FINAL REPORT

PROJECT NAME: PROJECT UN-MUTE.

CONTRIBUTORS:

ANSHUMAN DWIVEDI,

KIRTHI KRISHNAN GANESHAN,

PRASAD GANDOLE,

VIHAN PARMAR,

YASHRAJ TAMBE.

Contents

1.PROBLEM STATEMENT:

This project demonstrates the ability to bridge the communication gap between the autistic-mute individuals and regular individuals in order to bring about a meaningful social impact and foster a sense of unity amongst the disabled by using the power of Artificial Intelligence.

The aim is to develop an efficient system that can improve the lifestyle conditions for the autistic mute and hopefully contribute to the greater whole. Disability figures in India have risen about 30% in the previous decade, with about 20.3% of people having movement disabilities, 18.9% having hearing impairments and 18.8% having visual impairments. A major study published in 2018 of five sites in India found that 9.2% of children aged between 2-5 years and 13.6% of the children aged between 6-9 years had at least one of seven neurodevelopmental disorders (vision impairment, epilepsy, neuro motor impairment including cerebral palsy, hearing impairment, speech and language disorders, autism spectrum disorders and intellectual disability.) The number of impaired people has recently reached about 400 million and therefore extensive research and studies have been accelerated in order to ease the means for communication amongst the disabled.

The objective of this project is to improve the conditions that autistic individuals deal with on a day to day basis by using several Artificial Intelligence techniques and algorithms.

2. EXPLORATORY DATA ANALYSIS:

2.1 DATASET INFORMATION:

Dataset: American Sign Language Lexicon Video Dataset

***Developers*:**

1. Stan Sclaroff, Professor of Computer Science, Boston University
2. Carol Neidle, Professor of French and Linguistics, Boston University
3. Vassilis Athitsos, Associate Professor of Computer Science and Engineering, University of Texas, Arlington
4. J. Nash, A. Stefan, Q. Yuan and A. Thangali

2.2 DATASET DESCRIPTION:

Number of Images: 667 images

Number of Features: Forty distinct features (gestures) namely:

1. All-2

2. Accident-1

3. Adopt

4. Advise

5. Again

6. Airplane

7. Call on phone

8. Chicago

9. All_right

10. Answer-fr

11. Apple

12. Appointment

13. Art

14. Approximately

15. Article

16. Awful-2

17. Awkward

18. Atlanta

19. Baseball

20. Become

21. Bad

22.Behavior-n1

23. Baltimore

24. Bread

25. Beer

26. Breakfast

27. Bird

28. Busy-2

29. Blue

30. Buy

31. Board

32. Camping

33. Can-1

34. Boss-2

35. Boston

36. Center-1

37. Brown

38. Chat

39. California

40. Any

Images in training dataset: 551

Images in testing dataset: 116

Image Size: 640*480

Number of Gestures: 40 gestures (each expressing a word)

2.3 FINDINGS:

Sign language videos contained about 40 gestures out of which about approx. 10-15% were single hand gestures & rest were double handed gestures. Since this video dataset was based on a project in University it was taken under best environments & there is a uniform ambience present, which actually removes all the noise.

3. OVERVIEW OF THE FINAL PROCESS

3.1 PROBLEM SOLVING METHODOLOGY:

Step 1: Acquire Real Time American Sign Language Dataset through WebCam or camera.

Step 2: The Real time hand gesture movements recorded by the camera which is in a video format is further given to Key Frame Extraction Algorithm. This algorithm breaks the video into key frames, and these key frames are further passed on as input to the Mask RCNN model.

Step 3: Mask RCNN model takes the key frames generated in the previous step as input and given an output in form of an image which has three noticeable and important features:

a) A class label of the gesture in an image,

b) A bounding box around the gesture in that image

c) Possibility of an element wise segmentation on the basis of training the model to understand various objects inside the image. Here we have trained the model to understand the gestures indicated by the hands in order to enhance the accuracy in a real time environment under various lighting conditions and orientations.

3.2 SALIENT FEATURES OF THE DATA:

The data is in video (.mov) format, with each video of approx. 2 mins performing an average of 3 signs & a total of 4 videos were used to get the desired images to create a set of required gestures & the frames were extracted to a size of 640*480 px since the movements were distinctive enough. Furthermore, that would help in faster training procedure.

3.3 DATA PREPROCESSING STEPS:

The only data pre-processing step we have used is the Key Frame Extraction algorithm which generates key frames from a real time video that is either pre-recorded/ or is being recorded live on camera.

The following are the steps for Key Frame Extraction Algorithm:

Step 1:

For each video frame k = 1 to N

a. Read frame Vk and Vk+1
b. Obtain the gray level image for Vk and Vk+1
   - Gk = Gray image of Vk
   - Gk +1 = Gray image of Vk +1
c. Find the edge difference between Gk and Gk + 1 using the Canny edge detector.
   - Let diff(k) be their difference.

- dif(k) = summation of i,j (Gk - Gk + 1) where i, j are row and column index.

Step 2:

Compute the mean and standard deviation:

Step 3:

Compute the threshold value

- Threshold = M + a*S , where a = constant

Step 4:

Find the key frames for k = 1 to (N-1)

- If diff(k) > Threshold then,
- Write frame Vk +1 as the output key-frame.

3.4 ALGORITHMS USED BEFORE:

Skin Detection Algorithm: Skin Region Detection, Feature Extraction: K-Convex Hull Algorithm, Classification: Convolutional Neural Network.

3.4.1 DISADVANTAGE:

1. Skin Detection Algorithm is only as good as it is trained. It will not intelligently detect skin regions according to dynamic lighting conditions.
2. In K Convex Hull Algorithm, the recursion is slow and therefore there is overhead of the repeated subroutine calls.
3. K Convex Hull is unable to control and guarantee sub-problem size resulting in sub-optimum worst-case time performance.
4. K Convex Hull requires excessive memory for storing intermediate results of sub-convex hulls to be combined in order to form the complete convex hull.
5. The use of Divide and Conquer is not ideal if the points to be considered are too close to each other.

