

University Institute of Engineering and Technology, CSJMU Kanpur

B.Tech Project

Report

GROUP-1

Submitted By

Adarsh Kumar Nishad

Aishwary Bajpai

Shubham Yadav

**B.Tech Project Report**

*on*

**“Sign Language Recognition Using Machine Learning”**

*Submitted in the fulfilment of the requirement for the award of the degree of*

**Bachelor of Technology**

*in*

**Computer Science & Engineering**

*By*

**Adarsh Kumar Nishad (CSJMA18001390004)**

**Aishwary Bajpai (CSJMA18001390007)**

**Shubham Yadav (CSJMA18001390050)**

*Under the Supervision of*

**Er. Suruchi Singh**



**Department of Computer Science & Engineering**

University Institute of Engineering and Technology

CSJMU Kanpur 208024

CERTIFICATE

*We hereby certify that the work which is being presented in the B.Tech Project Report entitled* **''****SIGN LANGUAGE RECOGNIGATION USING MACHINE LEARNING"** *, in partial fulfillment of the requirements for the award of the* **Bachelor of Technology in Computer Science and Engineering** *and submitted to the Department of Computer Science and Engineering of University Institute of Engineering and Technology ,Kanpur , UP is an authentic record of my own work carried out during a period from July 2016 to December 2016 under the supervision of* **Er. Suruchi Sigh (Assistant Professor, Department Computer Science & Engineering,**

**UIET Kanpur).**

*The matter presented in this thesis has not been submitted by me for the*

*award of any other degree elsewhere.*

**ADARSH KUMAR NISHAD (CSJMA18001390004)**

**AISHWARY BAJPAI (CSJMA18001390007)**

**SHUBHAM YADAV (CSJMA18001390050)**

*This is certify that the above statement made by the candidate is correct to the best of my knowledge.*

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Date: Er. Suruchi Singh**

(**Assistant Professor,** Department Computer Science & Engineering,

UIET Kanpur)

**DECLARATION**

*We declare that this project report titled* “**SIGN LANGUAGE RECOGNITATION USING MACHINE LEARNING”** *submitted in partial fulfillment of the degree of* “**B. Tech in Computer Science and Engineering and Technology”** *is a record of original work carried out by us under the supervision of* **Er. Suruchi Singh** *and has not formed the basis for the award of any other degree or diploma, I this or any other Institute or University. In keeping with the ethical practice in reporting scientific information due acknowledgments have been made wherever the findings of other have been cited.*

**ADARSH KUMAR NISHAD (CSJMA18001390004)**

**AISHWARY BAJPAI (CSJMA18001390007)**

**SHUBHAM YADAV (CSJMA18001390050)**

ACKNOWLEDGEMENT

*There is always a sense of gratitude which one express towards others for their help and supervision in achieving the goals. This formal piece of acknowledgement is an attempt to express the feeling of gratitude towards people who helpful me in successfully completing of my training*

*We would like to express our deep gratitude to* ***Er. Suruchi Singh*** *mam for his valuable guidance and advice. He was always there with his competent guidance and valuable suggestion throughout the pursuance of this project.*

*We would like to take this opportunity to thanks to the Computer Science and Engineering Department of UIET for offering this project. It gave us the opportunity to participate and learn about sign language recognition using machine learning*

*Above all no words can express our feelings to our friends all those persons who supported us during our project. We also thankful to all the respondents whose cooperation & support has helped me a lot in collecting necessary information*

**ADARSH KUMAR NISHAD (CSJMA18001390004)**

**AISHWARY BAJPAI (CSJMA18001390007)**

**SHUBHAM YADAV (CSJMA18001390050)**

ABSTRACT

Sign language, as a different form of the communication language, is important to large groups of people in society. There are different signs in each sign language with variability in hand shape, motion profile, and position of the hand, face, and body parts contributing to each sign. So, visual sign language recognition is a complex research area in computer vision. In today's era computer and human interaction technology are developing day by day. One of them is sign language recognition system in which we convert gesture into text of known language.

Our aim is to solve this problem by developing a web application using computer vision and machine learning algorithm which takes input as images through webcam and hence, predicts the correct alphabet up to maximum accuracy. Sign recognition is very innovative and user-friendly technology to interact with computer machine. Sign language mostly used to communicate with dumb and deaf and also used in some artificial intelligence loaded machines. Our project is mainly focusing on hand gesture recognition system in which we develop a real time gesture interpretation web-based application.

