SIGN LANGUAGE RECOGNITION

Team: Adithya Abhishek Chenthilkannan, Amulya Gupta Vangapalli, Pavan Sai Guduru

Problem Statement

Those who are deaf or hard of hearing primarily communicate through sign language. This type of gesture-based language reduces barriers caused by hearing difficulties and allows people to communicate ideas and thoughts efficiently. A big issue with this practical method of communication is that the vast majority of people do not speak the language. Like learning any other language, studying Sign Language takes a lot of time and effort, which discourages the general population from doing so. Our project aims to close this gap by utilising image identification and machine learning techniques.

Dataset

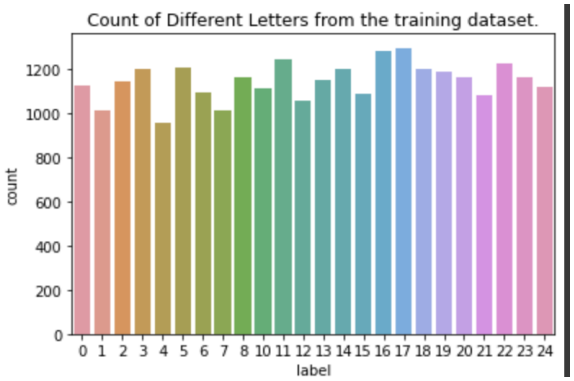
We are using the MNIST dataset, which is available in the public domain and is free to use. It contains pixel information for each of the 24 ASL letters over around 1000 photos. Additionally, we'll make use of the 27 Class Sign Language dataset, which was compiled from data on 173 people and includes fresh classes that represent expressions in American Sign Language.

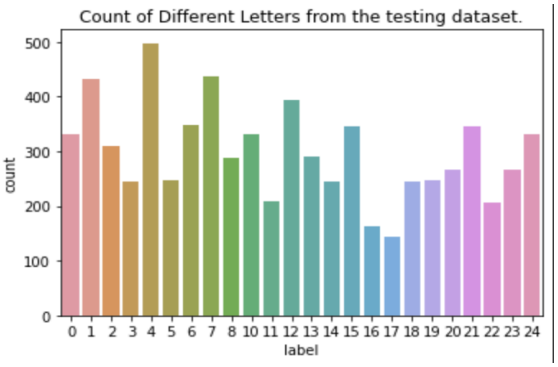
Methodology

Low Risk Methodology

The object detection is done using the YOLO and R-CNN models. We found Yolo model gave the best results.
The YOLO (You Only Look Once) model is a state-of-the-art object detection algorithm that can quickly and accurately detect objects in images and videos. In order to estimate the bounding boxes and class probabilities for each cell in the input picture or video, the YOLO model first divides the input into a grid of cells, and estimates box.

When contours are detected, We start to save the image of the ROI in the train and test set respectively for the letter or number we are detecting it for, this is later used for the GAN model. The saved image is converted into greyscale and flipped. For differentiating between the background we calculate the accumulated weighted avg for the background and then subtract this from the frames that contain some object in front of the background that can be distinguished as foreground. This is done by calculating the accumulated_weight for some frames (here for 60 frames) we calculate the accumulated_avg for the background. After we have the accumulated avg for the background, we subtract it from every frame that we read after 60 frames to find any object that covers the background.

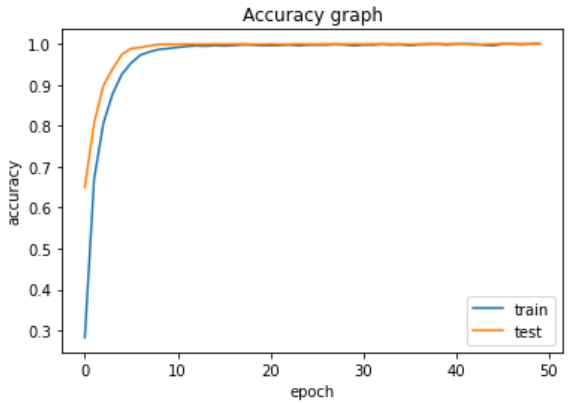




Medium Risk Methodology

The medium-level risk is the training and implementation of CNN and AlexNet models.
CNN is a type of deep learning algorithm that uses convolutional and pooling layers to extract features from grid-like data, and fully connected layers to perform the final classification or regression task. It is a feedforward neural network that is designed to process grid-like data such as an image, where each layer of the network learns a set of filters that are applied to the input data to extract increasingly complex features.