3.5 ALGORITHM USED NOW:

1. We have used the Key Frame extraction algorithm in our pre-processing step in order to extract the frame sequence from the videos to uniquely identify frames at the start of the execution process.
2. Mask RCNN algorithm is also used here. Here feature extraction, classification, instance-based segmentation is done in the single architecture.
    - Mask RCNN is a widely used Computer Vision algorithm for object detection providing maximum accuracy while at the same time solving the problem for vanishing gradients.

3.6 HOW WE COMBINED THE TECHNIQUES:

We used Key Frame Extraction algorithm in order to uniquely identify the frames from a video and process these frames as input to the Mask-RCNN model.

The Mask-RCNN model takes care of various steps inside its architecture all the way from extracting features in the first step, determining if that extracted feature consists of an object or not in the second step in order to ensure that only those features with objects in it are process for the next step, bring all the features consisting of objects to the same size in the third step by using region of interest pooling layer and further classifying the image with a bounding box in the fourth step, passing on the classified images along with their bounding boxes to intersection over union and mean average precision to extract even richer images and finally giving the rich extracted images to segmentation mask in order to person instance based segmentation.

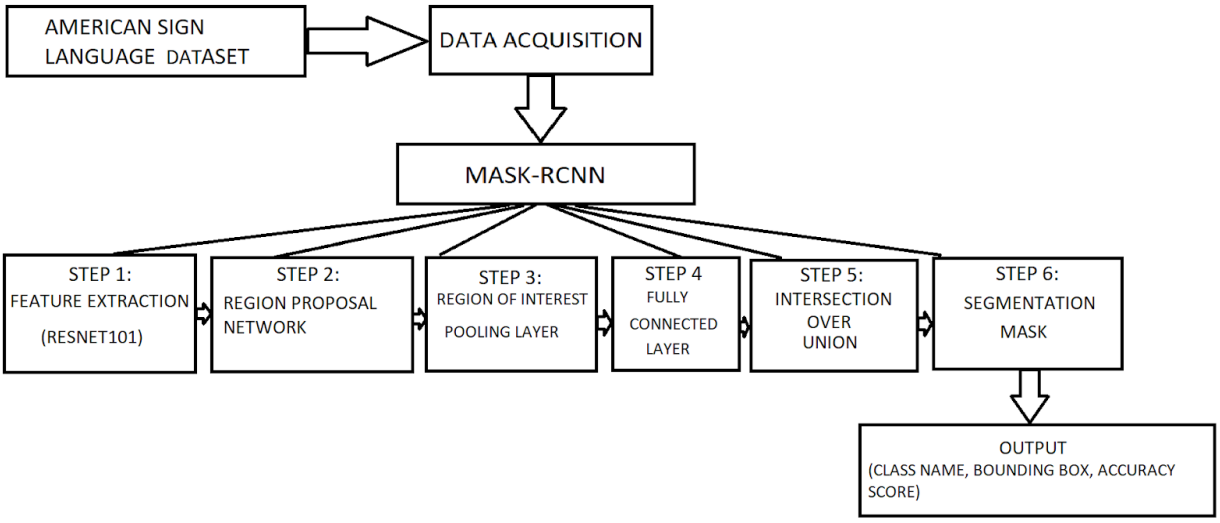
Our model gives a high accuracy output as Mask-RCNN algorithm goes through a series of steps in order to make sure that we not only have a classification of the object of interest but also give an instance-based segmentation of the image.

The fact that we used the Key frame extraction method in pre-processing helps us segregate only the frames of interest and it helps us speed up the process of frame retrieval. Key Frame extraction plays a pivotal role in order to process real time videos which challenges the algorithms we have laid out fairly.

4. STEP BY STEP WALK THROUGH OF THE SOLUTION

4.1 BASE MODEL AND ARCHITECTURE (PROJECT EXECUTION WORKFLOW):

MODEL ARCHITECTURE:



The project workflow is as follows:

STEP 1: ***DATA ACQUISITION***

There are two ways by which data can be formulated:

1. Using an existing dataset of sign languages.

2. Creating an original dataset with multiple gestures. This flexibility can help developers utilize local sign languages as they differ from country to country.

3. Here we are using a dataset created by Suresh Kumar where there are five distinct gestures, and each gesture has multiple images in different orientations and locations to train and test it.

Data acquisition is implemented by the following approach:

Hand gesture is taken as an I/P by a USB camera or webcam.

STAGE 2: ***KEY FRAME EXTRACTION ALGORITHM***

We have used the Key Frame extraction algorithm in our pre-processing step in order to extract the frame sequence from the videos to uniquely identify frames at the start of the execution process. The data that we are providing is in the form of a video format. In order for our model to understand the video significantly, we use the Key Frame Extraction algorithm. The key frame extraction is a set of discrete frames which contain the most important information in the video. It can greatly reduce the time of video

retrieval by extracting key frames. Key Frame extraction plays a pivotal role in order to process real time videos which challenges the algorithms we have laid out fairly. The Key Frame extraction algorithm will divide the video into discrete frames and obtain its gray level image. Once the gray level images are obtained, it will compute the difference between the present frame and the consecutive frame using Canny Edge Detector. It follows a mathematical process where it uses a threshold value and extracts the key frames based on the difference computed between each of the frames (current and consecutive).

STEP 3: MASK-RCNN

Mask RCNN algorithm is also used here. Here feature extraction, classification, instance-based segmentation is done in the single architecture. Mask RCNN is a widely used Computer Vision algorithm for object detection providing maximum accuracy while at the same time solving the problem for vanishing gradients.

4.2 FOLLOWING ARE THE STAGES FOR MASK-RCNN MODEL:

STEP 1: *FEATURE EXTRACTION:*

ResNet (Residual Networks), a classic neural network algorithm used as a backbone for our project.

This model has been the winner of the ImageNet challenge held in 2015.

The fundamental breakthrough with ResNet was, it allowed training on extremely deep neural networks while being able to solve the problem of vanishing gradients at the same time.

Here we will be using ResNet101 architecture.

