# EVALUATING MACHINE LEARNING MODELS FOR SEMANTIC SEGMENTATION OVER CLOUD IMAGES FOR CLASSIFICATION

by
Harsh Nagarkar

A thesis submitted to the faculty of The University of Mississippi in partial fulfillment of the requirements of the Sally McDonnell Barksdale Honors College.

Oxford
May 2020

Approved by

Advisor: Dr. Yixin Chen

Reader: Dr. Dawn Wilkins

Reader: Dr. Feng Wang

ABSTRACT

Due to the increasing number of available approaches now a days, choosing the most accurate image semantic segmentation model has become hard. The purpose of this research is to find the best performing image semantic segmentation model for Cloud classification.For the purpose of this study, a data set of cloud images from the Max Planck Institute for Meteorology is used. These images were taken from the by two NASA space satellites.

Three main models UNet, PSPNet and FPN were used in combination of 4 different encoder Inception-ResNet-v2, MobileNet-v2, ResNet-34, and ResNet 101. After training all the models in Mississippi Center for Super Computing, the results were plotted. Overall the models turned out broadly similar to each other. Even so, the FPN model with the MobileNet-v2 encoder backbone stood out first followed by the UNet model with the Inception-ResNet-v2 encoder backbone in second place.

# DEDICATION

This thesis is dedicated to my Mom and Dad who have always inspired and loved me.

ACKNOWLEDGEMENTS

First and foremost I would like to thank Dr. Yixin Chen for his advice, guidance, and patience through the process of writing this paper. His motivation and knowledge has inspired me to write this thesis. I would also like to thank my committee members Dr. Dawn Wilkins and Dr. Feng Wang for their encouragement and support. Without their help, this thesis would not have been possible.

Secondly, I would like to thank the Sally McDonnell Barksdale Honors College for giving me this wonderful opportunity. I would like to extend my sincere acknowledgment to the exceptional faculty at the Computer and Information Science department for the past four years in the University of Mississippi.

Lastly, I would like to thank my family and friends for their unconditional love and support through this project. They kept me going even when times were hard.

TABLE OF CONTENTS

LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

New advancements in artificial intelligence are coming to light, especially in the field of digital image processing. New methods like gated CNN, Deeplabv3, etc. of supervised and unsupervised kind are emerging. Even though these models might look new, fundamentally they rely upon the core base models created earlier. This paper attempts to explore such widely used semantic segmentation models to classify and segment clouds objects from the sky image.

The images used in this paper are taken from the Max Planck Institute of Meteorology. These images were captured from the NASA world view from two NASA space satellites 'TERRA' and 'AQUA'. The masks containing clouds in these images were human drawn by 67 people as a part of a crowd funding initiative using the Zoo-Universe platform. [7]

Fundamentally each model can interact differently based on a data set. A model's performance is based on the input data it has, the features extracted and learned from the input and the gradient of learning itself. A good high quality sharp image compared to low quality smoothed image produces a different outputs in segmentation. A model's learning can related to the locality of the pixels to the exact location. On top of that these model's training can be stopped at various points. Taking into consideration these parameters, here we are trying to explore the performance of UNet, FPN, and PSPNet models with the combination of four different encoders ResNet-34, ResNet-101, MobileNet-v2 and Inception-ResNet-v2 using the crowd source data set. Data set being crowd sourced also raises the question of quality furthermore which model would handle such noisy data set.

# CHAPTER 2

# BACKGROUND AND MOTIVATION

In order to confirm the performance of each base models with the combination different encoder,we have to use empirical evidence. From this evidence then we have to backtrack and find out why one model may perform better than the other.

## 2.1 Brief model summary

In this paper three fundamental model were applied to process images.

### 2.1.1 UNET

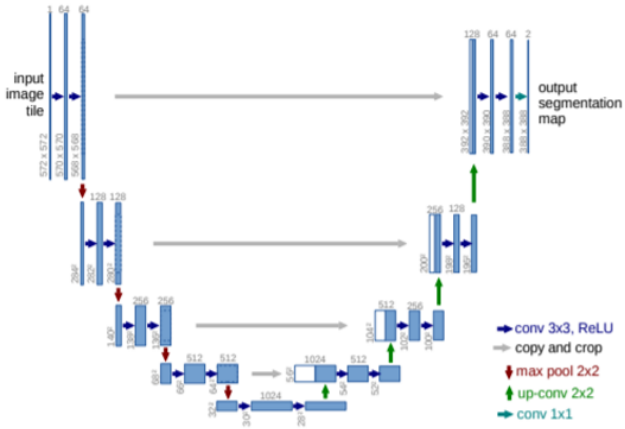


Figure 2.1: Unet model layers [1].

