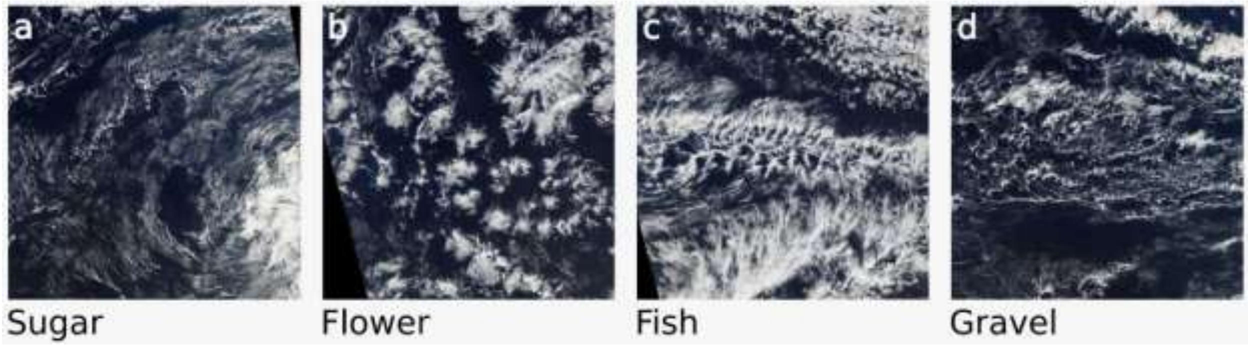
Understanding Clouds from Satellite Images

- **Course Details**
  - Course Name: Topics in Data Analytics
  - Course ID: CISC-839
  - Course Instructor: Hazem Abbas

- **Group Members**
  - Karthik Ranga Swamy: 20188884
  - Abhishek Awasthi: 20143052
  - Dhanush Reddy Nandyala: 20134832

❖ PROBLEM DESCRIPTION & MOTIVATION

Climate change has been at the top of our minds and there are many ways in which clouds can organize, but the boundaries between different forms of organization are not fully explained. This makes it challenging to build traditional rule-based algorithms to separate cloud features to better understand the clouds. Shallow clouds play a huge role in determining the Earth's climate and they are difficult to understand and to represent in climate models. In this Project we will be identifying regions in satellite images that contain certain cloud formations shown below, with label names: Fish, Flower, Gravel and Sugar. The segment for each cloud formation label for an image is encoded into a single row, even if there are several non-contiguous areas of the same formation in an image. Each image has at least one cloud formation, and can possibly contain up to all four. Stating the above we can consider this as a multiclass segmentation task which is finding 4 different patterns in the images. Since, we make predictions for each pair of image and segmentation mask(label) separately, this could be treated as a 4 binary segmentation problem. In order to solve this task, we decided to implement two deep learning segmentation models Mask Region based Convolution neural network(Mask-RCNN) and UNET.



❖ CHALLENGES FACED:

- ➢ Selection of the Deep learning model was the biggest challenge we faced as none of us had any prior experience in image segmentation. We wanted to implement a powerful model with good documentation and also with a reasonable training time but most of the state-of the-art models required a lot of training time and had poor documentation as a result we to compromise among performance or training time in selecting the segmentation model.
- ➢ Lack of computation power was another challenge as training images are computationally expensive and require a large amount of time. (On an average it took 90 min for one epoch for Mask-RCNN)
- ➢ Data structuring for selected ANN Models and corresponding Backbones proved to be difficult task as each model and implemented backbone requires custom data generator classes.
- ➢ One of the biggest challenges was to train and predict masks which were overlapping each other shown in the figure-1 and masks which contained a lot of background noise. In instance segmentation each pixel is classified and given a class and when masks overlap each other it is very hard for the model to classify these pixels in the overlapped masks.
- ➢ Removing the black strip in the images was also difficult. We were using various stitching algorithms to remove the black strip and stich the images but in doing so we realized that some of the mask in the form of encoded pixels contained this black strip.

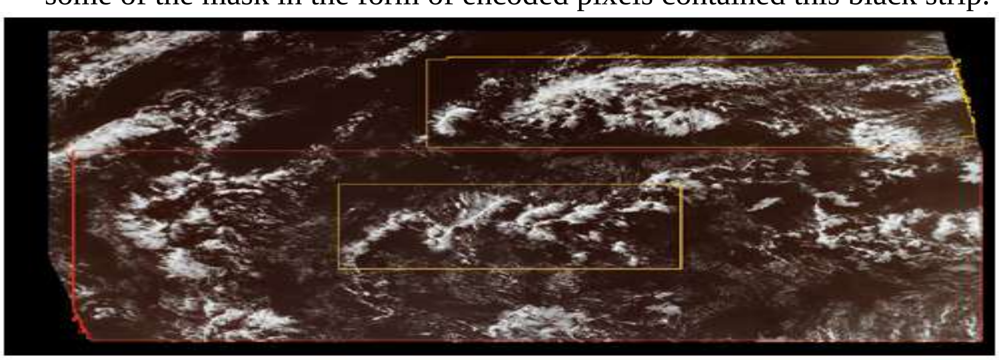


Figure 1: Flow chart for the Solution Approach

❖ METHODOLOGY

All the below steps shown in Figure-2 were implemented in Python Programming language using Keras Deep learning library, Matterport Mask-RCNN implementation[8] and Segmentation Models[7]on a Nvidia 1070 GTX GPU and Google Colab kernels.