LIBRARIES USED FOR RESNET101: *Keras*

HOW IS RESNET101 CONVERTED INTO CODE:

The Resnet101 is a deep neural architecture with 101 blocks in it.

```
171 > def resnet_graph(input_image, architecture, stage5=False, train_bn=True):
207       ......
208       ......
```

Each block contains convolution layer, activation layer as Relu, again a convolution with a skip connection

Skip connection adds input directly to the output of the previous convolution layer and it is commonly known as *'Residual Block'*.

ResNet solves the problem of vanishing gradients by using Skip connections in its architecture.

Stacking of 101 residual blocks/layers creates ResNet101.

Using Python and Keras library the Resnet101 architecture which was readily available in pre-trained mode was implemented for creation of the model.

STEP 2: *REGION PROPOSAL NETWORK*

Region Proposal Network is basically used to determine whether a specific region has an object or not.

```
def build_rpn_model(anchor_stride, anchors_per_location, depth):
    .....
    .....
```

If it does not find an object in certain features, it eliminates those features and considers only the features which have an object in it.

Region Proposal Network is used to return the candidate bounding boxes.

STEP 3: *REGION OF INTEREST*

Region of Interest is found in order to extract even more richness to the existing image. Region of interest determines how an image can contain a specific region with more information in them.

```
class PyramidROIAlign(KE.Layer):
        .....
        .....
```

The regions obtained from RPN will be of different sizes, hence by using a pooling layer we bring them to the same size.

STEP 4: *FULLY CONNECTED LAYER:*

Now these regions are forwarded to a Fully Connected layer to predict the binding boxes and class labels

```
mrcnn_class_logits = KL.TimeDistributed(KL.Dense(num_classes),##Detecting the class
x = KL.TimeDistributed(KL.Dense(num_classes * 4, activation='linear'),#Detecting the bounding box
```

STEP 5: *INTERSECTION OVER UNION*

For all the predicted regions, we compute the intersection/union (intersection over union). Condition: If IoU is >= 0.5, then consider the region as ROI & if not, then exclude the region.

```
def overlaps_graph(boxes1, boxes2):
    ....
    ....
```

STEP 6: *SEGMENTATION MASK*

Once the ROI is found wrt the IoU values, we must add a mask branch to existing architecture. This will give the segmentation mask for each region that contains an object (element wise segmentation).

```
def build_fpn_mask_graph(rois, feature_maps, image_meta,
    .....
    .....
```

It also returns a mask of size 28*28 for each region which is then scaled up for inference. This is the final step, and it will give predictions of the images.

-

5. MODEL EVALUATION

Prominent Parameters:

```
Configurations:
BACKBONE                       resnet101
BACKBONE_STRIDES               [4, 8, 16, 32, 64]
BATCH_SIZE                     1
BBOX_STD_DEV                   [0.1 0.1 0.2 0.2]
COMPUTE_BACKBONE_SHAPE         None
DETECTION_MAX_INSTANCES        100
DETECTION_MIN_CONFIDENCE       0.9
DETECTION_NMS_THRESHOLD        0.3
FPN_CLASSIF_FC_LAYERS_SIZE     1024
GPU_COUNT                      1
GRADIENT_CLIP_NORM             5.0
IMAGES_PER_GPU                 1
IMAGE_CHANNEL_COUNT            3
IMAGE_MAX_DIM                  1024
IMAGE_META_SIZE                53
IMAGE_MIN_DIM                  800
IMAGE_MIN_SCALE                0
IMAGE_RESIZE_MODE              square
IMAGE_SHAPE                    [1024 1024    3]
LEARNING_MOMENTUM              0.9
LEARNING_RATE                  0.0001
LOSS_WEIGHTS                   {'rpn_class_loss': 1.0, 'rpn_bbox_loss': 1.0, 'mrcnn_class_loss': 1.0, 'mrcnn_bbox_loss': 1.0, 'mrcnn_mask_loss': 1.0}
MASK_POOL_SIZE                 14
MASK_SHAPE                     [28, 28]
MAX_GT_INSTANCES               100
MEAN_PIXEL                     [123.7 116.8 103.9]
MINI_MASK_SHAPE                (56, 56)
NAME                           ASL
NUM_CLASSES                    41
POOL_SIZE                      7
POST_NMS_ROIS_INFERENCE        1000
POST_NMS_ROIS_TRAINING         2000
PRE_NMS_LIMIT                  6000
ROI_POSITIVE_RATIO             0.33
RPN_ANCHOR_RATIOS              [0.5, 1, 2]
RPN_ANCHOR_SCALES              (32, 64, 128, 256, 512)
RPN_ANCHOR_STRIDE              1
RPN_BBOX_STD_DEV               [0.1 0.1 0.2 0.2]
RPN_NMS_THRESHOLD              0.7
RPN_TRAIN_ANCHORS_PER_IMAGE    256
STEPS_PER_EPOCH                100
TOP_DOWN_PYRAMID_SIZE          256
TRAIN_BN                       False
TRAIN_ROIS_PER_IMAGE           200
USE_MINI_MASK                  True
USE_RPN_ROIS                   True
VALIDATION_STEPS               50
WEIGHT_DECAY                   0.001
```

The evaluation of the model is being done by using mAP (Mean Average Precision). Since the Mask-RCNN model is a highly unlikely usual model that either performs either a classification task or regression task wherein we've established ways of determining the model's prowess for the solution it built. Mask-RCNN works on both classification which is used for classifying to a specific class as well as regression which is used for generating the box regions of which a major part is played in localizing which is why the purpose cannot accept the usual rules.

mAP was majorly coined from the COCO & Pascal VOC challenges although both differed in terms of calculation. So, the approach is to rank the recall with the one having the highest confidence at the topmost & the precision & recall is calculated which when plotted give precision-recall curve. AP simply means calculating the area under this P-R curve. But the graph is pretty zigzag & hence at each recall level, we replace each