This is the most basic and well known model, it consists of two path ways; contracting and expansive pathway. In the contracting pathway, repeated application of two 3x3

convolution (unpadded) and one max pooling operation followed by rectified linear function on each step is performed (as shown in Figure 2.1). This reduces the resolution on each contracting pathway step, doubling the image feature channels. The expanding pathway consists of repeated up sampling of the feature maps by up convolution of concatenation of cropped feature map from the contracting channel. This is followed by two 3x3 convolution and a rectified linear function. The end is later connected to fully connected layer for classification [1].

### 2.1.2 PSPNET

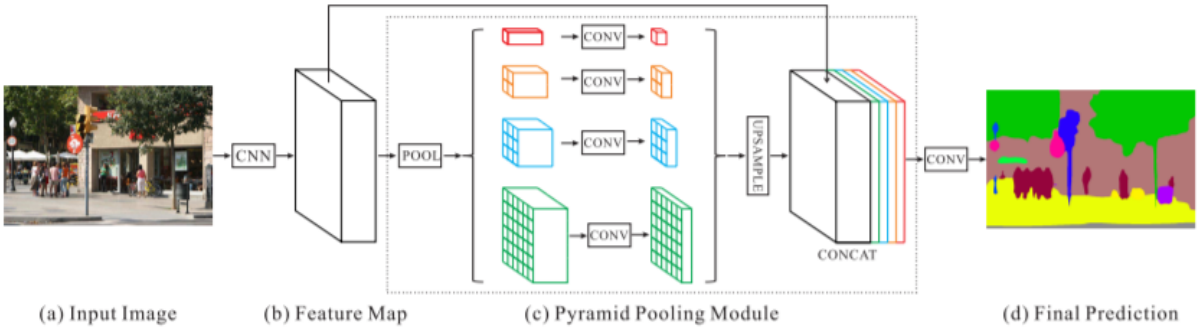


Figure 2.2: PSPNet model layers [2].

Pyramid parsing network(PSPNet) is a popular advance network where an initial feature map is generated using backbones like pertained ResNet. This feature map is then passed into 4 layered pyramid pooling. Here, whole maps, half maps and so on are generated, convolved and then up sampled to original feature map size. These up sampled feature maps are then concatenated with the original feature map and then convolved to produce the final output prediction (Figure 2.2). This allows the network to preserve and use the local semantic information as a global prior in prediction[2].

### 2.1.3 FPN

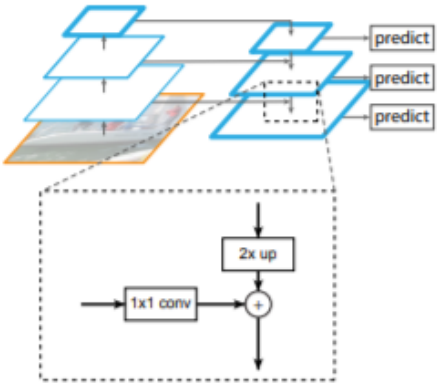


Figure 2.3: FPN model layers [3].

Feature pyramid network(FPN) is the next generation of PSPNet. It consists of two pathways, bottom up and top to bottom. Here the feature maps are calculated at each level independent of the backbone architecture as shown in Figure 2.3. The bottom up pathway is feed forward covnet which calculates the feature maps and then passes the data to the next stage with a dimensional reducing factor of two. There can be multiple layers in a stage. In the top to bottom stage the feature maps are up sampled and laterally merged with the bottom top path's feature map hence matching the same size. This creates semantic, high locality and accurate information, which is used for prediction [3].

## 2.2 Brief encoder blocks summary

Each model was paired with an encoder so as to reduce the computation work while processing and at the same time increasing its performance. Hence in Unet, PSPNet and FPN instead of plain old convolution, encoders were used to extract features.