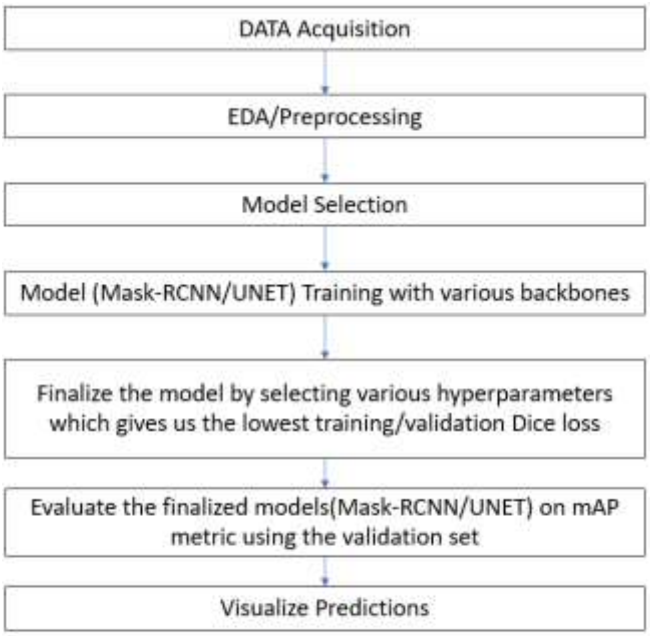


Figure 2: Flow chart for the Solution Approach

1. Data Acquisition

Data set has been taken from kaggle competition "Understanding Clouds from Satellite Images" which is acquired by NASA Worldwide. Data set can be found https://www.kaggle.com/c/understanding_cloud_organization/data .

2. EDA and Preprocessing

➢ The experimental data considered in this project consists of a 'train_images' folder which has the 5546 training images with size 2100*1400 and a train.csv file containing the run length encoded segmentations for each image-label pair in the 'train_images'.

➢ Taking a closer look at the train.csv shown in figure-3 file we can see that:

- For each image from the training dataset there are 4 lines for each type of clouds.
- 'Image_Label' is a concatenation of the image filename and a cloud type.
- If a certain type of clouds in present on the image, the 'EncodedPixels' column is non-null and contains the segmentation map for the corresponding cloud type.

Index	Image_Label	EncodedPixels
0	0011165.jpg_Fish	264918 937 266318 937 267718 937 269118 937 270518 937 271918 937 273318 937 274718 937 276118 937 277518 937 278918 937 280318 937 281718 937 283118 937 284518 937 285918 937 287318 937 288718 937 290118 937 291518 937 292918 937 294318 937 295718 937 2…
1	0011165.jpg_Flower	1355565 1002 1356965 1002 1358365 1002 1359765 1002 1361165 1002 1362565 1002 1363965 1002 1365365 1002 1366765 1002 1368165 1002 1369565 1002 1370965 1002 1372365 1002 1373765 1002 1375165 1002 1376565 1002 1377965 1002 1379365 1002 1380765 1002 1382…
2	0011165.jpg_Gravel	nan
3	0011165.jpg_Sugar	nan

Figure 3: contents of train.csv file.

- We have total 22184 Image labels (segmentation masks) and while looking for null values shown in figure-4,we observed that 10348 rows don't have encoded pixels i.e. they have empty segmentation maps, as a result we have deleted those data entries and will be working with 11836 image labels.

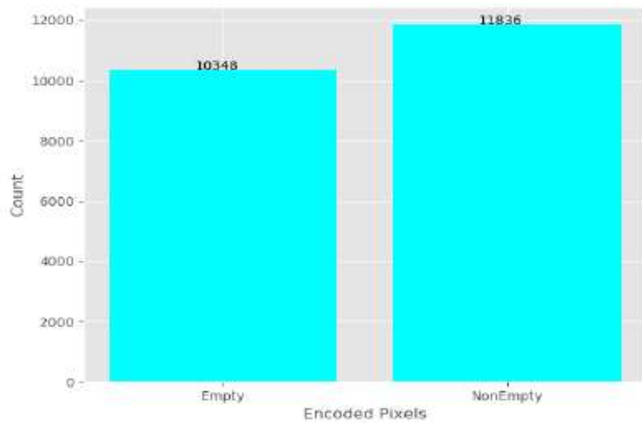


Figure 4: Count plot showing empty and non-empty segmentation masks.

- Counting the number of labels of each cloud type (Figure-5) we found that there are 2781 of Fish, 2365 of Flower, 2939 of Gravel and 3751 of Sugar observations respectively. From the plots below we can see that the dataset is somewhat balanced making the task a bit easier.