The method used in our project is totally based on hand gesture its does not depend on physical appearance. first, we will capture the hand gesture from web cam and then we will convert it into readable text according to American sign language by using CNN (convolutional neural network and some useful function of python language.

TABLE OF CONTENT

**S.NO. TOPIC PAGE NO.**

**Certificate……………………………………………………….…………………………i**

**Declaration………………………..…..…………………………………………………ii**

**Acknowledgement……………………………………………………………………iv**

**Abstract…………………………………………………………………………………….v**

1. **Introduction …………………………………………………..…….……..……09-11**
2. **Algorithms and Methods…….………………………………………………12 - 14**

2.1. Convolutional Neural Network (CNN)……………………………………………12

2.2. Convolution…………………………………………………………………………..………12

2.3. Subsampling……………………………………………………………………………..……13

2.4. Activation……………………………………………………………………………….………13

2.5. Fully Connected………………………………………………………………………………14

1. **Implementation………………..….…………………………………………………15**
2. **Description of Overall Software Structure………………………..…..…16**
3. **Sources of Data………..………..….………………………………………..………17**
4. **Model Performance..………..…...……………………………………….………18**
5. **Conclusion & Future Work..………..…...…… ……………………………19 -20**
6. **References ……………………………………………………………………………..21**

Introduction

Motion of any body part like face, hand is a form of gesture. Here for gesture recognition we are using image processing and computer vision. Gesture recognition enables computer to understand human actions and also acts as an interpreter between computer and human. This could provide potential to human to interact naturally with the computers without any physical contact of the mechanical devices. Gestures are performed by deaf and dumb community to perform sign language. This community used sign language for their communication when broadcasting audio is impossible, or typing and writing is difficult, but there is the vision possibility. At that time sign language is the only way for exchanging information between people. Normally sign language is used by everyone when they do not want to speak, but this is the only way of communication for deaf and dumb community. Sign language is also serving the same meaning as spoken language does. This is used by deaf and dumb community all over the world but in their regional form like ISL, ASL. Sign language can be performed by using Hand gesture either by one hand or two hands. It is of two type Isolated sign language and continuous sign language. Isolated sign language consists of single gesture having single word while continuous ISL or Continuous Sign language is a sequence of gestures that generate a meaningful sentence.

In this report we performed isolated ASL gesture recognition technique. Sign Language Deaf people around the world communicate using sign language as distinct from spoken language in their everyday a visual language that uses a system of manual, facial and body movements as the means of communication. Sign language is not an universal language, and different sign languages are used in different countries, like the many spoken languages all over the world. Some countries such as Belgium, the UK, the USA or India may have more than one sign language. Hundreds of sign languages are in used around the world, for instance, Japanese Sign Language, British Sign Language (BSL), Spanish Sign Language, Turkish Sign Language.



**Finger Spelling American Sign Language**

**Algorithms and Method**

**Convolutional Neural Network (CNN)**

Neural networks, as its name suggests, is a machine learning technique which is modeled after the brain structure. It comprises of a network of learning units called neurons. These neurons learn how to convert input signals (e.g. picture of a cat) into corresponding output signals (e.g. the label “cat”), forming the basis of automated recognition.

A convolutional neural network (CNN, or ConvNet) is a type of feed­forward artificial neural network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex.

CNNs have repetitive blocks of neurons that are applied across space (for images) or time (for audio signals etc). For images, these blocks of neurons can be interpreted as 2D convolutional kernels, repeatedly applied over each patch of the image. For speech, they can be seen as the 1D convolutional kernels applied across time windows. At training time, the weights for these repeated blocks are 'shared', i.e. the weight gradients learned over various image patches are averaged.

