Real-Time American Sign Language Recognition With Convolutional Neural Networks

Submitted To:
Dr. Ramasubba Reddy
Department of Applied Mechanics
IIT Madras

Done By
Rishab Balasubramanian
NIT Trichy

Acknowledgement

I would like to thank Dr. Ramasubba Reddy, who has been supportive and helpful throughout the course of this project, and also for giving me the oppurtunity to work on this project under him. I would also like to thank all the members of the Biomedical Engineering Lab who have been a constant source of inspiration and for their advice.

Real-Time American Sign Language Recognition With Convolutional Neural Networks

Introduction:

Sign language is a natural language in the form of gestures. It is the primary form of language for the deaf and mute. The language is comprised of several signs and gestures, each sign representing a letter, while each gesture represents a word. It also involves the position of the hands and also facial expressions. In this project we will only focus on the gestures formed by hand.

The motivation of this project is to develop a helpful interactive application, that can be installed anywhere that would be of utmost importance for aiding people. The focus of this project will be limited to static images

Dataset:

To work with sign languages we need a dictionary of images which can be used as a reference. Sign languages are classified as Indian, American, Chinese, etc. Here we will use the American Sign Language as the reference language.

Several images of static hand gestures for a particular alphabet is obtained from the ASL Alphabet Dataset from Kaggle, as well as the Surrey University and Massey University ASL datasets, containing totally 70,000 useful images.

The dataset is divided into 27 classes – 26 letters and 1 class for ‘space’ gesture

Problem Overview:

The problem consists of three tasks to be done in real time:
1. Obtaining video of the user signing (input)
2. Classifying each frame in the video to a letter
3. Reconstructing and displaying the most likely word from classification scores (output)

From a computer vision perspective, this problem represents a significant challenge due to a number of considerations, including:

- Environmental concerns (e.g. lighting sensitivity, background, and camera position)
- Occlusion (e.g. some or all fingers, or an entire hand can be out of the field of view)
- Sign boundary detection (when a sign ends and the next begins)

In this project a normal webcam, present in all laptops, is used to capture and classify the images. However research is also done using depth images obtained from Xbox Kinect. From a real time video each frame is individually obtained, which is then passed through a Convolutional Neural Network(CNN), which is then used to classify the gesture into one of the twenty-seven classes.

Input from the user (using OpenCV):

The input from the user is obtained using the webcam. The video received is split into frames (static images). For the first few frames, the face is detected in the video using the LBP algorithm for face detection. Once the face is detected four points from the face are selected, such that they cover the darkest regions (near the eye), and brightest regions (near the nose, and the forehead). The colour of the skin at these points is taken and the minimum and maximum values of hue, saturation, and value (HSV) from these points is taken, to form a range of skin colours we are to detect (Fig 1).

These values thus form a range of colours very similar to the colour of the face. This is then used as a mask to remove any other colours present in the frame, such that all other coloured noises are removed. The resulting image is a black-and-white image containing only the regions of skin like colour (Fig 2). To this median blurring is applied to remove the salt and pepper noises. The median blurring algorithm is in such a way that a window of a given size is taken, and is slided through the given image, and the value of the centeral pixel is calculated as the median of the other pixel values in the window. The face is then removed so that only the hand region would be detected (Fig 3). After this the frame is then repeatedly eroded and dilated so as to smoothen the image. Erosion takes a window of given size. If all the pixels in the window are 1 (white), the centeral pixel is replaced with a value of 1. Else the centeral pixel is set as 0. This thus reduces the thickness of the white areas. Dilation is the opposite of erosion, and hence increases the area of the white portion (Fig 4).

The contours for this image are then determined and drawn, and the center and moments of the contour is found. A bounding rectangle is drawn on the contour so that the rectangle completely covers the contour, and hence the entire hand (Fig 5).

The area within the rectangle is cropped out, thus forming the region of interest of the hand which we use to determine the letter shown (Fig 6).