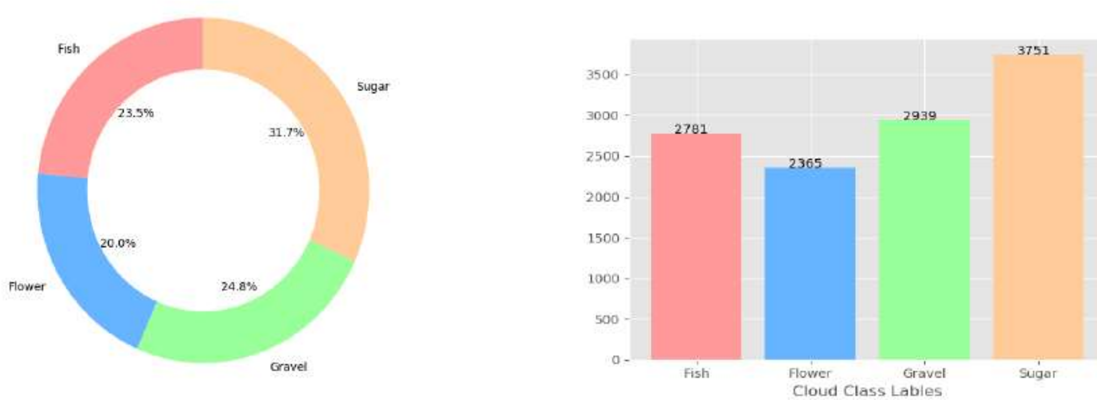


Figure 5: Donut plot(left) and count plot(right) showing the Cloud class distribution in the dataset.

- We wanted to see the number of mask labels per image and observed that 2372 Images which has two classes, 1560 Images with three classes, 266 images with all the four classes and 1348 with only one mask. We can conclude from figure-6 that most of the times we have 2 or 3 types of cloud formations in one image, all the 4 types of cloud formation in one image is very rare. Only one type of cloud formation in the image is also somewhat common

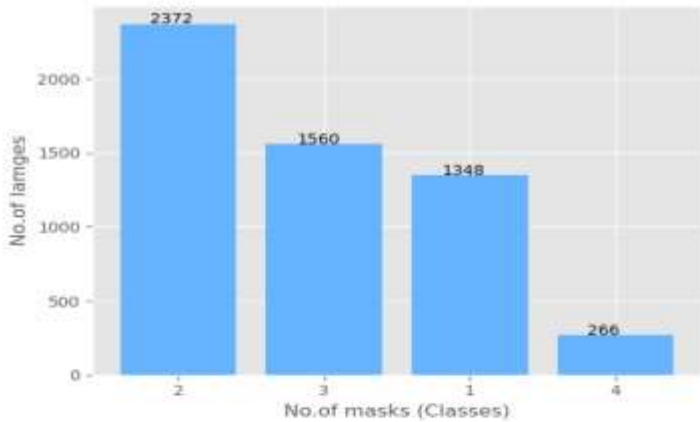


Figure 6: Bar plot showing frequency of labels per image.

- We also wanted to see which combination of cloud formations which occur frequently and combinations that hardly appears together. To do this we used a simple datamining algorithm called Frequent Patter Mining and plotted the results shown in Figure-7. We can state from the figure below that Sugar cloud formation frequently appears together in the images with Gravel or Fish cloud formation. Sugar also appears with Flower cloud formation but is less frequent. Gravels and Fish cloud formation also appears with other cloud formation. Sugar, Gravel, and Fish also appears all together in some instances. Flower tends to occur less frequently with other clouds, and the combination of Gravel and Flower occurs but at much less frequency compared to others. In fact, Sugar, Gravel, and Fish appear all together more frequently than Grave and Flower. However, it's not like Flower cloud formation never occurs with other cloud formation, just occurs less frequently compared to others. In summary, they are all combination of cloud formations appearing together is a possibility, and the combinations between Sugar, Fish, and Gravel are more likely than with Flower cloud formation.