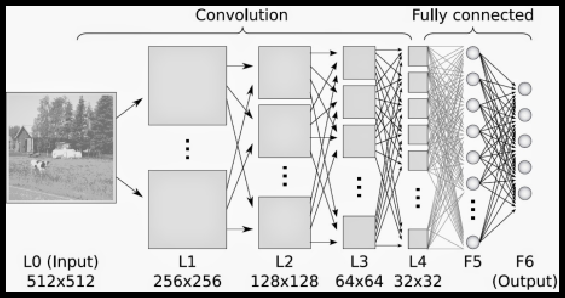
**CNN Summarized in 4 Steps**

There are four main steps in CNN:

•**convolution**

**•subsampling**

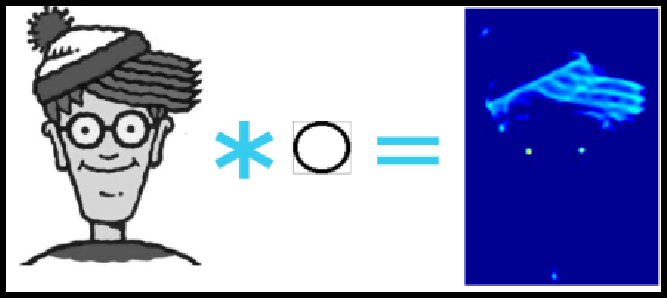
**•activation**

**•full connectedness**

**Convolutional neural network**

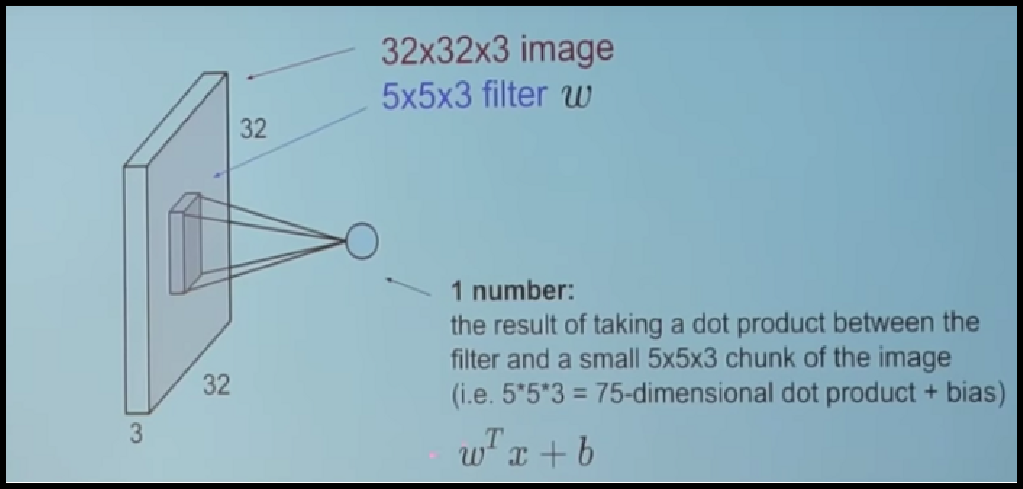
**Convolution**

The first layers that receive an input signal are called convolution filters. Convolution is a process where the network tries to label the input signal by referring to what it has learned in the past. If the input signal looks like previous cat images it has seen before, the “cat” reference signal will be mixed into, or convolved with, the input signal. The resulting output signal is then passed on to the next layer.

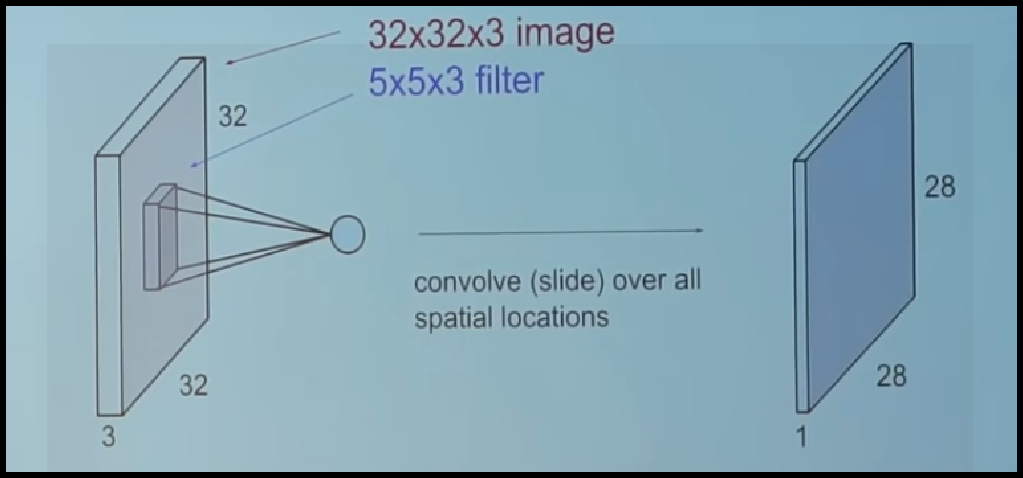


Convolving Wally with a circle filter. The circle filter responds strongly to the eyes.