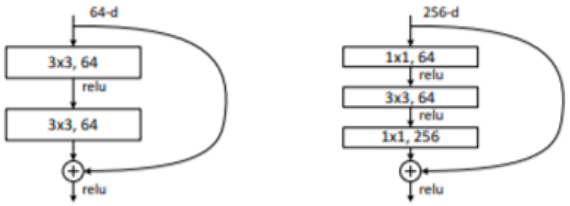


Figure 2.4: ResNet block and BottleNeck Block [4].

### 2.2.1 ResNet

ResNet encoder was created to solve the problems of varnishing gradient and low convergence rate. The residual and bottle neck block shown in Figure 2.4 uses the residual information passed with the skip connections for learning. These blocks use skip connections as either identity mapping or as identity plus zero padding for increasing size[4].

### 2.2.2 MobileNet-v2

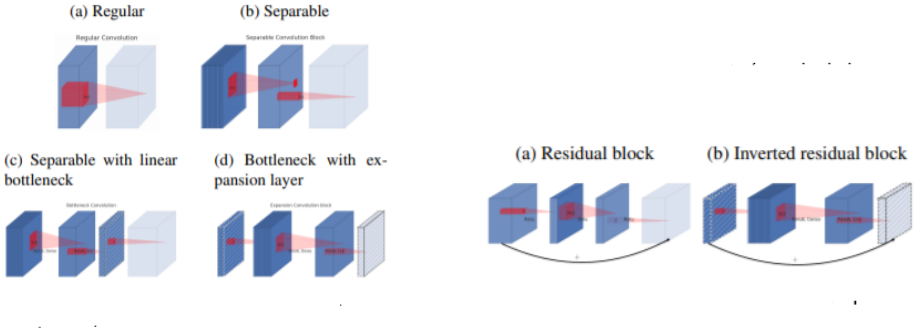


Figure 2.5: MobileNet v2 depth wise convolution and inverted Residual block [5].

MobileNet-v2 is a novel encoder which is tuned for a more efficient and memory friendly approach. This is achieved by using a linear bottleneck and rectified linear Unit with 6 bit compression function. This model inside uses inverted residual blocks that means the output size will be smaller compared to input of the block and depth wise convolution to speed up. The comparison of residual block vs inverted residual block and convolution vs depth wise convolution can be seen in the Figure 2.5 [5].

### 2.2.3 Inception-ResNet-v2

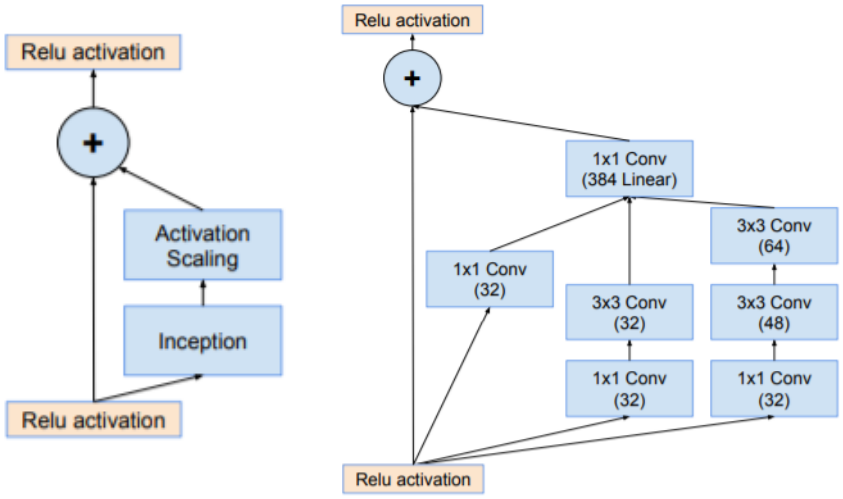


Figure 2.6: Inception-ResNet-v2 overview and Inception-ResNet-v2 block A [6].

Inception-ResNet-v2 model is an adaptation of the ResNet-v2 model. The former uses the latter's convolution 3x3 and convolution 1x1 along with combinations of different inception blocks. This model is designed towards replacing one high convolution dimensional filter with lower multiple convolution blocks, keeping the dimension of the data same throughout the model. This model claims to perform better than ResNets. This model's architecture and its type A block are shown in Figure 2.6 [6].