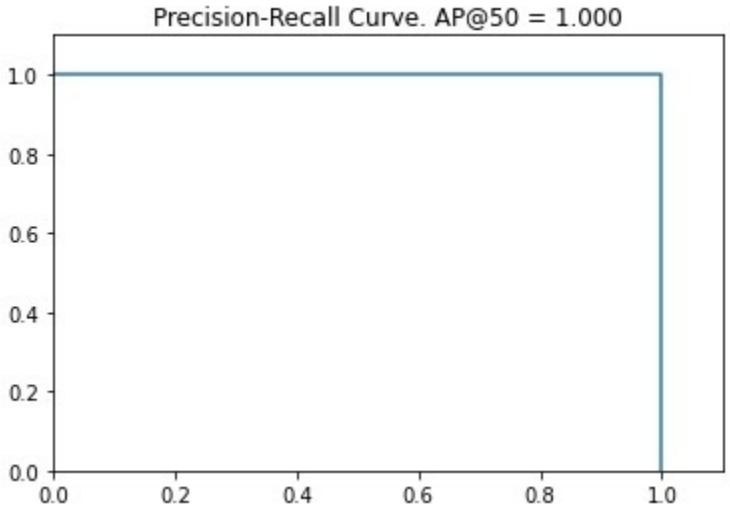
precision value that is at the next higher recall value having the highest precision. Such smoothened areas are divided in blocks & the area is averaged over the no. of recall received & hence we get AP. This AP when averaged at multiple IOU's gives mAP.

Model was being built on a Resnet-101 architecture which is alone one of the state-of-the-art model with plus Mask-RCNN already was selected due to its higher accuracy. The dataset used was developed on a controlled recording environment hence removing the noises which would help the model in realising the exact gesture features. Plus the model was feed on a variety of augments as well to add changes which will provide model with an extra disturbed environment data as to increase the dataset for the model as well to increase the feature extraction capability.

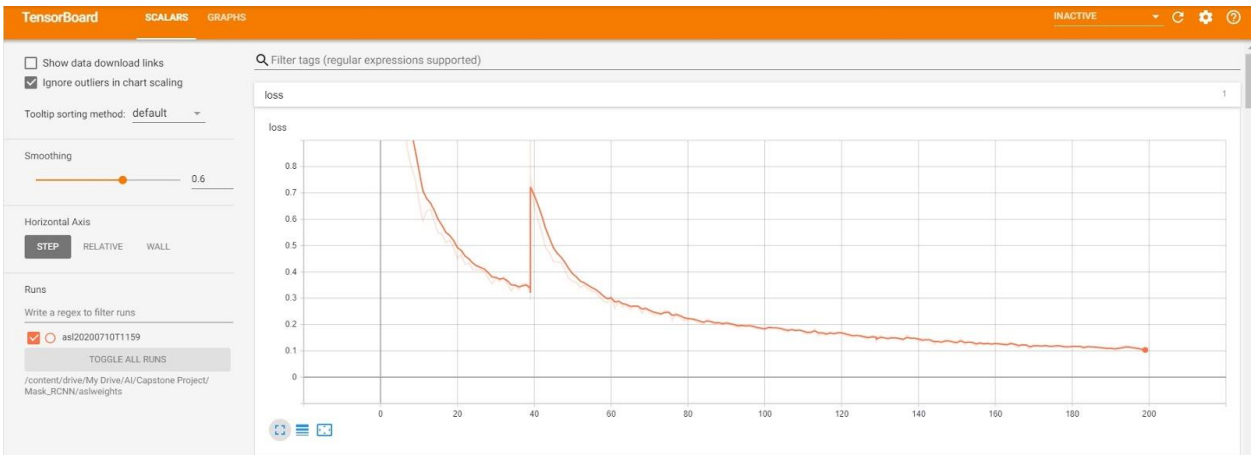
6. VISUALIZATION(S)

6.1 Validation Results (Bounding Box with semantic segmentation and confidence of prediction):

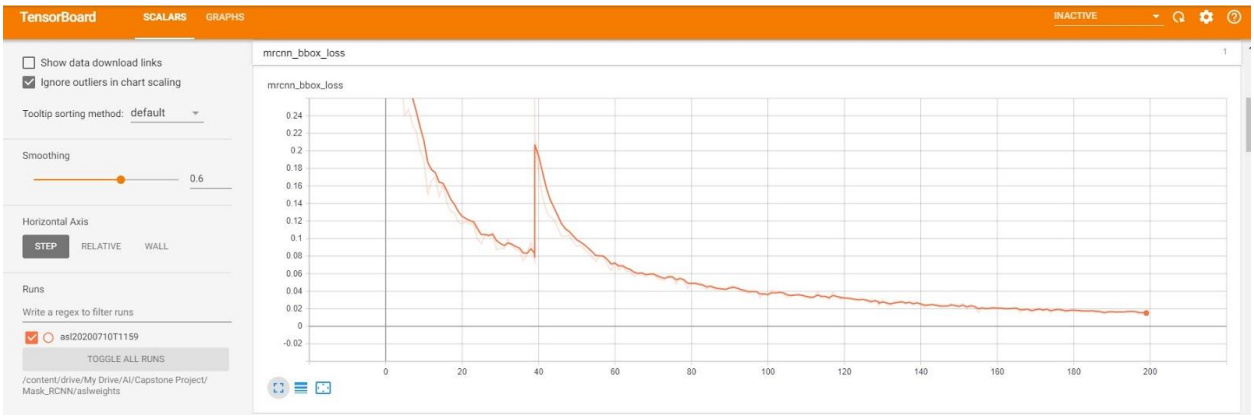
6.2 P-R CURVE FOR EACH GESTURE (Precision-Recall Curve):