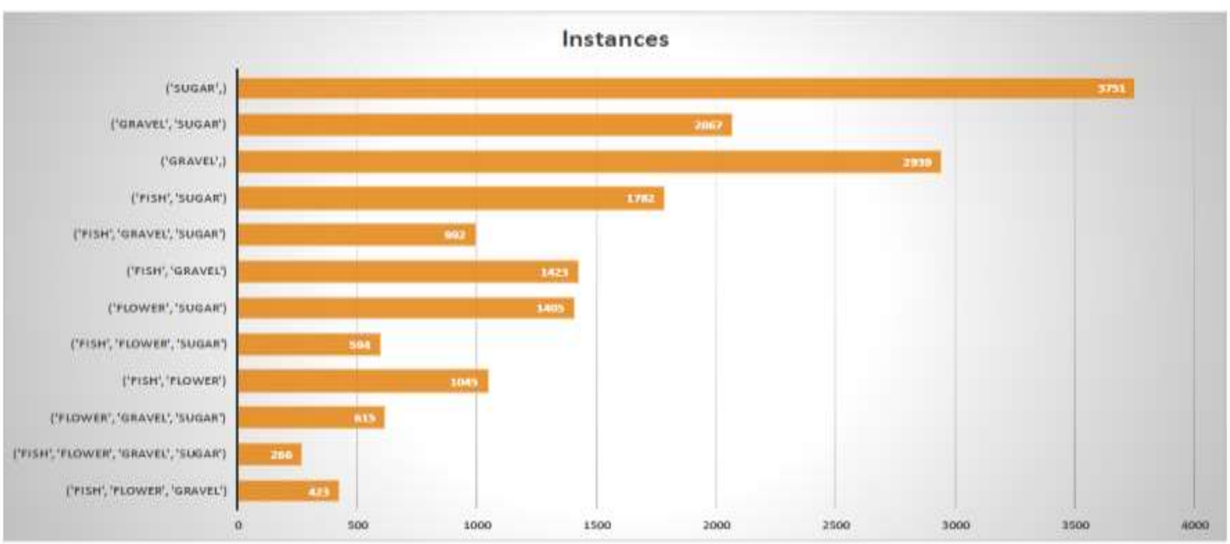


Figure 7: Plot showing frequent patterns of the cloud formations.

- In preprocessing after the removal of missing values stated above, we resized the images into sizes 512*512 and 320*320 and created custom image data generator classes for the two models(Mask-RCNN and UNET) respectively. We also used a train-validation split with a ratio of 90:10 for both the models.

3. Model Selection

Reading various literature, upon further research and with the help of the professor we decided to implement two artificial deep neural networks, Mask-RCNN and UNET.

3.1 MASK RCNN

Mask RCNN is an artificial deep neural network aimed to solve object instance segmentation problem in machine learning or computer vision. Mask R-CNN efficiently detects objects in an image while simultaneously generating a high-quality segmentation mask for each instance. It extends Faster R-CNN by adding a branch for predicting a binary object mask in parallel with the existing branch for bounding box recognition [1].

There are two stages of Mask R-CNN shown in figure-8. The first stage, called an RPN, proposes potential regions where there might be objects for a given image. The second stage, extracts features using ROI Pool from each candidate box(proposed potential regions), performs classification along with bounding box regression and generates a binary mask for each ROI. Both stages are connected to an FPN style deep neural network usually termed as the backbone structure. The backbone is usually a pretrained network like ConvNet, VGG or ResNet.

In the first stage the RPN scans all the feature maps and proposes regions which may contain objects. To bind features to its raw image location efficiently the network uses a set of boxes with predefined locations and scales relative to the image called Anchors. The true masks and the bounding boxes are assigned to individual anchors according to some preset IoU value. The RPN uses anchors with different scales bind to different levels of feature map to figure out where the feature map 'should' get an object and what size of its bounding box is [3].

In the second stage the regions proposed by the first stage are assigned to several specific areas of a feature map, scans these areas, and generates object classes, bounding boxes and masks. This procedure looks similar to RPN, but the difference is that, stage-two uses ROIAlign to locate the relevant areas of feature map instead of anchors, and there is an additional branch which generates masks for each object in pixel level (pixel level classification).

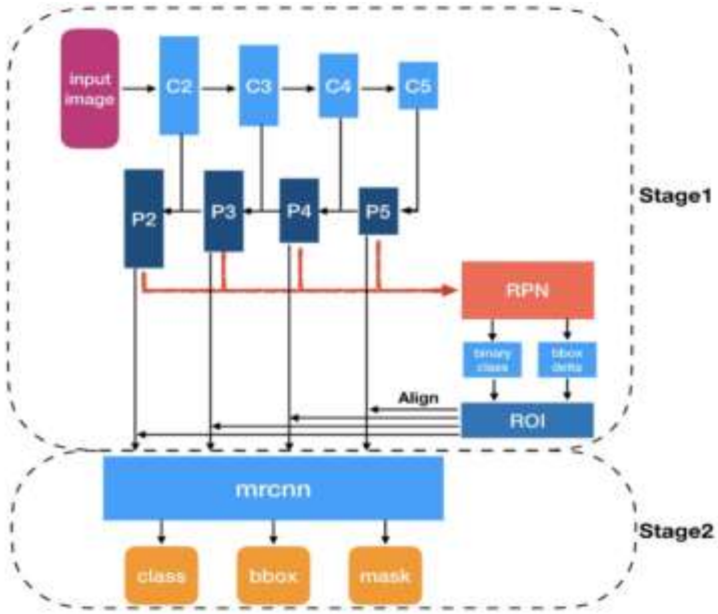


Figure 8: Mask – RCNN architecture

3.2 U-NET

The Artificial deep neural network takes the shape of an 'U' hence termed as U-net. The U-Net architecture for the most part is symmetric and consists of two major parts shown in Figure-9, the left/down part which is a contracting/downsampling path and the right/up part which is an expanding/upsampling path.

The contracting path is similar to an encoder and consists of several convolution layers followed by an activation function(ReLU), batch norm and max-pooling layers. Its purpose is to capture the context of the input image via a compact feature map in order to perform segmentation. This coarse contextual information will then be transferred to the upsampling path by means of skip connections [5]. The encoder part is referred as a backbone of the U-Net and is usually a pretrained network like VGG, ResNet, InceptionNet, EfficientNet, DenseNet, etc.

