

For each sign language that is part of the dataset, the work relates to this dataset containing the data. Further, it helps to comprehend the data and extract the corresponding meaning. The outcome was eventually presented on the screen for the user to read, understand, and communicate with the disabled person. The data used in the code is freely available and can be accessed from Kaggle. Figures 6 and 7 show the labels of the images along with the pixel values of images for the training and test set respectively. Figures 8 and 9 below show examples of the images in the training and testing sets of the dataset respectively.

(27455, 785)

[3]:

	label	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	...	pixel775	pixel776	pixel777	pixel778	pixel779
0	3	107	118	127	134	139	143	146	150	153	...	207	207	207	207	206
1	6	155	157	156	156	156	157	156	158	158	...	69	149	128	87	94
2	2	187	188	188	187	187	186	187	188	187	...	202	201	200	199	198
3	2	211	211	212	212	211	210	211	210	210	...	235	234	233	231	230
4	13	164	167	170	172	176	179	180	184	185	...	92	105	105	108	133

5 rows × 785 columns

Figure 6 : Labels of the images and the pixel values of images for the training set

(7172, 785)

[4]:

	label	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	...	pixel775	pixel776	pixel777	pixel778	pixel779
0	6	149	149	150	150	150	151	151	150	151	...	138	148	127	89	82
1	5	126	128	131	132	133	134	135	135	136	...	47	104	194	183	186
2	10	85	88	92	96	105	123	135	143	147	...	68	166	242	227	230
3	0	203	205	207	206	207	209	210	209	210	...	154	248	247	248	253
4	3	188	191	193	195	199	201	202	203	203	...	26	40	64	48	29

5 rows × 785 columns

Figure 7 : Labels of the images and the pixel values of images for the testing set