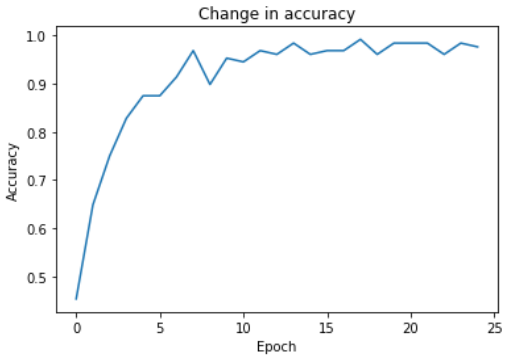
In training process callbacks of Reduce LR on plateau and earlystopping is used, and both of them are dependent on the validation dataset loss. After every epoch, the accuracy and loss are calculated using the validation dataset and if the validation loss is not decreasing, the LR of the model is reduced using the Reduce LR to prevent the model from overshooting the minima of loss and also we are using the earlystopping algorithm so that if the validation accuracy keeps on decreasing for some epochs then the training is stopped.
We used two different optimization algorithms used – SGD (stochastic gradient descent, that means the weights are updated at every training instance) and Adam (combination of Adagrad and RMSProp) is used. We found for the model SGD seemed to give higher accuracies. As we can see while training we found 99% training accuracy and validation accuracy of about 88%



AlexNet is a deep neural network architecture for image classification tasks, which means it predicts the object or class to which an image belongs after receiving it as input. It was first shown in 2012 and is composed of numerous layers that gradually learn to separate various elements from the input image. Several convolutional layers make up the network, and these layers apply filters to the input image to extract various features including edges, corners, and blobs. After that, a nonlinear activation function is applied to the output of each convolutional layer in order to introduce nonlinearity and increase the network's expressiveness. After a number of convolutional layers, the output is flattened and sent through a number of fully connected layers. These layers do classification by giving each potential class a probability score.

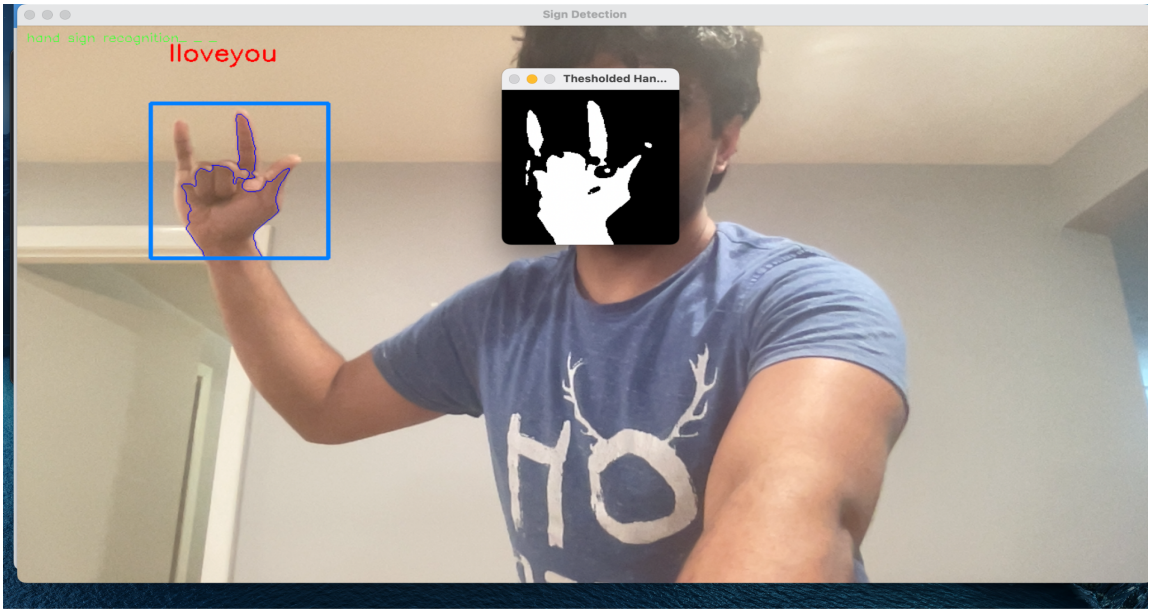
Since the ASL dataset may contain images of various sizes, AlexNet's input size is modified accordingly. AlexNet's final layer is intended to categorise photos into 1000 different groups. The number of classes in sign language recognition is often

substantially fewer, at 26 for each English alphabet letter. As a result, the final fully connected layer of the AlexNet architecture is swapped out for one that has the right amount of output neurons. By using methods like flipping, rotating, and scaling the photos, data augmentation can be used to fictitiously increase the size of the ASL dataset and enhance the model's resistance to fluctuations in the input images.