The expanding path is similar to a decoder which consists of deconvolution layers (up-sampling) and concatenation followed by a symmetrical number of convolution layers with an activation function (ReLU) and batch normalization to that of the encoder part. The decoder part's purpose is to enable precise localization combined with contextual information from the contracting path. This step is done to retain boundary information (spatial information) despite down sampling and max-pooling performed in the encoder stage [2].

In general terms the encoder part encodes the input image into feature representation at multiple different levels and the decoder part semantically projects the discriminative lower resolution features learnt by the encoder into higher resolution pixel space to get a dense classification.

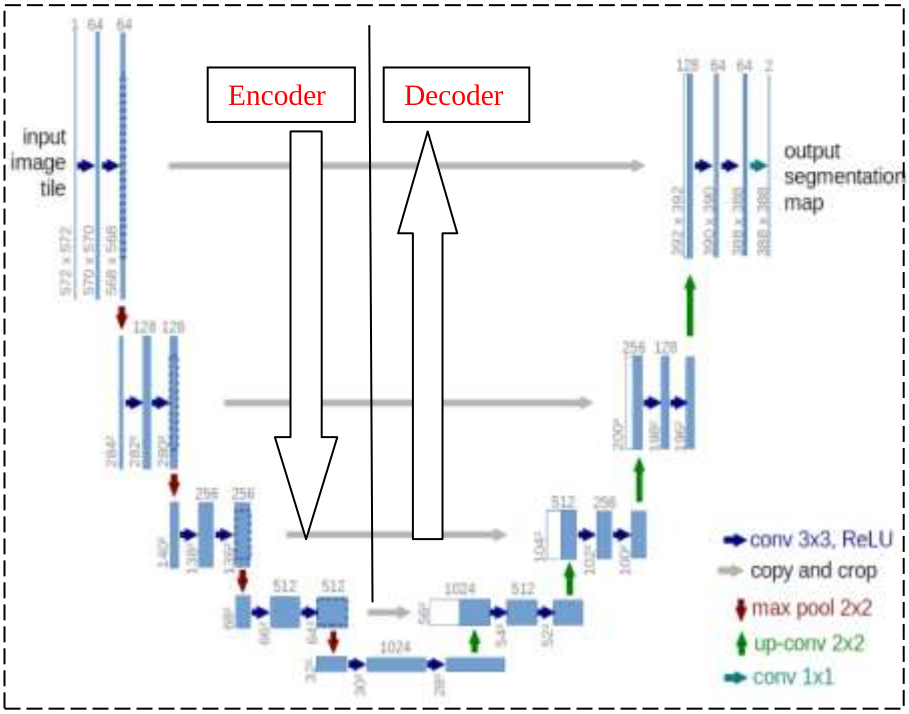


Figure 9: U-Net architecture

4. Model Training

- The Mask-RCNN model designed used a ResNet-101 architecture as a backbone encoder and was initialized with COCO (Common Objects with Context) weights. Most of the hyperparameters used default values except some parameters like RPN Anchor Scales, Steps per epoch, etc, were tuned accordingly. We split the dataset into 90% train and 10% validation sets and images were resized to 512*512 to speed up the computation. Image augmentations were used during training due to the small dataset. The entire model was train and validated using Adam optimizer with average binary cross-entropy loss and Dice loss(1-Dice Coefficient) as the performance metric for 25 epochs on a learning rate of 0.0002. Initially for the first epoch only the heads (top layer) were trained and for remaining epochs all the layers were trained. Training and validation loss plots for the model are shown in figure-10. The top plot entails the combined loss for all the different sub networks(RPN, class, bounding box and mask) in the Mask-RCNN model. The bottom plot shows the Dice loss for the generated masks.

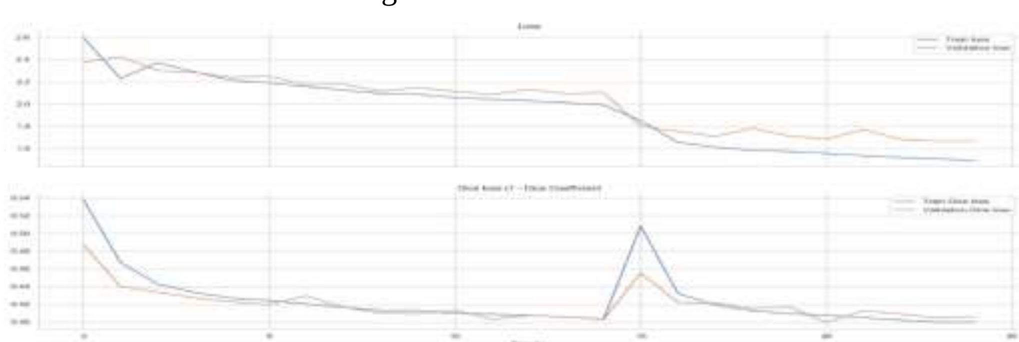


Figure 10: Loss plots for Mask-RCNN

- The implemented UNET model used an EfficientNet-b3 architecture as a backbone encoder and was initialized with Imagenet weights. We split the dataset into 90% train and 10% validation sets and images were resized to 320*320 to reduce the training time. We relied heavily on image augmentations to ensure the model could generalize well even though the images were reduced to a very small size. We used a modified Adam optimizer so that several hyperparameters and learning rate can be improved and optimized adaptively. The entire model was train and validated using the modified Adam optimizer with balanced cross-entropy dice loss(BCE) and Dice Coefficient as the performance metric for 15 epochs with an initial learning rate of 0.002. Unlike Mask-RCNN all the layers were trained for each epoch. Training and validation loss plots for the UNET model are shown in figure-11.