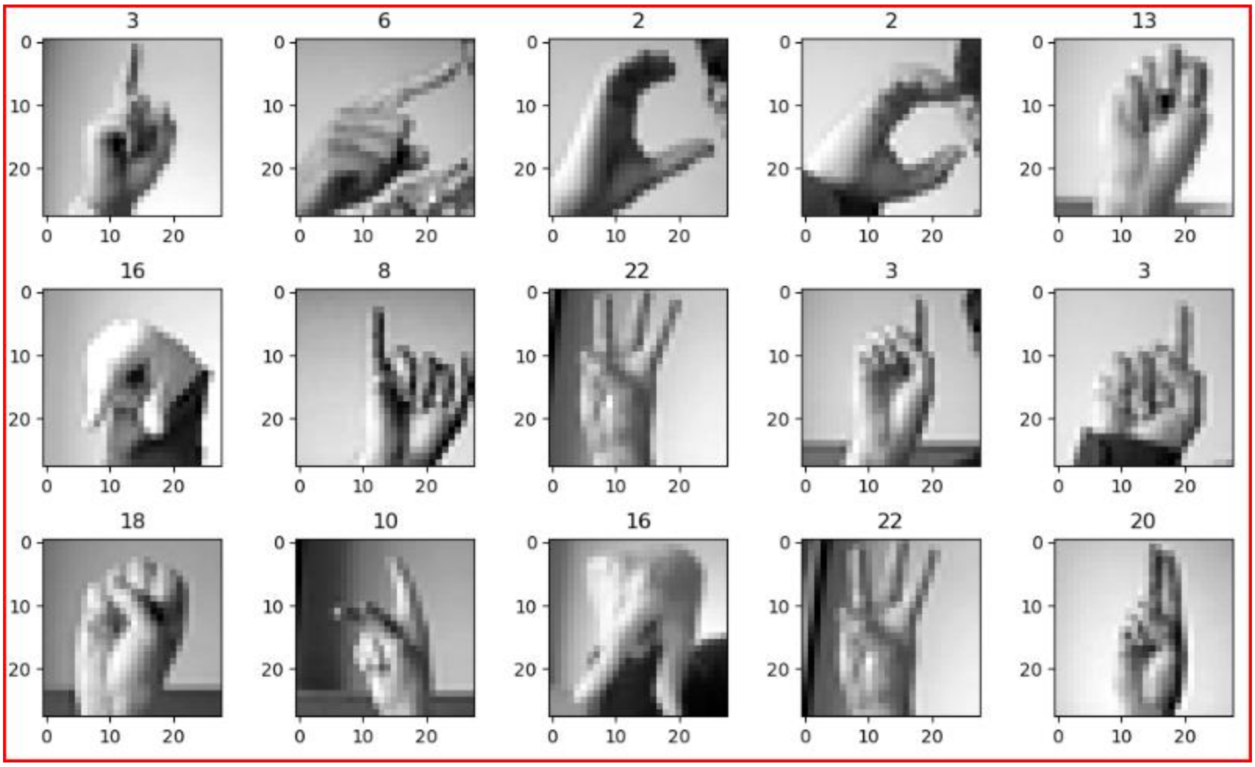


Figure 8 - Examples of the images in the training set

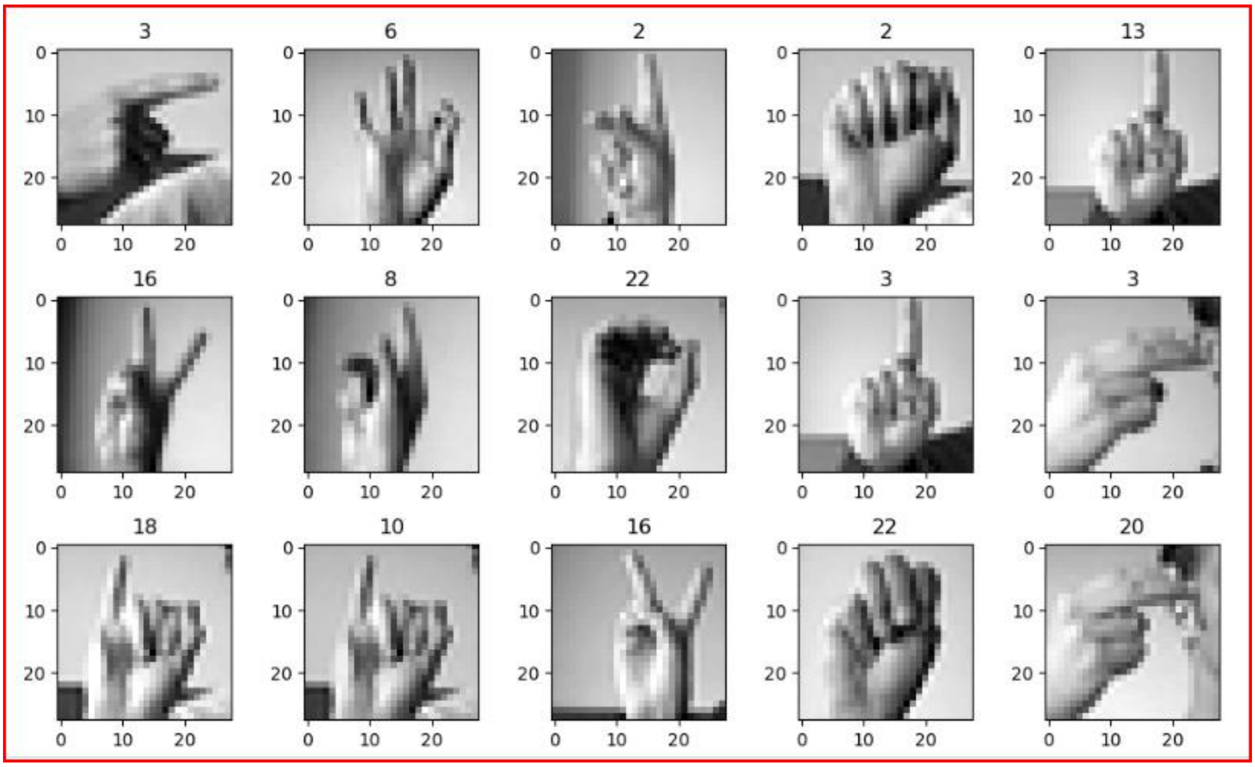


Figure 9 - Examples of the images in the testing set

3.2. Model:

A novel and efficient CNN Model has been constructed for the prediction of sign language. This model features 3 convolutional layers, 3 pooling layers of depth, and 2 fully connected layers. I used a filter size of 3x3. I had 24 outcomes. I have also done batch normalization. For all the convolutional and pooling layers, I have used the ReLU activation function. ReLU (rectified linear unit) is an activation function defined as a positive part of its argument. It can be expressed mathematically as shown below :