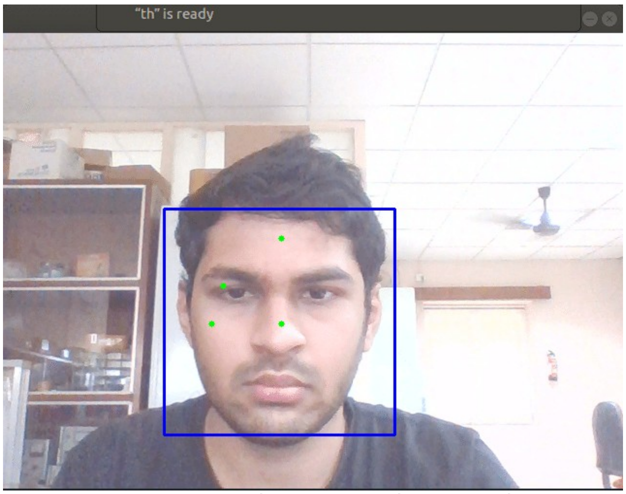


Fig 1: Indicating the four points from the face which are chosen to form the mask

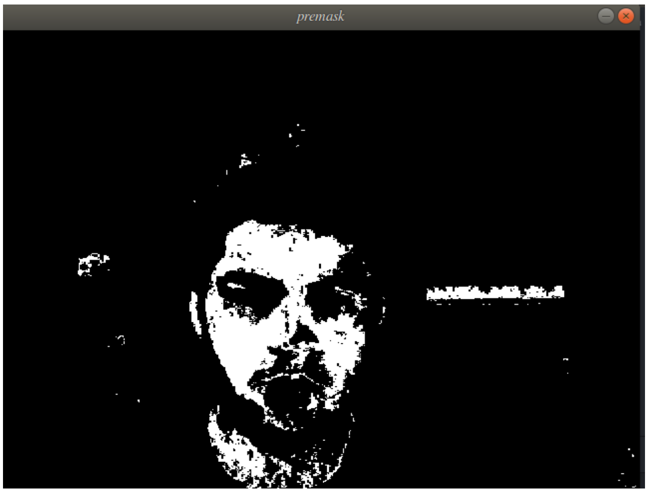
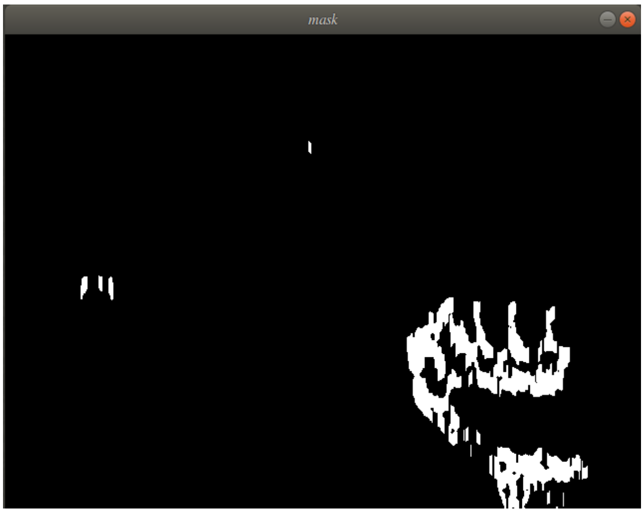