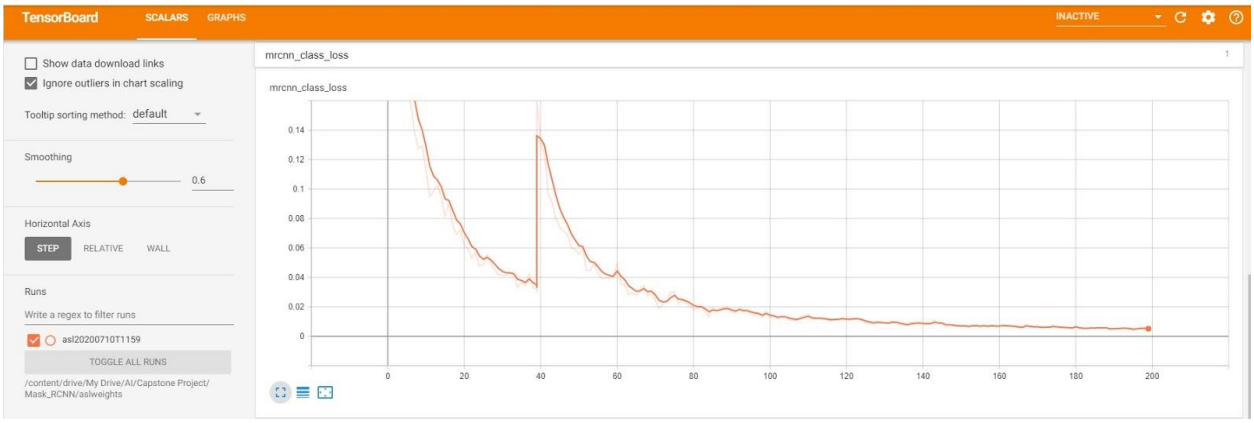
6.3 Loss (Loss which is a combination of all the losses occurred during training):



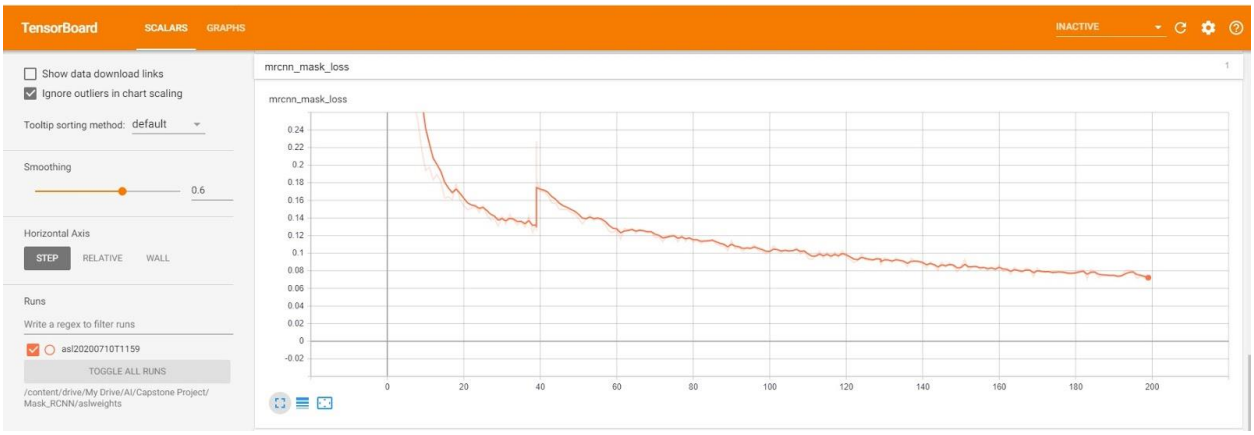
6.4 MRCNN BBOX Loss (The bounding box loss while training):



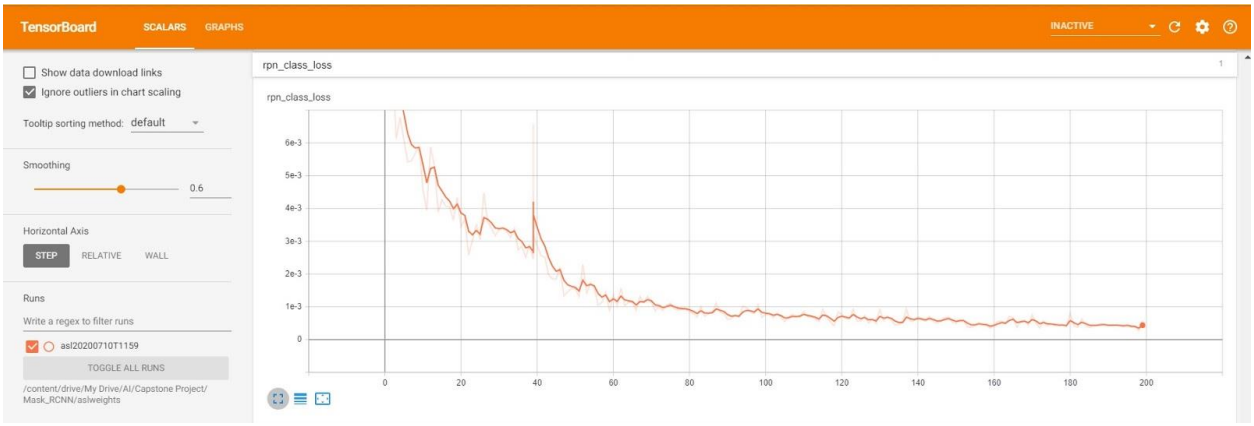
6.5 MRCNN Class Loss (The classification loss during training):