$$f(x) = x^+ = \max(0, x) = \frac{x + |x|}{2} = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{otherwise.} \end{cases} \quad f'(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{if } x < 0. \end{cases}$$

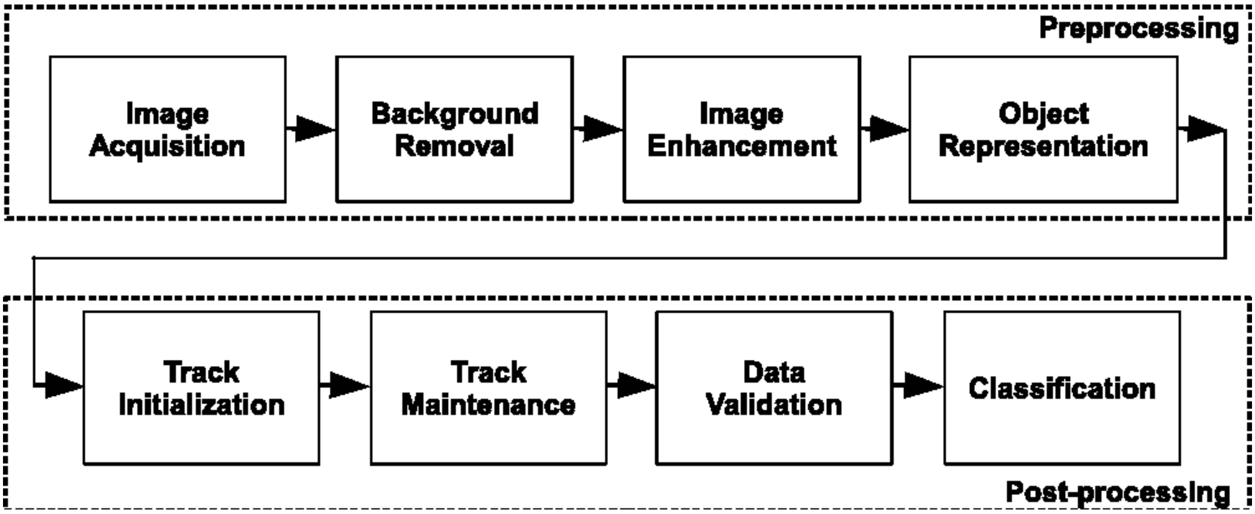
For the final outcome, I used the SoftMax activation function. Softmax is usually used as the last activation function. It helps to normalize the result of the convolutional networks. To compile the model, I used the Adam optimizer with a categorical cross-entropy loss function. The pooling layers were operated after each convolution layer.

3.3. Pre-Processing and Experiment:

The images in the dataset were of varying sizes and dimensions. There was a need to convert all these images into equal sizes for input to train the CNN Model. These images were thus resized into 28x28s. This would reduce the size of the data and increase the speed of computation of the working model. The task was achieved by converting all RGB pixel values and dividing them by 255.

The following steps present the training and validation process of the Algorithm :

1) The pre-processing training images are passed into the model as input.
2) These images enter the convolution layers which extract the features of the images.
3) Finally, the information is interpreted by the convolution neural network's machine learning algorithm.
4) The output is presented on the computer screen.



[12] Figure 10 - Various steps involved in the pre-processing of images in CNN

4. Results and Discussion:

This CNN model is a robust model which also allows for moderating through various parameters. The components of the model can easily be modified. Thus, the model is flexible as well. There were many changes that were made in order to obtain the best version of this model and in order to remove the possibility of errors to the largest extent possible. We experimented with various activation functions including Sigmoid, Tanh, Linear, and ReLU. Among these, ReLU proved to be the most accurate function. A range of epoch values from 5 to 20 were tried. It was observed that the model presented the best results at 13.