Convolution has the nice property of being **translational invariant**. Intuitively, this means that each convolution filter represents a feature of interest (e.g whiskers, fur), and the CNN algorithm learns which features comprise the resulting reference (i.e. cat). The output signal strength is not dependent on where the features are located, but simply whether the features are present. Hence, a cat could be sitting in different positions, and the CNN algorithm would still be able to recognize it.

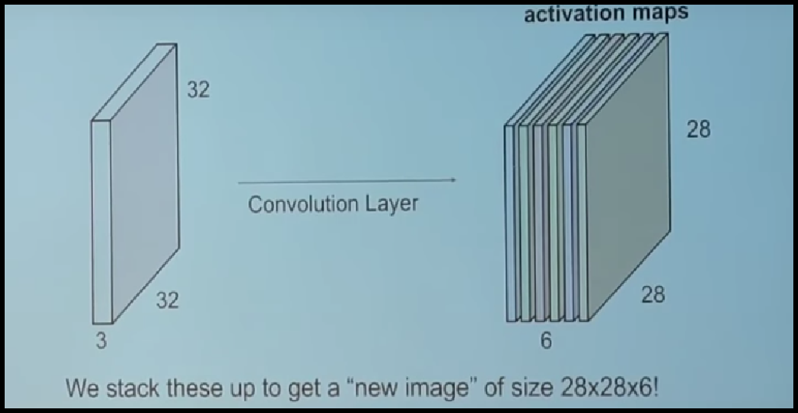
For e.g. suppose we convolve a 32x32x3 (32x32 image with 3 channels R, G and B respectively) with a 5x5x3 filter. We take the 5\*5\*3 filter and slide it over the complete image and along the way take the dot product between the filter and chunks of the input image.

Dot Product of Filter with single chunk of Input Image



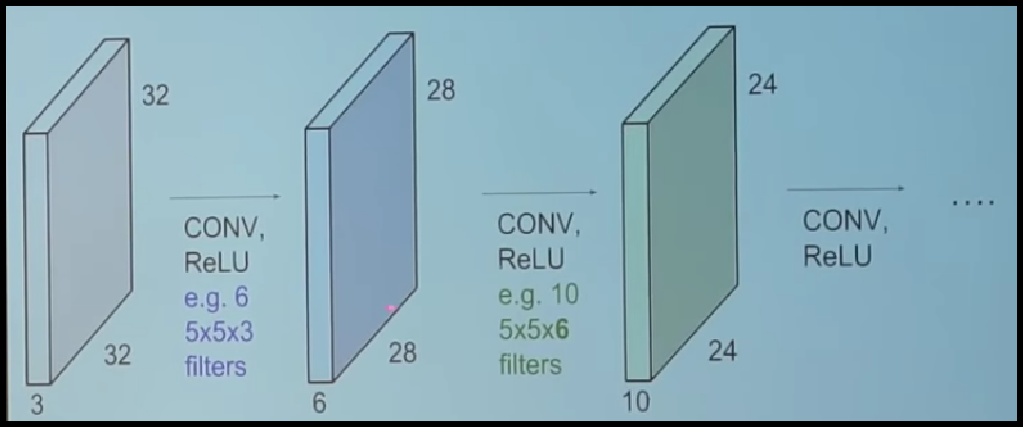
Dot Product or Convolve over all possible 5x5 spatial location in Input Image

The convolution layer is the main building block of a convolutional neural network. The convolution layer comprises of a set of independent filters (6 in the example shown). Each filter is independently convolved with the image and we end up with 6 feature maps of shape 28\*28\*1.