# CHAPTER 3

# DATA SET DESCRIPTION

## 3.1 Source of image

The images were taken by the satellites under National Aeronautical Space Administration (NASA).These images were then transported by the International Space Station Institute (ISSI) to NASA. After which the Max Planck Institute of Meteorology compiled it from NASA world view. Finally, using these images Max Planck created a competition on Kaggle.com and published the dataset on `https://www.kaggle.com/c/understanding_cloud_organization`. The images were taken to study the cloud patterns over the region of trade wind in east of Barbados. Three regions, spanning 21 degrees' in longitude and 14 degrees' in latitude were captured by the TERRA and AQUA satellites. These images were used as a part of crowd sourcing data collection study for testing human and machine learning cloud classification. This study was conducted for the mesoscale organization of shallow convection by using deep learning[7].

## 3.2 Background of data set

The data set is comprised of images and their respective masks marked by participants in the 'Mesoscale organization of shallow convection by using deep learning' paper [7].

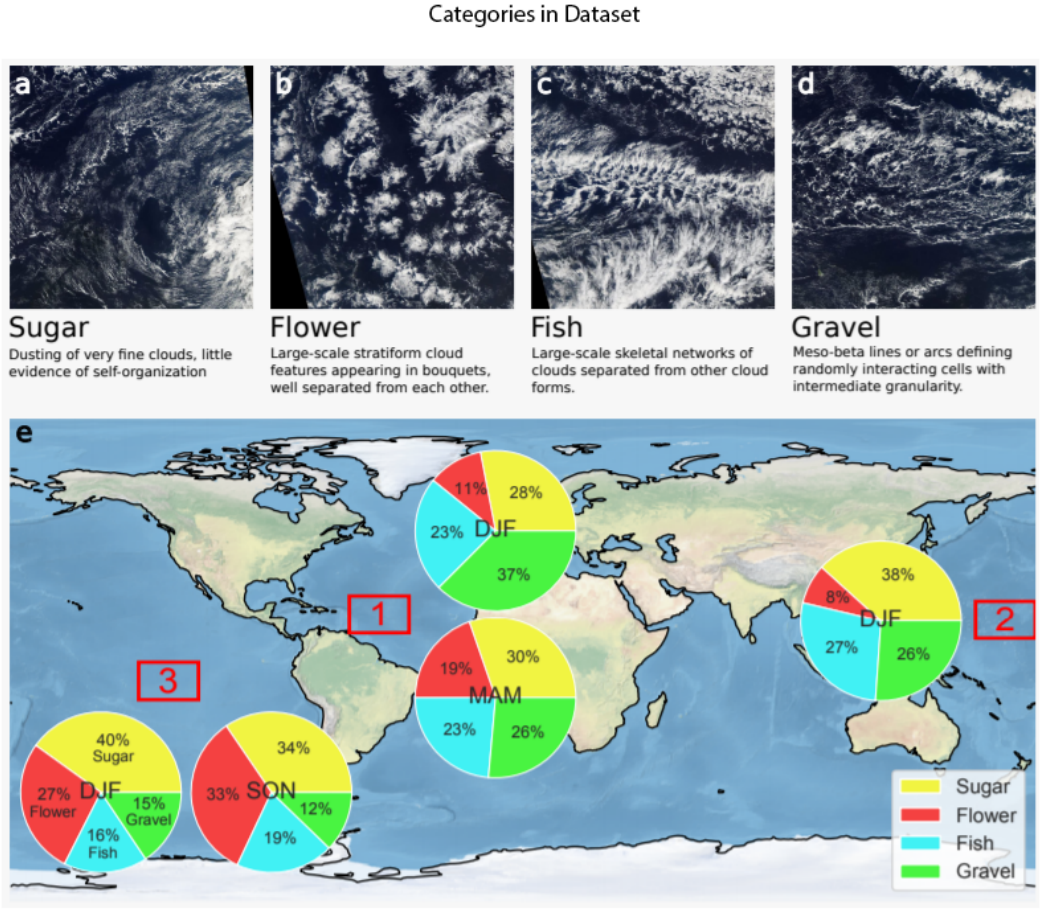


Figure 3.1: Categories in the data set [7].

### 3.2.1 Categories in the Data set

There are 4 categories of cloud identified based on the season, winds, shape and size as shown in Figure 3.1.