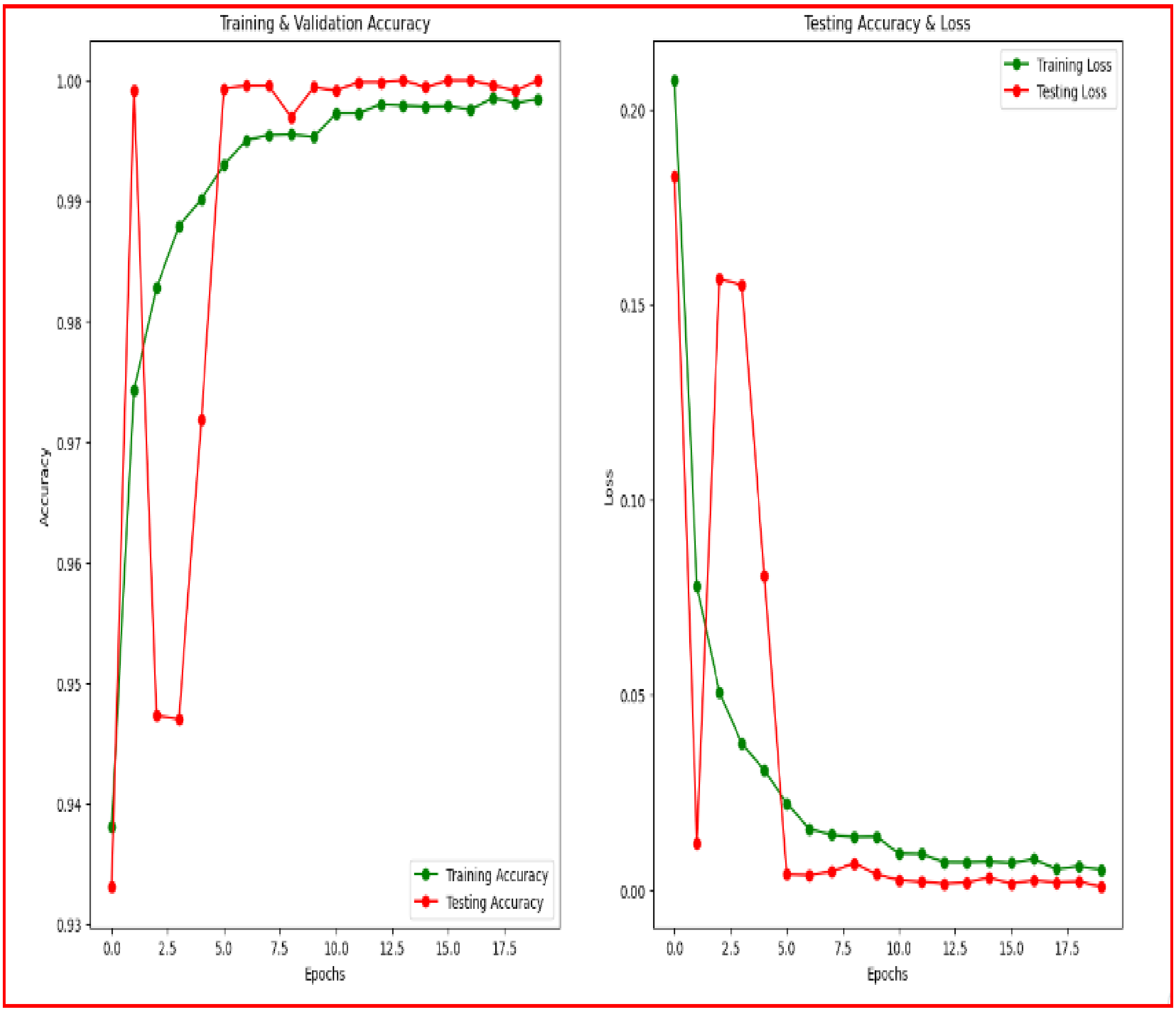


Figure 11 - Accuracy-Epochs and Loss-Epochs Graph

While training, the model achieved an accuracy of XX. The figure (Figure 5) on the left plots the Accuracy-Epochs graph while the figure (Figure 5) on the right plots the Loss-Epochs graph. The two graphs attached below show the details of the outcomes. It shows that the accuracy achieved is much larger than the loss, which is nearly zero. Loss is defined as the difference between the expected and observed results. We can calculate how much worse than the expected outcome was the obtained result. The lesser the loss, the more accurate and precise the model is. In the figures, we can clearly see the two graphs, showcasing the accuracy and loss respectively. We can clearly observe that the accuracy has burgeoned on training and testing the model. On the other hand, the loss has fallen significantly, making our model very strong and trustworthy. On the testing data, we obtained 100% accuracy.