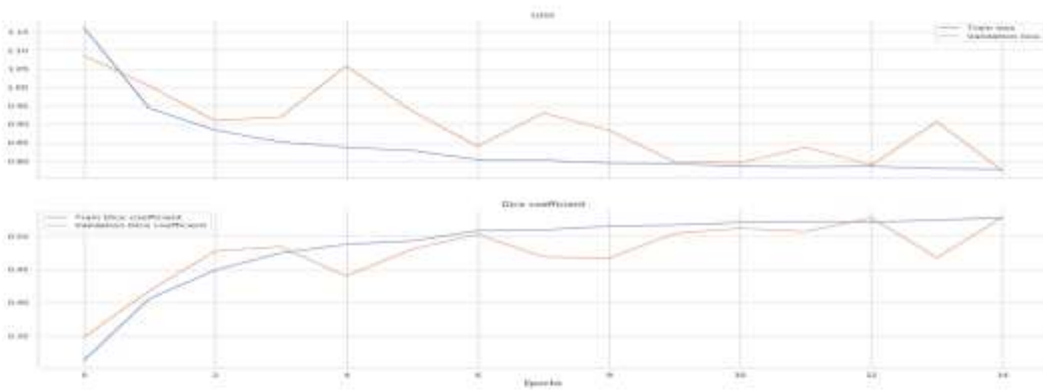


Figure 11: Loss plots for UNET

5. Model Evaluation

Both the models were evaluated on MAP (Mean Average Precision) metric using the validation set and the results are show in Table-1

- **Mean Average Precision:**

  Average precision is a popular metric in measuring the accuracy of object detectors like Masked RCNN. Average precision is a measure that combines recall and precision for ranked retrieval results and the Mean Average Precision (MAP) is the arithmetic mean of the average precision values for an information retrieval system over a set of Q query topics.

$$\mathrm{MAP} = \frac{\sum_{q=1}^{Q} \mathrm{AveP(q)}}{Q}$$

- MAP for two models:

Mask-RCNN	U-Net
0.33	0.31

6. PREDECTED RESULTS

- **Mask RCNN**: Model is initialized with its best weights and masks along with the cloud class were predicted for some sample images in the validation set.