Input Image Convolving with a Convolutional layer of 6 independent filters

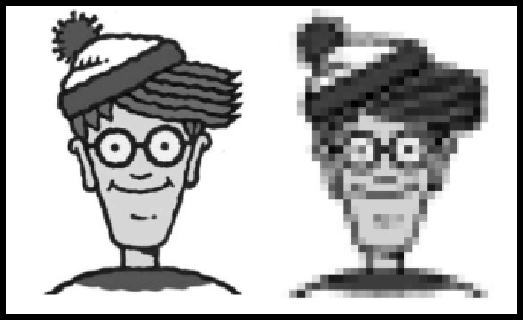
The CNN may consists of several Convolutional layers each of which can have similar or different number of independent filters. For example the following diagram shows the effect of two Convolutional layers having 6 and 10 filters respectively.



Input Image Convolving with two Convolutional layers having 6 and 10 filters respectively

All these filters are initialized randomly and become our parameters which will be learned by the network subsequently

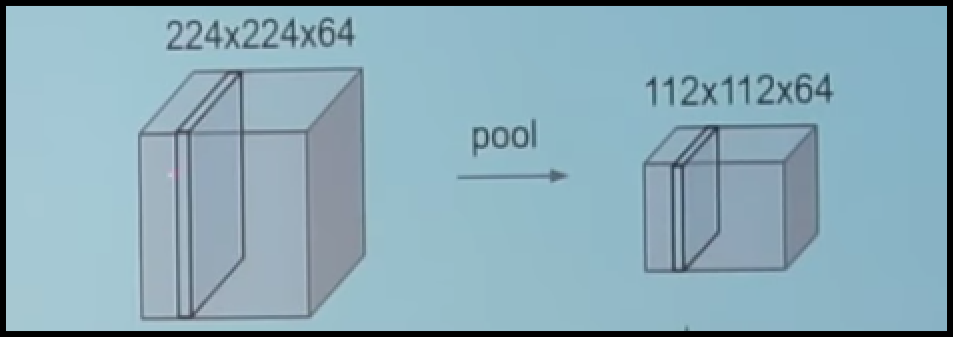
**Subsampling**

Inputs from the convolution layer can be “smoothened” to reduce the sensitivity of the filters to noise and variations. This smoothing process is called subsampling, and can be achieved by taking averages or taking the maximum over a sample of the signal. Examples of subsampling methods (for image signals) include reducing the size of the image, or reducing the color contrast across red, green, blue (RGB) channel.

Sub sampling Wally by 10 times. This creates a lower resolution image.

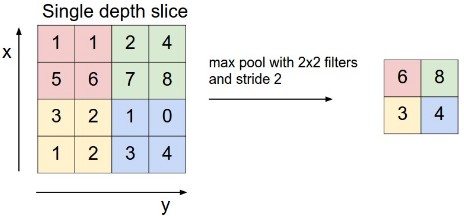
**Pooling**

A pooling layer is another building block of a CNN.



Pooling to reduce size from 224x224 to 112x112

Its function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network. Pooling layer operates on each feature map independently.

The most common approach used in pooling is max pooling in which maximum of a region taken as its representative. For example in the following diagram a 2x2 region is replaced by the maximum value in i.

Max Pooling

**Activation**

The activation layer controls how the signal flows from one layer to the next, emulating how neurons are fired in our brain. Output signals which are strongly associated with past references would activate more neurons, enabling signals to be propagated more efficiently for identification.

CNN is compatible with a wide variety of complex activation functions to model signal propagation, the most common function being the Rectified Linear Unit (ReLU), which is favored for its faster training speed.

**Fully Connected**

The last layers in the network are fully connected, meaning that neurons of preceding layers are connected to every neuron in subsequent layers. This mimics high level reasoning where all possible pathways from the input to output are considered.

**(During Training) Loss**

When training the neural network, there is additional layer called the loss layer. This layer provides feedback to the neural network on whether it identified inputs correctly, and if not, how far off its guesses were. This helps to guide the neural network to reinforce the right concepts as it trains. This is always the last layer during training.