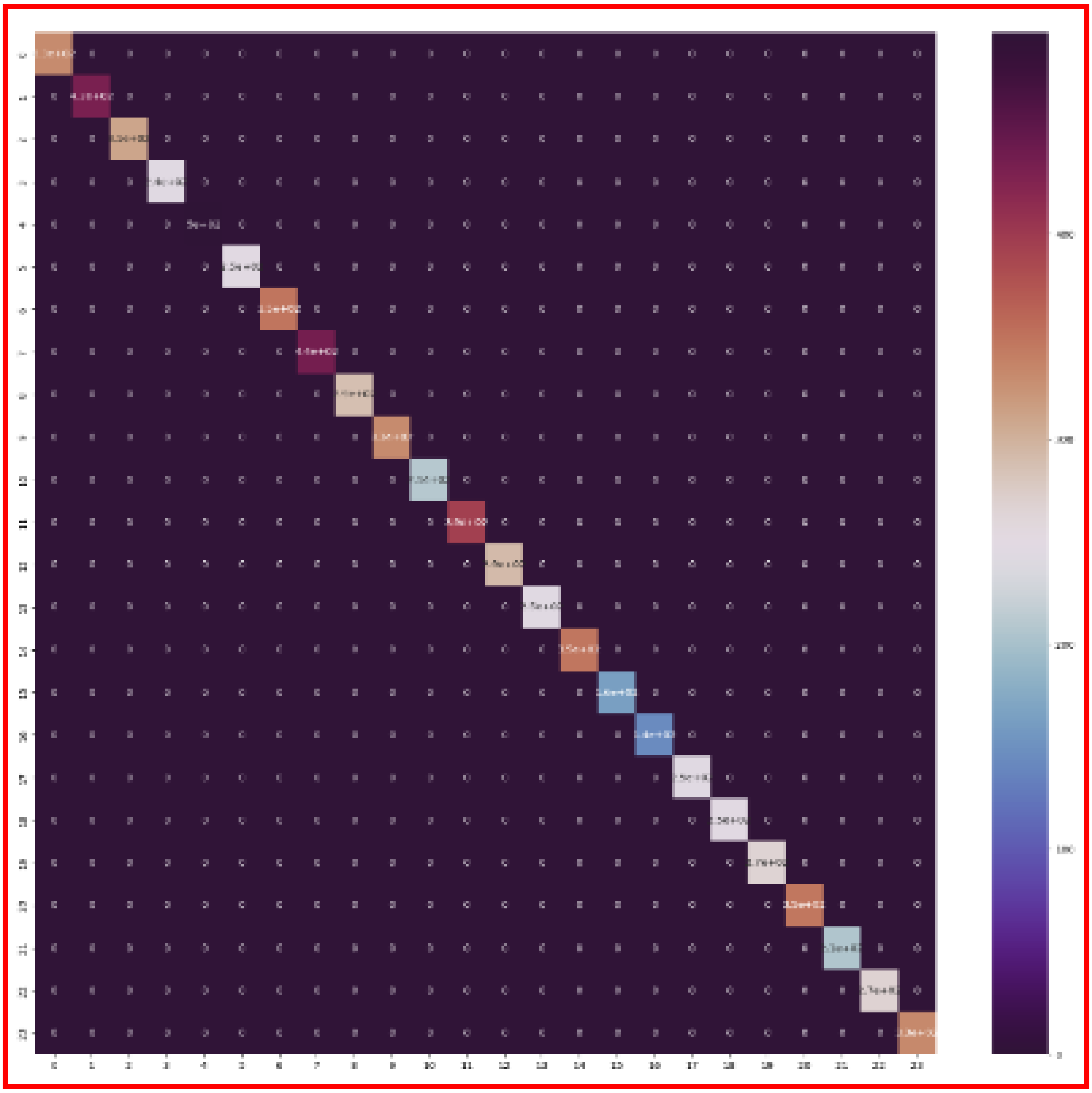


Figure 12 - Confusion Matrix : Showcases how well the algorithm is performing

In the figure attached above, we can see that our model and algorithm is leaning well on all the images that have been fed into the dataset. The confusion matrix obtained is very impressive. Confusion matrix shows how well the model has performed across all aspects.

4.1. Strengths:

We were able to produce a 100% accurate model through the methods and sources we used. We used a large dataset in order to train and test our model to the best extent possible. It proved to be extremely helpful. The model has performed very well, as is evident in the images and figures attached. The images were successfully diluted and cleaned to be of a level that is easily readable and can be sensed by the convolutional neural network. We used all the processes that we talked about in the previous sections of our report. We used different parameters, kept modifying the algorithm and used the best activation function ReLU. These together added up and proved to be the core elements of a perfectly accurate sign language prediction model.

4.2. Limitations:

Although we managed to achieve a model with a 100% accuracy, there are always some limitations which draw a line to the extent we can go to. We could have used more diluted, cleaner and resolution images. We could have made use of video to train and test our model. This is something that can definitely be ventured in the future.

5. Conclusion:

I took on this project as a challenge to help the disadvantaged people who are speech or hearing impaired. We need to appreciate the fact that we are in perfect condition without any disabilities or discomforts. We need to respect and help people who are not as privileged as us. Throughout this project, I came across many different models and methods of