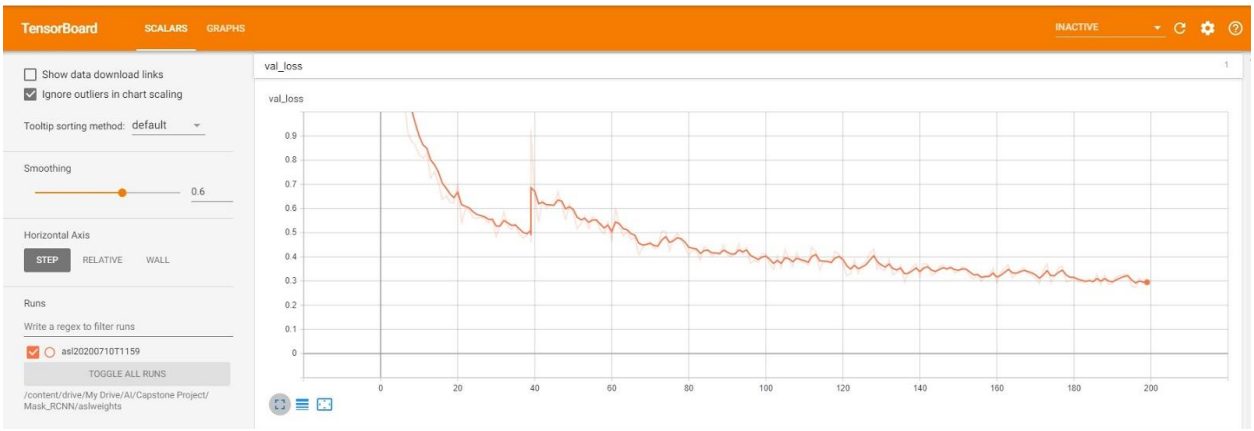
6.6 RPN BBox Loss (RPN bounding box loss graph for training):



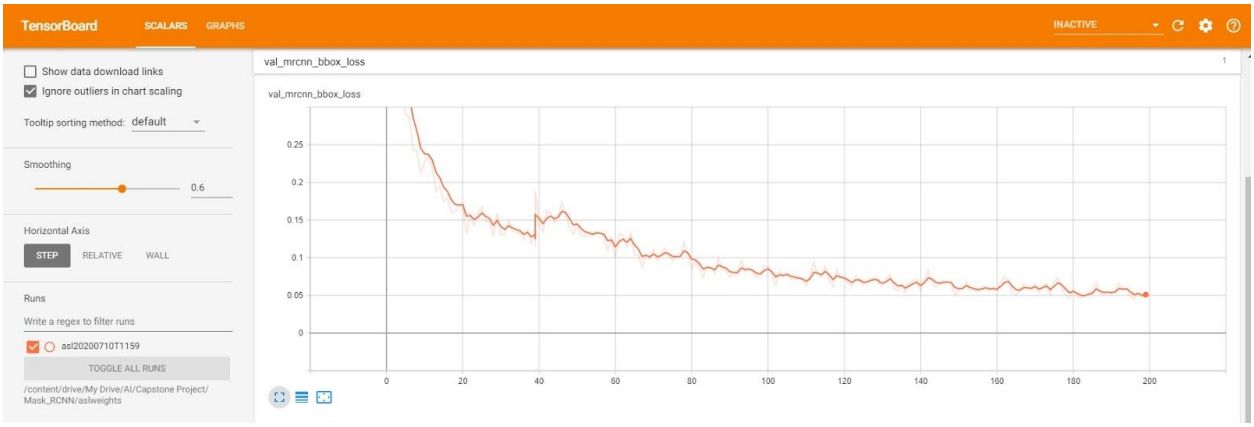
6.7 RPN Class Loss (RPN anchor classifier loss while training):



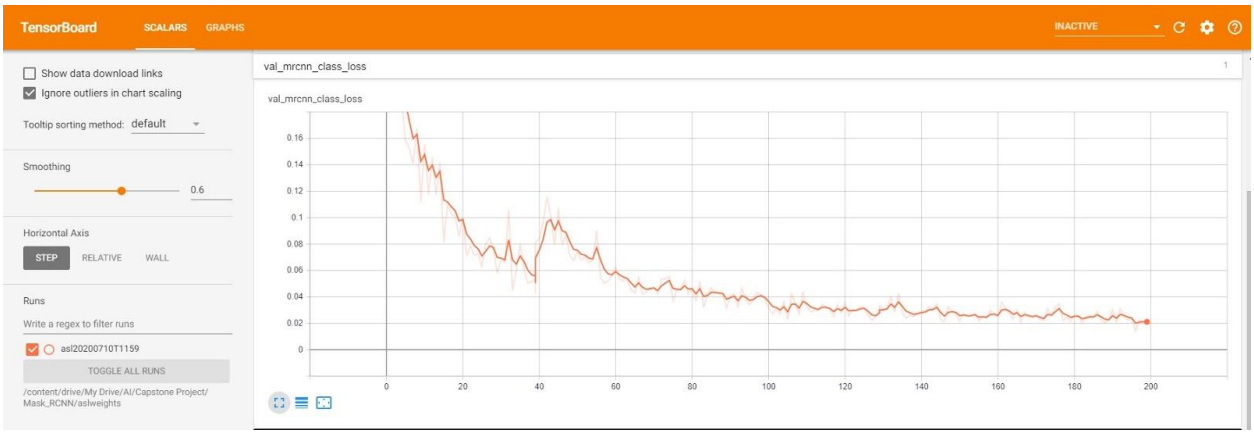
6.8 Val Loss (Validation Loss which is again a combination of all the losses occurred during validation):



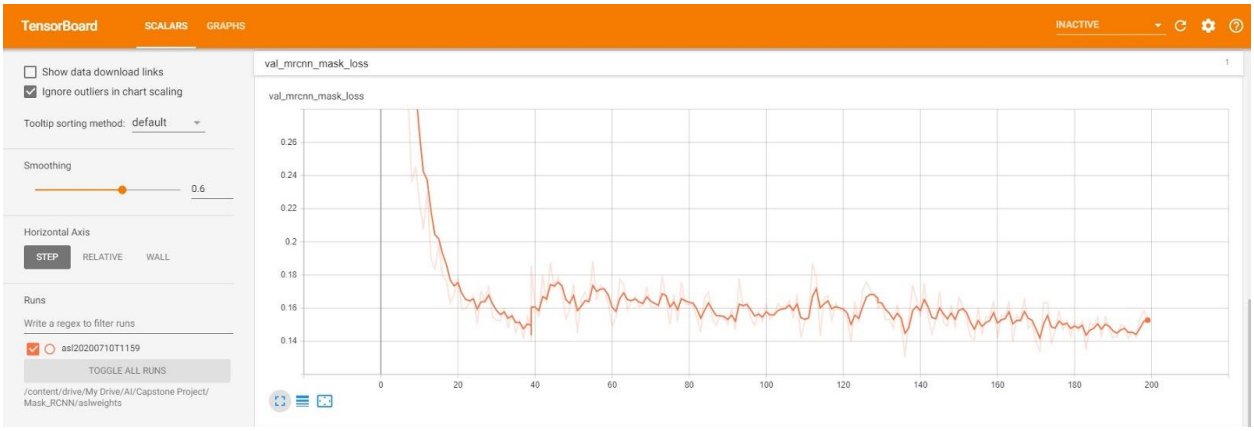
6.9 Val MRCNN BBox Loss (Bounding box Loss during validation):