Fig 2: Applying the mask on a given frame removing most of the background



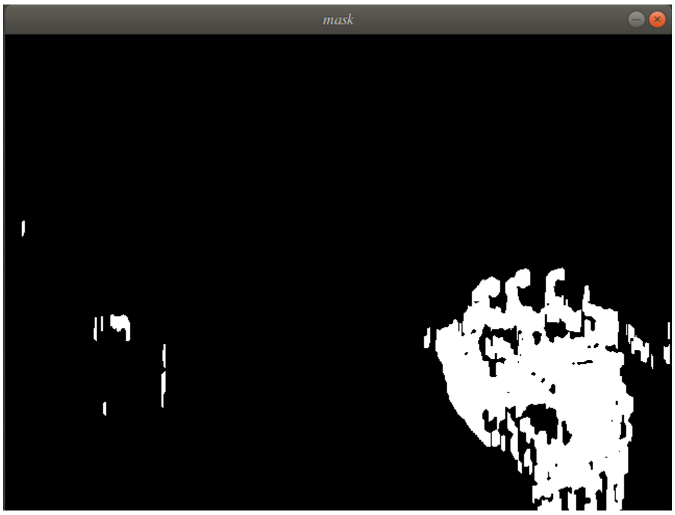
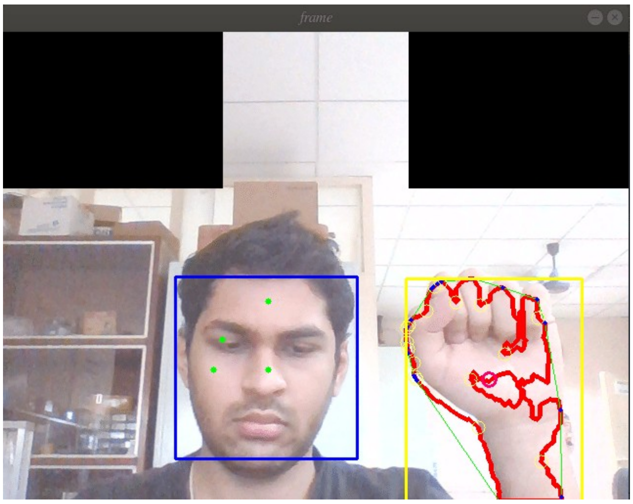


Fig 3: Removing the face region and applying median blur to the given frame

Fig 4: Applying Erosion and Dilation to the frame.



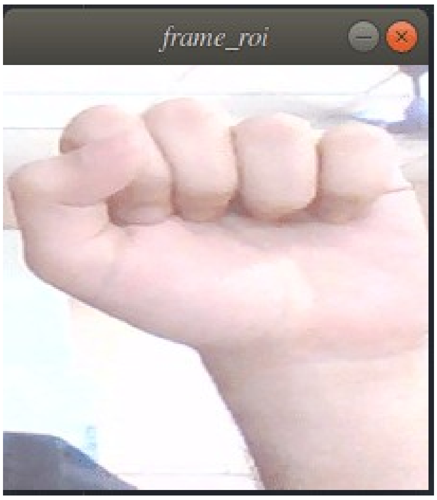


Fig 5:The contours and bounding rectangle are drawn on the hand

Fig 6: The area of the bounfing rectangle is cropped out from the frame thus forming our region of interest

Convolutional Neural Network Architecture:

The machine learning architecture used is the GoogLeNet Architecture, with the number of channels reduced and two more fully connected (Dense) added. Adam optimizer with categorical crossentropy loss function is used. The model is built on Keras and trained from scratch.

TYPE	SIZE/ STRIDE	OUTPUT SIZE	1*1 LAYER	3*3 REDUCE	3*3	5*5 REDUCE	5*5 LAYER	POOL LAYER	PARAMETERS
Input		224*224*3							0
Batch Normailization		224*224*3							12
Conv 2D	7*1 / 2	109*112*32							704
Conv 2D	1*7 / 2	55*53*32							7200
Max Pooling 2D	3*3 / 2	27*26*32							0
Conv 2D	3*3 / 1	25*24*64							18496
Max Pooling 2D	3*3 / 2	12*11*64							0
Inception Layer1		12*11*192	48	72	96	12	24	24	69,618
Inception Layer2		12*11*255	68	68	102	17	51	34	116,059
Max Pooling 2D	3*3 / 2	5*5*255							0
Incaeption Layer3		5*5*480	180	90	195	15	45	60	268,165
Inception Layer4		5*5*480	150	105	210	23	60	60	386,258
Inception Layer5		5*5*480	120	120	240	23	60	60	439,823
Inception Layer6		5*5*512	108	140	280	30	62	62	545,264
Inception Layer7		5*5*525	160	100	203	20	81	81	409,163
Max Pooling 2D	3*3 / 2	2*2*525							
Inception Layer8		2*2*527	162	102	203	21	81	81	347,241
Inception Layer9		2*2*600	225	112	225	28	75	75	498,120
Average Pooling 2D	2*2 / 1	1*1*600							0
Dense (FC) Layer		500							300500
Dense (FC) Layer		200							100200
Dense (FC) Layer		27							5427