solving this problem and tried to arrive at the best and most accurate solution possible. I incorporated all the steps in a careful and thoughtful manner. I used CNN, which is a part of deep learning algorithms, to perform this task. I utilized the three fundamental steps, including convolution layers, pooling layers and fully connected layers. At the end, with the help of the large dataset that I used, I was able to produce an algorithm that predicted sign language with an accuracy of 100%. I hope this model proves to help people in need and makes the world a better place.

REFERENCES:

[1]https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.istockphoto.com%2Fphotos%2Famerican-sign-language-alphabet&psig=AOvVaw3331hN3LGyM104iqzjNz0S&ust=1691990159782000&source=images&cd=vfe&opi=89978449&ved=0CBEQjRxqFwoTCLiTstXw2IADFQAAAAAdAAAAABAK

[2]https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.pinterest.com%2Fpin%2F477100154249864807%2F&psig=AOvVaw1E4W2x10EhrRi5m_7YrPbL&ust=1691990258353000&source=images&cd=vfe&opi=89978449&ved=0CBEQjRxqFwoTCNil0ITx2IADFQAAAAAdAAAAABAE

[3]https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.mygreatlearning.com%2Fblog%2Fcnn-model-architectures-and-applications%2F&psig=AOvVaw0bbS21JlLNsudn86B2buZR&ust=1691980712987000&source=images&cd=vfe&opi=89978449&ved=0CBEQjRxqFwoTCOjmoLzN2IADFQAAAAAdAAAAABAE

[4] Yamashita, R., Nishio, M., G, R. K., DO, & Togashi, K. (2018b). Convolutional neural networks: an overview and application in radiology. *Insights Into Imaging*, *9*(4), 611–629. https://doi.org/10.1007/s13244-018-0639-9

[5] Jain, V., Jain, A., Chauhan, A., Kotla, S. S., & Gautam, A. (2021). American Sign Language recognition using Support Vector Machine and Convolutional Neural Network. *International Journal of Information Technology*, *13*(3), 1193–1200. https://doi.org/10.1007/s41870-021-00617-x