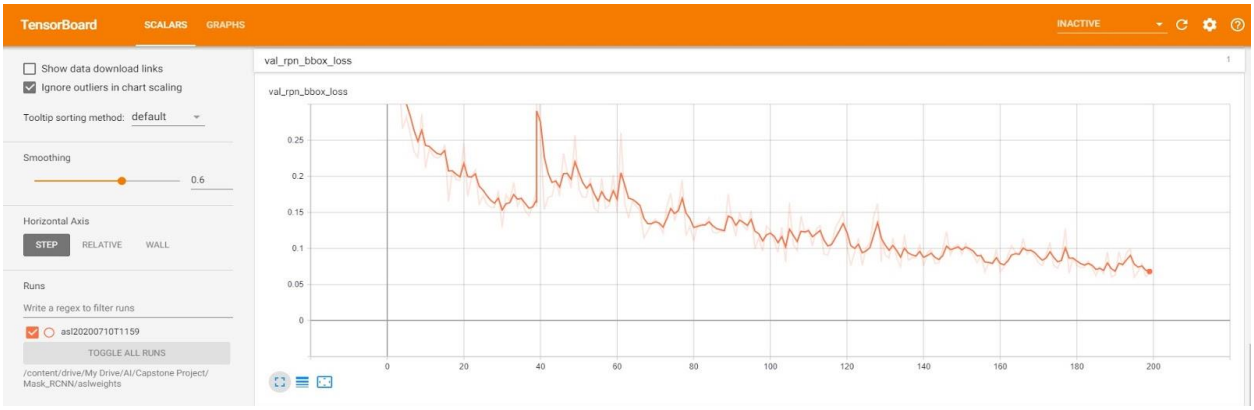
6.10 Val MRCNN Class Loss (The Classification loss during validation):



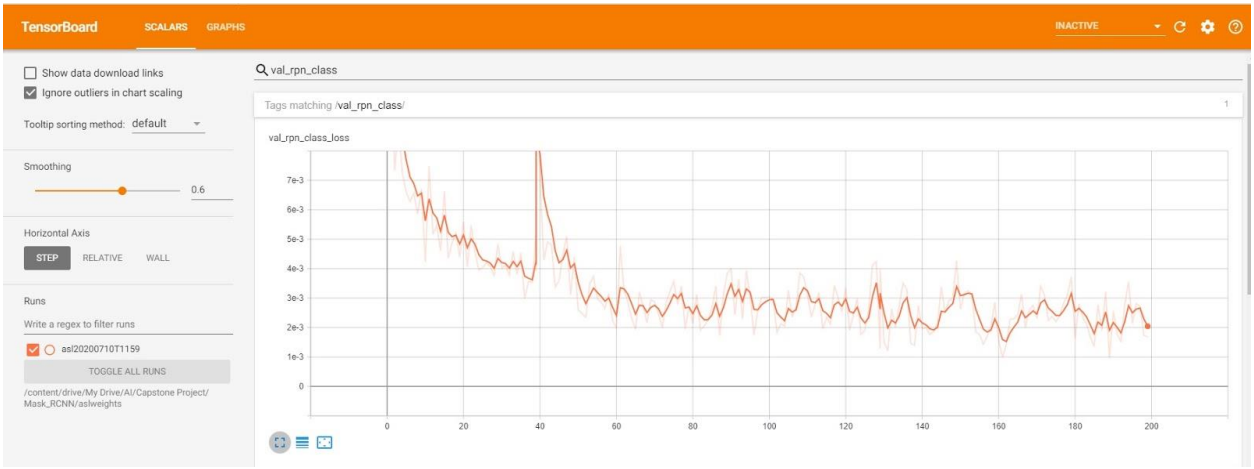
6.11 Val MRCNN Mask Loss (The Mask Loss during Validation):



6.12 Val RPN BBOX Loss (The RPN network's bounding box loss):



6.13 Val RPN Class Loss (The Validation RPN anchor classifier Loss):



6.14 FINAL TESTING RESULTS:

7. CHALLENGES:

1. Data Acquisition was a major problem in the initial stages of development as it was difficult to extract frames from the dataset which would later be fed to Mask RCNN.
2. Data Preparation was an issue since the data wasn't in the correct format for feeding the model. There was a concern because some gestures such as funny which use a subsequence of gestures, if that was to be fed to Mask RCNN, it would classify those sub-sequences of a single gesture as different.
3. Training the model is computationally intensive because of the nature of the model and the architecture which the model follows. It was challenging to utilize the resources efficiently in order to train the model.
4. The model is only as good as it is trained. The more it is trained with various environments, the smarter our model gets. Although this approach might be sound, our model can get computationally very expensive in order to encapsulate various environments under which it is trained.

8. IMPLICATIONS

- With the advent in Artificial Intelligence and its supporting algorithms, it is getting easier for humans to make a shift from traditional practices to technically sound approaches. The objective of this project is to bridge the communication gap amongst Autistic-mute individuals. With the help of Artificial Intelligence, we are able to achieve that objective with highly accurate results.

- Two important things are achieved by this process: It exempts the effort of learning sign language at the first place, which greatly helps regular individuals communicate with the Autistic-mute, second- a wide range of opportunities open for Autistic-mute individuals which are nearing about 400 million across the globe. The fact that there is a huge number of people suffering from such setbacks not only acts as a communication bridge but also serves as a moral high ground. This solution can be widely used by NGOs and other related divisions which look forward to contributing to the overlooked aspects of society bringing equality amongst people and serving as a booster for everyone who deserves to put forth their ideas and life.