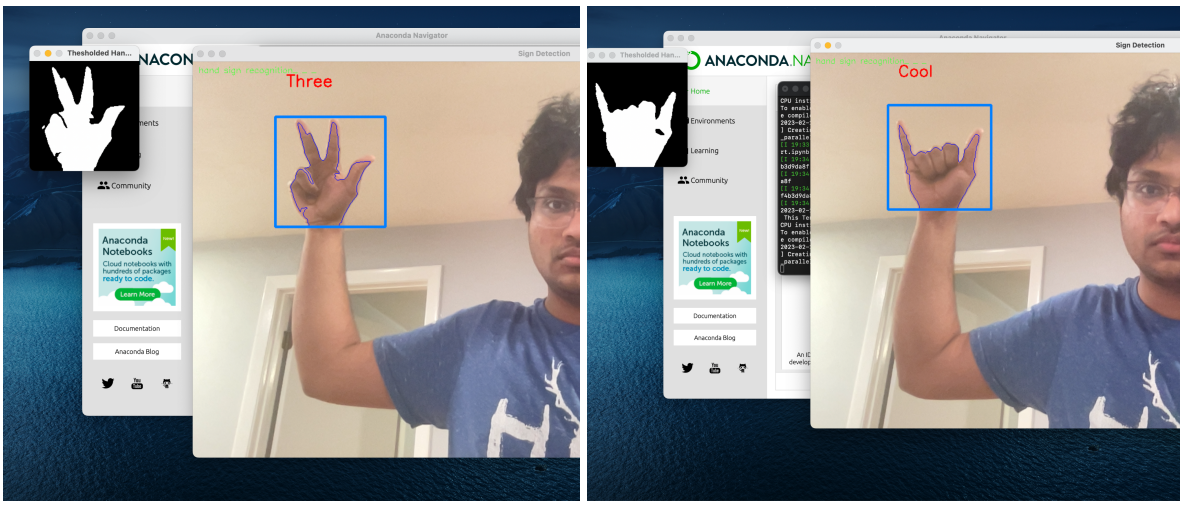
High Risk Methodology

Using generative adversarial networks and feedback mechanisms to construct an enriched dataset, the high-risk objective is to allow users to create unique ASL signs for custom expressions. We are using the Keras ImageGenerator for the same.The generator and discriminator are the two components that make up the GAN. The discriminator discriminates the generated image with regard to the original photos once the generator has been trained on training images. This creates a feedback loop between the generator and the discriminator until we attain the desired metrics. The dataset created by the GAN is then fed to the Deep learning classification models, and this classifier can now be used to classify these new signs. Below is an image of the example expression created. We created a symbol for the expression “Cool”. Using live video capture the model is able to classify the sign live. Similarly we experimented with many symbols like “I love you”, “Welcome” etc.. and our program can be used to create any custom expression need by the user.



Results

Model	Training Accuracy	Testing Accuracy	Testing F1 Score
CNN with MNIST	99.8	88.1	84.5
CNN with GAN data	96.25	84.11	82.25
AlexNet with MNIST	97.65	96.42	93.5
Alexnet with GAN data	94.65	86.01	80.4



Conclusion

In conclusion, the recognition of sign language is a difficult issue that calls for sophisticated computer vision and machine learning approaches. Even though this field has made tremendous strides recently, there is still much to be done to raise the reliability and accuracy of sign language recognition systems. Exploring more sophisticated deep learning models that can better capture the intricate spatial and temporal characteristics of sign language is one intriguing area for future research. The development of more efficient data augmentation techniques is a crucial field of research because it can be challenging to gather extensive and varied datasets on sign language.Recognition of sign language has the potential to significantly enhance communication and quality of life for deaf or hard of hearing people. This promise can be realised and made more accurate, dependable, and accessible by continued study in the field.

References

1. G. A. Rao, K. Syamala, P. V. V. Kishore and A. S. C. S. Sastry, "Deep convolutional neural networks for sign language recognition," 2018 Conference on Signal Processing And Communication Engineering Systems (SPACES), Vijayawada, India, 2018, pp. 194-197, doi: 10.1109/SPACES.2018.8316344.
2. K. Amrutha and P. Prabu, "ML Based Sign Language Recognition System," 2021 International Conference on Innovative Trends in Information Technology (ICITIIT), Kottayam, India, 2021, pp. 1-6, doi: 10.1109/ICITIIT51526.2021.9399594.
3. https://link.springer.com/article/10.1007/s13042-017-0705-5
4. https://link.springer.com/chapter/10.1007/978-0-85729-997-0_27.