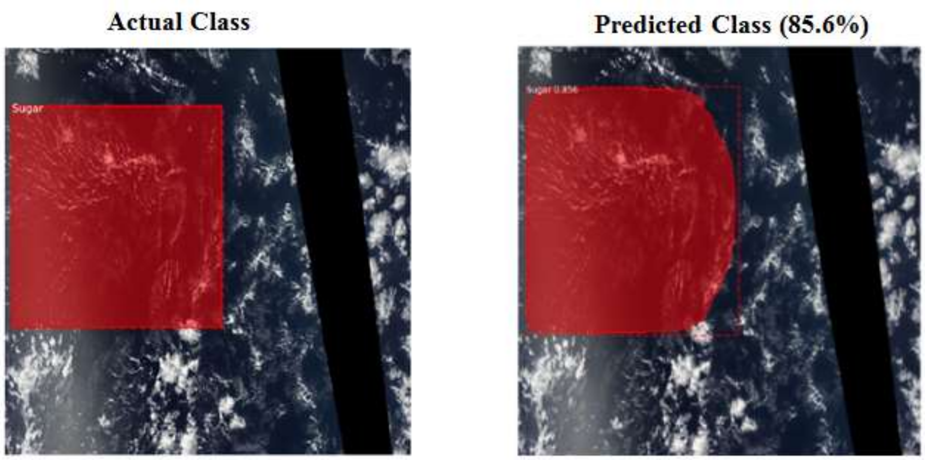


Figure 12: Prediction for one class label

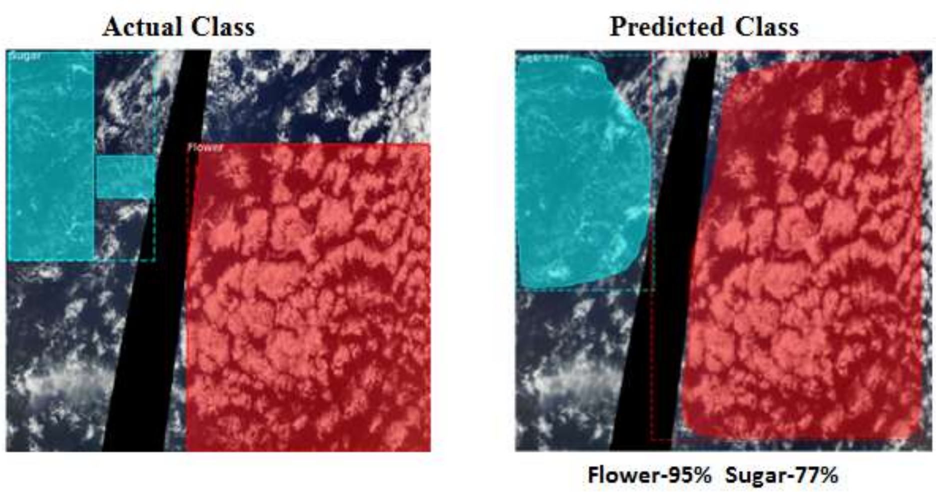


Figure 13: Prediction for Two class label

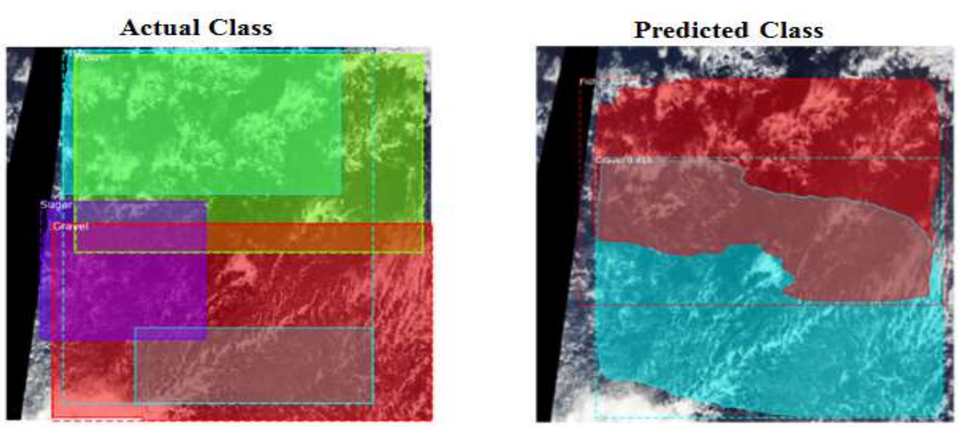


Figure 14: Prediction for Three class label

- **U-Net:** Model is initialized with its best weights and masks along with the cloud class were predicted for some sample images in the validation set after post processing.