- In the world of Computer vision, object detection is one of the most researched and explored topics. Our problem statement also revolves around the same old school Object Detection functionality, while attempting to create a channel for people to decipher and understand the sign language which is used by Autistic mute people as their mode of communication.

- As the signs vary across languages, our project takes into consideration the most widely used sign language dataset- American Sign language dataset. The proposed solution to

the problem defined, captures the essence of identifying gestures and creates a tag around it which will help the users to understand what the other person is trying to communicate. It is a modification over the previous attempts by researchers as the previous work was largely done on Alphabets in the American sign language, but going one step further, we have attempted to capture gestures pertaining to actions rather than letters as it will help build sentences.

- Also, the algorithm used does not use skin colour or light to identify the gesture but directly learns the gestures using the features trained which makes it independent of the race/physique and orientation of the person, which we think is a major advantage of our project.

- It makes use of the Masked R-CNN algorithm at the back-end which is a computationally expensive algorithm, it gives us accurate identification of gestures which gives us an edge over a wide variety of algorithms.

- In relation to the Computer Vision domain this should surely help us bridge the communication gap between differently abled and normal humans. A broader understanding of all the Sign languages which is different in different countries will help us improve similar projects.

- Our model gives high accuracy and it presents a high possibility of being successfully augmented to be an end to end solution.

9. LIMITATIONS

- There are limitations with the GPU while training the model. The computation can be even more expensive considering the high end resolution which the world is routing towards - 4K resolution.

- In the current proposed solution, even though the purpose is served, there is a huge scope of improvement. The back-end algorithm used – Masked R-CNN has various internal proceedings making it a computationally intensive algorithm. This in turn, makes the project highly dependent on heavy computing resources like heavy processing of GPUs and also limits us in implementing the model on Android based systems.

- Since the project does not serve the masses, getting investors to up-scale the project can get tricky.

- Maintenance of the project in terms of regular usage of the GPUs can be very difficult. With a wide number of languages that encompass the sign languages, the total number of gestures will increase and running a wide range of gestures can serve to be a challenging process considering the complexity of the algorithm. We need to either find an environment which allows us unlimited GPU resources which can only be achieved by paying for the resources. VMWares give us sufficient access to resources but even that has certain limits.

- The code in Tensorflow 1.x is extensive, which requires us to create multiple modules for every function adding to the existing complexity of the algorithm. Functions that can be utilized by entering a single command actually take a vast number of coding capabilities to be created and computed, which adds to the overall complexity of the project.

- The proposed model falls short for the real-time application in terms of giving bounding boxes, class labels and instance-based segmentation in the real-time video. The model decodings are shown in individual images rather than a real time video. This is because colab has deprecated the use of cv2 functions for some major processing & output displaying functions required.

- Given the gestures and signs involving two different hands with their relative movement to communicate, the model is not able to classify accurately as to which gesture it is where two hands are involved. It requires additional training of gestures which involves two hands which is time consuming.

9.1 WHAT CAN WE DO TO ENHANCE IT?

- In order to promote the developed model to the next level, we can train the model on a Mobile NET architecture with the learnt features from the current model. This will significantly decrease the size of the model which in turn will allow us to design a Mobile app around it.

- Using multiple GPUs would be a good way of dealing with this project or wait for a high-end cost-effective GPU (which are basically not that far since NVIDIA is gearing) that basically serve the high-resolution videos.

- Using Jupyter notebook with a computer providing a good GPU of at least 16GB with a HDD space of 512GB (since a longer training time) for this project should suffice the need.

- Over and above what we have currently, a bi-directional LSTM model can be built to frame multiple gestures into a grammatically correct sentence which then can be converted to speech and in another form as per the customer requirement – this will increase the business scope of the model and its application.

- Considering various competitors in the market, the project needs to be lightly threaded, efficient enough to be used across a range of devices, with an online as well as offline functionality. Therefore, creating a functionally diverse solution for the project problem and deploying it at a larger scale

- Migrating to Tensorflow 2.x can serve to be a very useful solution considering the computational complexity at hand. This can significantly reduce extensive code to mere wholesome functions.

- If the solution is offered in form of an application or a browser plugin, we can probably launch the functionality of the model on an application which can be used globally.

10. CLOSING REFLECTIONS

- Computer vision is one of the major breakthroughs in implementing/mimicking human visualization power to machines thereby enabling machines to have ’eyes’ so direct human assistance would be reduced

- Although the field is vast and the applications are changing, with different problems these techniques in various ways help with efficient solutions. Here the purpose of the project is to establish direct communication with differently-abled and normal humans is satisfied successfully.

- With a state-of-the art object detection enabled with the advanced “semantic segmentation” the Mask-RCNN is a major breakthrough process that helps to find the gestures with a greater accuracy on detection of objects. An attempt for implementing the same model using Mobile Net should surely be considered.

- A better tool/model/algorithm is required for automating the frame annotation process, which if fed with initial geometry of a target object, can decide what possible shapes the target object can have.

- If this 'new annotations procedure' if fed with geometry of a hand, i.e. five fingers, gaps (hull) between the fingers, relation between their lengths/breadth and with that of palm, then the auto-annotation will correctly contour the hands in any posture and can give additional output of hand rotation angles which serve as additional input feature.

- The model can be lightly threaded by using Tensorflow 2.x and object detection in it. The model can be put to test with various other algorithms such as YOLO (You Only Look Once) which are comparatively faster and require less computational resources as compared to Mask-RCNN.

- Finding a platform that gives us unlimited access to computational resources is a must in computer vision projects which take multiple factors into consideration making the computation heavy despite the algorithms we use

- As many people across the world do not have access to the internet, there has to be a solution which integrates a full end to end product. Combining a task specific software which can have room for various other functionalities in-built onto a well-designed, and a user-friendly hardware that can work even without internet connection. This will serve to target those who don't have the internet and bring a level playing field for them to communicate to the wider audience.