**Implementation**

Algorithms used in training CNN are analogous to studying for exams using flash cards. First, you draw several flashcards and check if you have mastered theconcepts on each card. For cards with concepts that you already know discard them. For those cards with concepts that you are unsure of, put them back into the pile. Repeat this process until you are fairly certain that you know enough concepts to do well in the exam. This method allows you to focus on less familiar concepts by revisiting them often. Formally, these algorithms are called gradient descent algorithms for forward pass learning. Modern deep learning algorithm uses a variation called stochastic gradient descent, where instead of drawing the flashcards sequentially, you draw them at random. If similar topics are drawn in sequence, the learners might over­estimate how well they know the topic. The random approach helps to minimize any form of bias in the learning of topics.

Learning algorithms require feedback. This is done using a validation set where the CNN would make predictions and compare them with the true labels or ground truth. The predictions which errors are made are then fed backwards to the CNN to refine the weights learned, in a so-called backwards pass. Formally, this algorithm is called backpropagation of errors, and it requires functions in the CNN to be differentiable (almost).

CNNs are too complex to implement from scratch. Today, machine learning practitioners often utilize toolboxes developed such as Caffe, Torch, MatConvNet and Tensor flow for their work.

**Description of Overall Software Structure**

**Data Processing**

**Classifying Gesture**

**Training Modal**

**Classifying Gesture**

**Terminal**

**Camera**

**Text**

**Camera**

**Text**

**Recognise.py**

**Camera**

**Text**

**Camera**

**Text**

**Cam Text**

**Front End**

**Flask app**

**App.py**

**Training & Saving Modal**

**Cnn\_modal.py**

**Modal.h5**

**Camera feed**

**Capture.py**

**Dataset**

As shown in Figure, the project will be structured into 3 distinct functional blocks, Data Processing, Training, Classify Gesture. The block diagram is simplified in detail to abstract some of the minutiae:

• **Data Processing:**

The capture.py script contains functions to load the Raw Image Data and save the image data as numpy arrays into file storage. The capturre.py script will load the image data from data.npy and preprocess the image by resizing/rescaling the image, and applying filters and whitening to enhance features. During training the processed image data was split into training, and testing data and written to storage. Training also involves a load cnn.py script that loads the relevant data split into a Dataset class. For use of the trained model in classifying gestures.

• **Training:**

The training loop for the model is contained in cnn.py. The model is trained with hyperparameters obtained from a config file that lists the learning rate, batch size, image filtering, and number of epochs. The configuration used to train the model is saved along with the model architecture for future evaluation and tweaking for improved results. Within the training loop, the training and validation datasets are loaded as Dataloaders and the model is trained using Adam Optimizer . The model is evaluated every epoch on the validation set and the model with best validation accuracy is saved to storage for further evaluation and use. Upon finishing training, the training and validation error and loss is saved to the disk.

• **Classify Gesture:**

After a model has been trained, it can be used to classify a new ASL gesture that is available as a direct input from camera. The user inputs the gesture image and the recognize.py script will pass the image to process and load and preprocess the file the same way as the model has been trained. Model Generated through this process is saved as model.h5.

**Sources of Data**

**Data Collection**

The primary source of data for this project was the compiled dataset of American Sign Language (ASL) called the ASL Alphabet . The dataset is comprised of 52000 images which are 200x200 pixels. There are 26 total classes, each with 2000 images, 26 for the letters A-Z . The images taken from his laptop’s webcam. These photos were then cropped, rescaled, and labelled for use.

**  **

Letter A Letter B Letter C

**Data Pre-processing**

**Dilation, Opening, Closing And Erosion**

These are two fundamental image processing operations. These are used to removing noises, finding an intensity hole or bump in an image and many more.

**Cropping**

It is one of the most important and fundamental techniques in image processing, Cropping is used to get a particular part of an image. To crop an image. You just need the coordinates from an image according to your area of interest

**Scaling, Interpolations, And Re-Sizing**

Re-sizing is one of the easiest tasks in OpenCV. It provides a resize() function which takes parameters such as image, output size image, interpolation, x scale, and y scale