### 3.2.2 Human Classification of Images

The human classification experiment for preparing this data set, comprised of 67 people, mostly scientists who carried out this experiment during a day. Online crowd sourcing platform named Zoo Universe was used to provide people with an interface to work on. All the participants first went through training before starting the data set classification and segmentation. For this task they were allowed to use bounding boxes only. An image was processed by maximum of four people and on average three people with no repeats on same participant. The participants were told that the masks can be either none or minimum of 10 % of the image. Furthermore, overlapping masks were allowed (Figure 3.2.1).

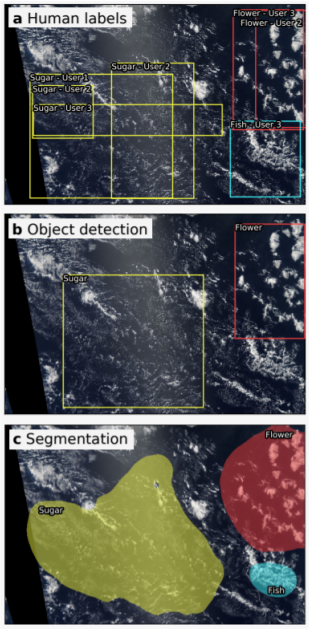


Figure 3.2: Trial Humans classification vs model training comparison [7].

### 3.2.3 Crosschecking and confirming the concept of human work

For the initial 900 images, comparison was done among different participant's bounding boxes to find such masks with at least 2 people's region being overlapped. The experimental model training was done on such 900 images. Which concluded nonrandom characteristics. We can see an example of such mask in Figure 3.2.1. In one category scientists agreed for 37 % of classification. These first 900 images used for training, showed a good chance that more data can be generated using this method and a model can be trained on it. Therefore, more data was compiled and published on Kaggle.com which is being used

in this work.

# CHAPTER 4

# DATA ANALYSIS AND PREPROCESSING

The data set consists of comma seperated values (CSV) file holding two columns. The first one holds image names concatenated with the mask type with an underscore and the second column holds the run length encoding of the pixels that are marked as part of the mask. There are 5546 images in train data set and 3698 in test data set. Each image is 1400 by 2100 pixels resolution.

## 4.1 Number of images with mask count from one to four

```
2    2372
3    1560
1    1348
4     266
Name: Image_Label, dtype: int64
```

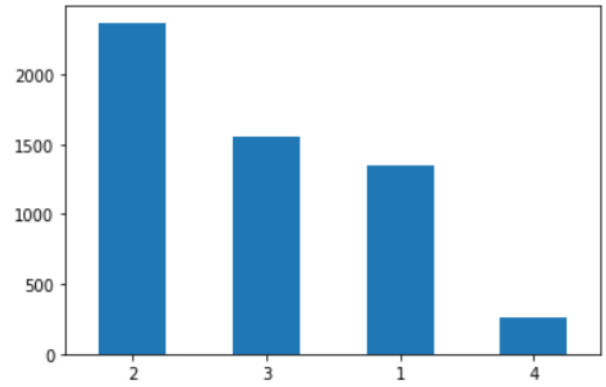


Figure 4.1: Mask counts per Images.

From Figure 4.1 we observe that images with two masks are of the highest volume and images of four masks are of lowest.

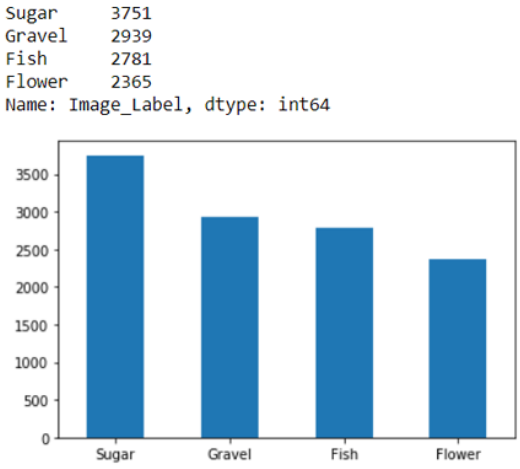


Figure 4.2: Images per mask categories.

## 4.2 Number of images per mask

We infer from Figure 4.2, that sugar is the most popular category among masks. Sugar category clouds are the most unorganized fine grain. The least popular category is flower, in this category clouds are organized in bouquet.