Totally 3,584,638 parameters (including 6 non-trainable parameters)

Table 1: The modified GoogLeNet Architecture used for this project

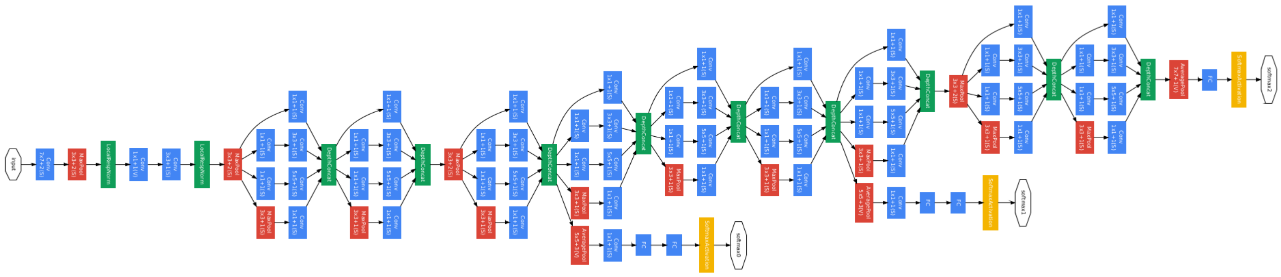


Fig 7: The original GoogLeNet Architecture

It can be observed that the auxillary branches are removed from the network, the number of channels reduced and two extra Dense (Fully Connected Layers) are added at the end of the network. Similar to the original network, the modified architecture uses Rectified Linear Unit (ReLU) activation function for all layers, except the last which uses Softmax activation to classify the images

Putting It All Together:

The cropped frames obtained from the video are first resized into 224*224 pixel size, before passing into the model architecture. The learned parameters are then used to predict the alphabet shown.

Results:

It was observed that the gestures like ‘B’, which had a unique symbol acheived high accuracies (of around 80%). However the signs for alphabets like ‘I’ and ‘J’ or ‘A’ and ‘E’ which have very similar had low accuracies (around 45-50%).

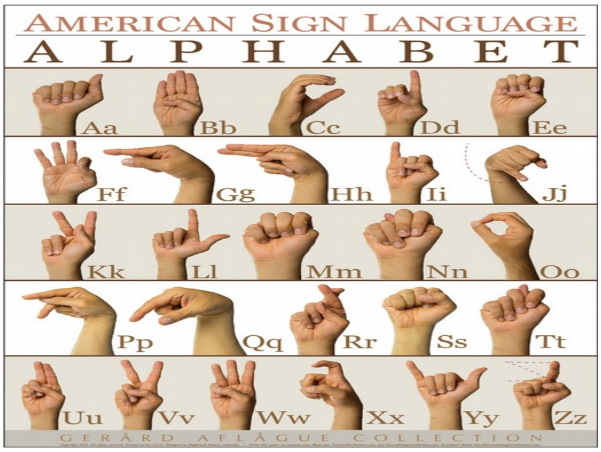


Fig 8: The signs for the American Sign Language

Further Development:

- The model is able to predict alphabets but struggles if the gestures are similar to each other. In this project we hav used the GoogLeNet architecture which has lesser computational abilities. A better algorithm like Google Inception v5 or Google Xception architecture may be used to acheive better results.
- The project focuses on the determination of the alphabets shown. It can be extended to words, which are gestures and hence comprise of video datasets. Using 3D Convolutional Layers and video processing machine learning algorithms this can be accomplished. Along with this comes a need for better letter segmentation and a more seamless process for retrieving images from the user at a higher rate
- A depth sensor can be used in addition to the normal RGB webcam so that accuracy of predictions can be improved.
- A more robust approach can be used to form another CNN to localize and crop the hand.