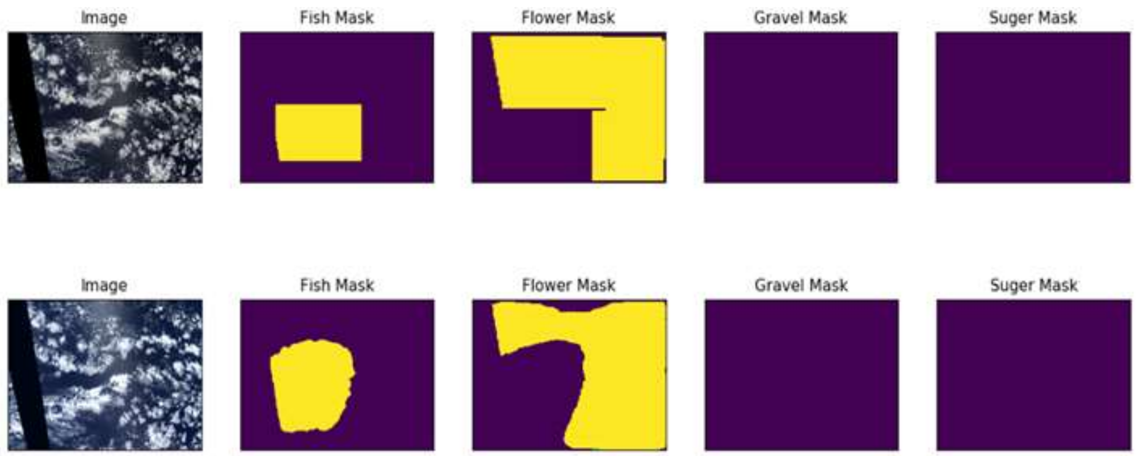


Figure 15: Prediction for one class label

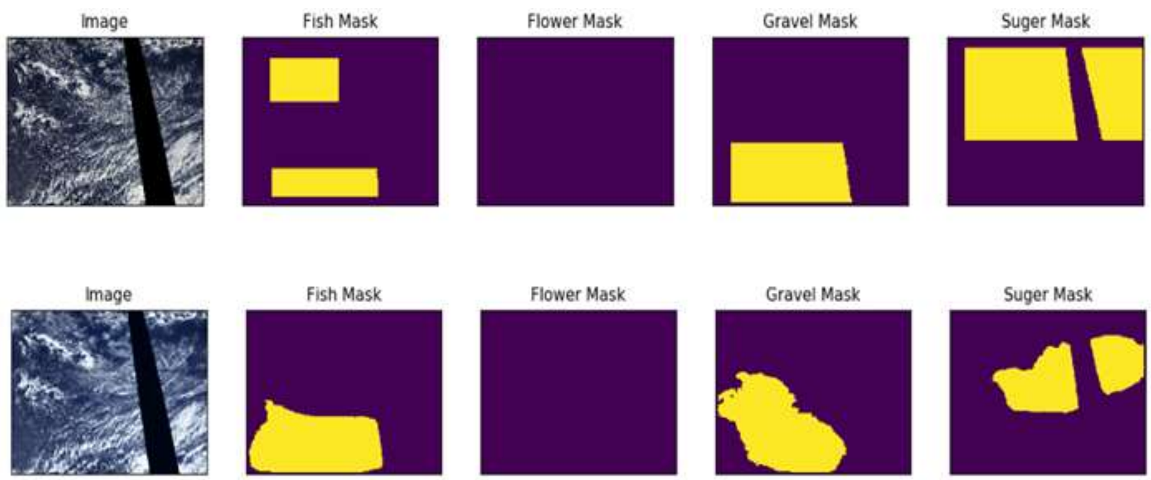


Figure 16: Prediction for Two class labels

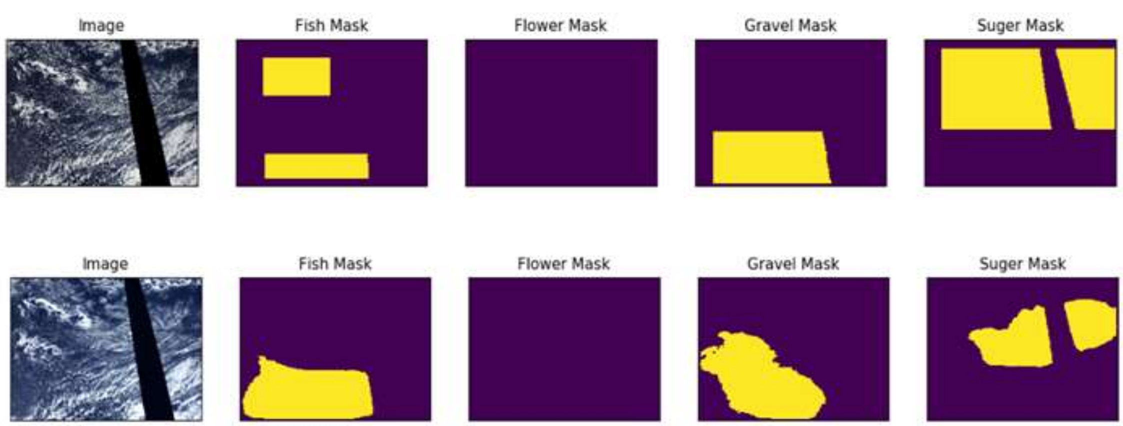


Figure 17: Prediction for Three class label

7. CONCLUSION

Climate change in the present day is an important issue and shallow clouds play a huge role in determining the earth's climate [6]. Classifying different cloud types can help researchers build better climate models. In this project we built two artificial deep neural network models Mask-RCNN and UNET to classify 4 cloud organization patters namely Fish, Flower, Sugar and Gravel from satellite images. We used the concept of transfer learning to train both models. The Mask-RCNN was trained using a ResNet-101 backbone with Adam optimizer and the UNET model was trained using EfficeintNEt-b3 backbone with a modified Adam optimizer. Both the models were trained using Dice loss as the performance metric and are evaluated on MAP (mean average Precision) metric. It was observed that the Mask-RCNN performed slightly better with a MAP of 0.33 on the validation set compared to a MAP score 0.31 of the UNET model. Although we used the same data this is not a fair comparison as many parameters were different input images sizes, backbones and main being the number of epochs trained (25 for Mask-RCNN and 15 for UNET). We also observed that it was difficult for both the models to predict mask which were overlapping each other, and which contained a lot of background noise, that being said with more training time and proper image preprocessing both the models can perform much better.

8. FUTURE WORK

The avenues for future work include the following:

- Improving the predictive results by using ensembles of models.
- Use Cross validation to better assess the effectiveness of the model.
- A better statistical approach in choosing the right thresholds in post processing.
- Better Hyper parameter Tuning and using different loss functions.
- Use different model architectures and Backbones(pre-trained networks),
- Use the classifier with explain ability techniques like Gradient-weighted Class Activation Mapping to generate a baseline.

9. REFERENCES

[1] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).

[2] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.

[3] https://wiki.ubc.ca/CNNs_in_Image_Segmentation

[4] Rasp, S., Schulz, H., Bony, S., & Stevens, B. (2019). Combining crowd-sourcing and deep learning to understand meso-scale organization of shallow convection. *arXiv preprint arXiv:1906.01906*.

[5] Zantedeschi, V., Falasca, F., Douglas, A., Strange, R., Kusner, M. J., & Watson-Parris, D.

(2019). Cumulo: A Dataset for Learning Cloud Classes. *arXiv preprint arXiv:1911.04227*.
[6] https://www.kaggle.com/c/understanding_cloud_organization/data#_=_
[7] Segmentation models: https://github.com/qubvel/segmentation_models
[8] Mask RCNN: https://github.com/matterport/Mask_RCNN
[9] https://engineering.matterport.com/splash-of-color-instance-segmentation-with-mask-r-cnn-and-tensorflow-7c761e238b46