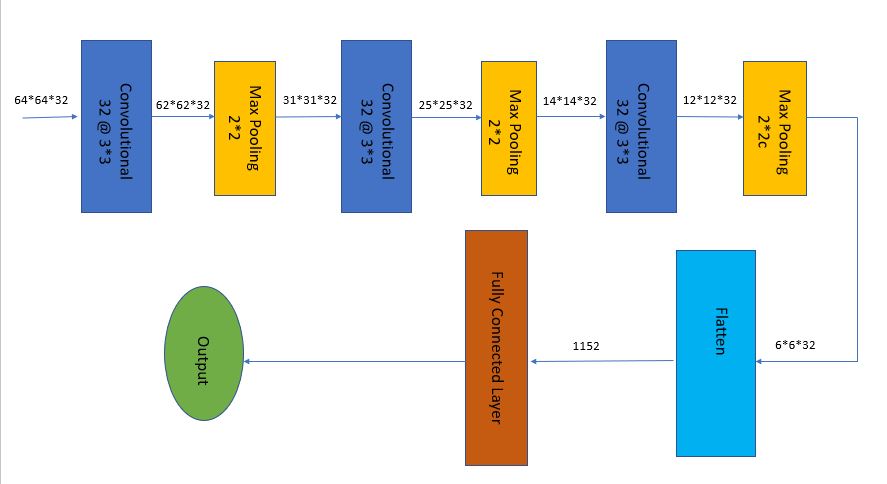
Image from Cam

Image After Contour

Image After Masking

**Machine Learning Model**

Overall Structure The model used in this classification task is a fairly basic implementation of a Convolutional Neural Network (CNN). As the project requires classification of images, a CNN is the go-to architecture. The basis for our model design came from Using Deep Convolutional Networks for Gesture Recognition in American Sign Language paper that accomplished a similar ASL Gesture Classification task . This model consisted of convolutional blocks containing two 2D Convolutional Layers with ReLU activation, followed by Max Pooling and Dropout layers. These convolutional blocks are repeated 3 times and followed by Fully Connected layers that eventually classify into the required categories. The kernel sizes are maintained at 3 X 3 throughout the model.



**Model Performance**

**Training and Validation**

Our models were trained using Adam optimizer and Cross Entropy Loss. Adam optimizer is known for converging quickly in comparison with Stochastic Gradient Descent (SGD), even while using momentum. However, initially Adam would not decrease our loss thus we abandoned it to use SGD. Debugging Adam optimizer after our final presentation taught us that lowering learning rate significantly can help Adam to converge during training. Thus allowing us to train more models towards the end of our project.

**Conclusion & Future Work**

The technology improves the fields of artificial intelligence, machine learning and computer vision. They all together help us to bring the technology which helps society in a better way and improves the lives of human being. Many scientists and researcher use techniques like ANN, CNN etc to conduct research in the field of sign language recognition. many of them use high power of computing our research paper minimizes the high power of computer processing and make it available for every people. We classify 26 alphabet letters of American sign language successfully by using CNN.

We wish to extend our work further in recognising continuous sign language gestures with better accuracy. This method for individual gestures can also be extended for sentence level sign language. Also the current process uses two different models, training inception (CNN) . For future work one can focus on combining the two models into a single model.

**References**

[1] T. Bohra, S. Sompura, K. Parekh and P. Raut, "Real-Time Two-Way Communication System for Speech and Hearing-Impaired Using Computer Vision and Deep Learning," 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2019, pp. 734-739, doi: 10.1109/ICSSIT46314.2019.8987908.

[2] Singha, Joyeeta & Das, Karen. (2013), “Recognition of Indian Sign Language in Live Video,” International Journal of Computer Applications. 70. 10.5120/12174-7306.

[3] H. Muthu Mariappan and V. Gomathi, "Real-Time Recognition of Indian Sign Language," 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), Chennai, India, 2019, pp. 1-6, doi: 10.1109/ICCIDS.2019.8862125.

[4] K. Bantupalli and Y. Xie, "American Sign Language Recognition using Deep Learning and Computer Vision," 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 4896-4899, doi: 10.1109/BigData.2018.8622141.

[5] S. Kim, Y. Ji and K. Lee, "An Effective Sign Language Learning with Object Detection Based ROI Segmentation," 2018 Second IEEE International Conference on Robotic Computing (IRC), 2018, pp. 330-333, doi: 10.1109/IRC.2018.00069.

[6] Deaf Cambridge Dictionary (2018). Retrieved from Cambridge Dictionary: <https://dictionary.cambridge.org/dictionary/english/deaf>

[7] Mehreen Hurroo , Mohammad Elham, 2020, Sign Language Recognition System using Convolutional Neural Network and Computer Vision, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 09, Issue 12 (December 2020),