[6]https://www.researchgate.net/figure/Sign-Language-Recognition-System-Architecture_fig4_260322218

[7] Mannan, A., Abbasi, A., Javed, A. R., Ahsan, A., Gadekallu, T. R., & Xin, Q. (2022). Hypertuned deep convolutional neural network for sign language recognition. *Computational Intelligence and Neuroscience*, *2022*, 1–10. https://doi.org/10.1155/2022/1450822

[8]https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.analyticsvidhya.com%2Fblog%2F2020%2F10%2Fwhat-is-the-convolutional-neural-network-architecture%2F&psig=AOvVaw19LZel44Bkc-U-GYYs8Jov&ust=1691951793287000&source=images&cd=vfe&opi=89978449&ved=0CBEQjRxqFwoTCOCJrd7h14ADFQAAAAAdAAAAABAE

[9] Rahman, M. M., Islam, S., Rahman, H., Sassi, R., Rivolta, M. W., & Aktaruzzaman. (2019c). *A New Benchmark on American Sign Language Recognition using Convolutional Neural Network*. https://doi.org/10.1109/sti47673.2019.9067974

[10] Jindal, N., Yadav, N., Nirvan, N., & Kumar, D. (2022). Sign Language Detection using Convolutional Neural Network (CNN). In *2022 IEEE World Conference on Applied Intelligence and Computing (AIC)*. https://doi.org/10.1109/aic55036.2022.9848844

[11] Wangchuk, K., Riyamongkol, P., & Waranusast, R. (2021). Real-time Bhutanese Sign Language digits recognition system using Convolutional Neural Network. *ICT Express*, *7*(2), 215–220. https://doi.org/10.1016/j.icte.2020.08.002

[12]https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.semanticscholar.org%2Fpaper%2FRealization-of-preprocessing-blocks-of-CNN-based-on-Savkay-Yildiz%2Fae2d99cf8e2def3fb01aebc8dd71af6aee98d856&psig=AOvVaw2NRx3OXWDYwXJSvYH9yf-X&ust=1691980065951000&source=images&cd=vfe&opi=89978449&ved=0CBEQjRxqFwoTCNCj7YfL2IADFQAAAAAdAAAAABAJ

7. Appendix

```
def build_model(hp):
  model = keras.Sequential([
    keras.layers.Conv2D(
        filters=hp.Int('conv_1_filter', min_value=75, max_value=200, step=25),
        kernel_size=(3,3),
        activation='relu',

        input_shape=(28,28,1)
    ),
    keras.layers.BatchNormalization(),
    keras.layers.MaxPool2D(pool_size=(2,2),strides=2,padding='same'),
    keras.layers.Conv2D(
        filters=hp.Int('conv_2_filter', min_value=50, max_value=125, step=25),
        kernel_size=(3,3),
        activation='relu',
    ),

    keras.layers.Dropout(
        rate = hp.Choice('drop_1_rate', values = [0.1,0.5])
    ),
    keras.layers.BatchNormalization(),
    keras.layers.MaxPool2D(pool_size=(2,2),strides=2,padding='same'),keras.layers.Conv2D(
        filters=hp.Int('conv_3_filter', min_value=25, max_value=75, step=25),
        kernel_size=(3,3),
        activation='relu',
    ),
    keras.layers.BatchNormalization(),
    keras.layers.MaxPool2D(pool_size=(2,2),strides=2,padding='same'),
    keras.layers.Flatten(),
    keras.layers.Dense(
        units=hp.Int('dense_1_units', min_value=128, max_value=1024, step=32),
        activation='relu'
    ),
    keras.layers.Dropout(
        rate = hp.Choice('drop_2_rate', values = [0.1,0.3])
    ),
    keras.layers.Dense(24, activation='softmax')
  ])

  model.compile(optimizer=keras.optimizers.Adam(hp.Choice('learning_rate', values=[1e-2, 1e-
3])),
              loss='categorical_crossentropy',
              metrics=['accuracy'])

  return model
```

Figure 13 - Code: Displays the steps that were applied to reach the desired output