## 4.3 Occurrence of Null values in CSV

The data set's CSV files assumes that each image consists of four masks, but we observe most images don't have four mask types at the same time. Hence the empty masks category of such images were filled with null values in CSV files. We can see the null value count in Figure 4.3.

## 4.4 Mask shapes and sizes

The masks in the data set were not square and rigid even though the scientist who did them were allowed to draw only bounding boxes. Thus some of the masks were made of irregular shapes. These masks were created by multiple scientists using union of masks approach. The examples of such masks are shown in Figure 4.4. Also some masks acquired

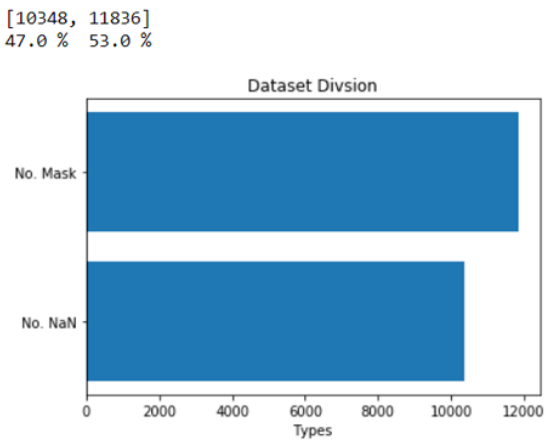


Figure 4.3: Count of Null values in total data set

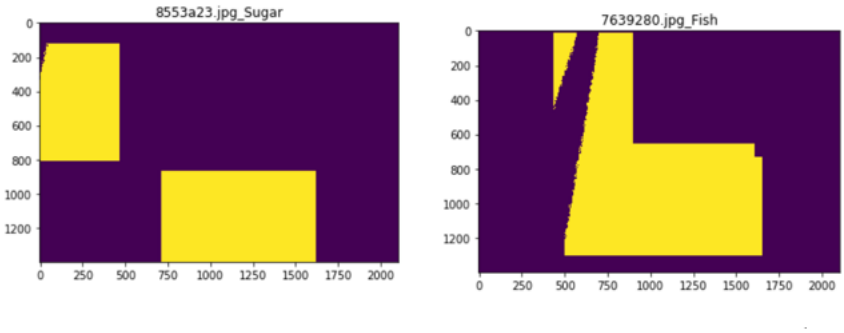


Figure 4.4: Example of irregular and multiple mask shapes

95% of the region, which is strange.

## 4.5 Sample Masks and encoding algorithm

The data set consists of run length encoded masks, that is, given a row of pixels min and max 'x' axis points are mentioned in the that row. In order to generate masks for an image the run length encoding was expanded and overlaid on the image file. Examples of these images are presented in Figure 4.5

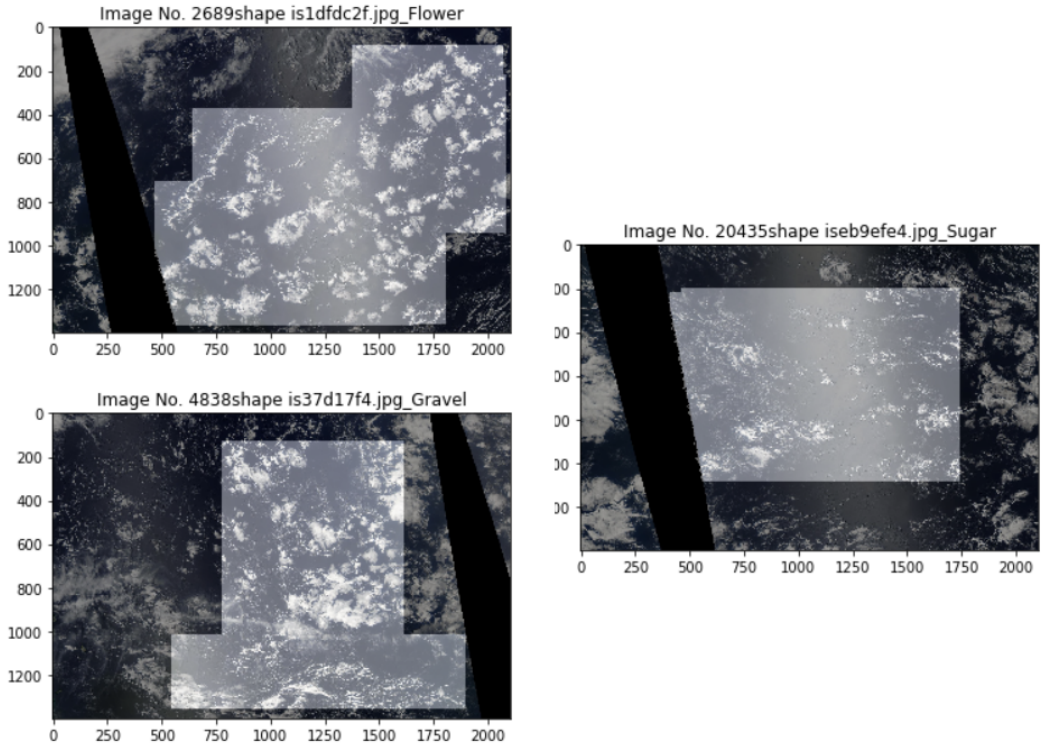


Figure 4.5: Example of irregular and multiple mask shapes.

## 4.6 Invisible Region in Images

The two NASA satellites used to capture the images in this data set have a relatively a small camera, due to which images from the two satellite orbits were stitched together. There are still remained some regions which were not covered by the satellites because of which these parts in the images were left blank and colored black. This is the reason why some images have black strips on them as shown in Figure 4.5.

# CHAPTER 5

# APPROACH TO EVALUTATE

## 5.1 Construction of model

To evaluate each model with a particular backbone involved, we first create one common input layer that would feed into the models. This first feeding layer was built using Keras python [8]. In this layer the images were first normalized between zero to one, resized to either 224x224 pixel resolution or in case of PSPNet, 240x336 pixel resolution and then split into individual color channels. The data was then passed into different models with different backbones in small batch sizes. For building these models a high level library called segmentation models was used [9]. This enabled testing and training for all the models without worrying too much about each model's implementation specific code. The models used the ImageNet weights as starting point to reduce the gradient of convergence. However the ImageNet data set is vastly different from the data set we have, hence we used a no top encoder freeze. The model's loss function was binary cross entropy with dice loss, as the binary cross entropy has a good backward propagation property where as dice loss has accurate area prediction. The model evaluation metric is dice coefficient as mentioned in the evaluation guideline on the Kaggle.com. Nesterov Adam optimizer was used with a learning rate of 0.0002 for these model. The Sigmoid activation function is along with these models in the fully connected layer.

## 5.2 Loss function and metric

### 5.2.1 Binary Cross Entropy Function

$$L(y, \hat{y}) = 1/N \sum_{i=0}^{N} (y * \log(\hat{y}_i) + (1 - y) * \log(1 - \hat{y}_i) \tag{5.1}$$

In binary cross entropy, Equation 5.1, $y$ is either one or zero. If it's one then it fall into one category, if zero then the other.

### 5.2.2 Dice Coefficient

$$\frac{2 \sum_i^N y \cdot \hat{y}}{\sum_i^N \hat{y}^2 + \sum_i^N y^2} \tag{5.2}$$

$$\frac{2TP}{2TP + FP + FN} \tag{5.3}$$

Dice coefficient is used to calculated the true region prediction based on local prediction and global set. In the Equation 5.3 'TP' stands for true positive, 'FP' stands for false positive and 'FN' stands for false negative. Using this formula a complete correct prediction will lead to one and complete wrong prediction will lead to zero. The mathematical formulation for this is depicted in Equation 5.2.

### 5.2.3 Dice Loss

$$1 - \frac{2 \sum_i^N y \cdot \hat{y}}{\sum_i^N \hat{y}^2 + \sum_i^N y^2} \tag{5.4}$$

The dice loss is the reverse of dice coefficient. That is, in dice coefficient zero means the prediction is false and one means it's true. To use this dice coefficient as a loss we need to reverse the scale where one means true and zero means false. The main reason for doing this is because the model is always trying to minimize loss. To do this we subtract dice

coefficient from one. See equation 5.4

## 5.3 Calculating results

Initially after running each different model for 25-50 epochs the first highest scoring dice's coefficient point was chosen from the graph of dice coefficient vs epochs for validation data set by visual means. The models were then retrained for only that many chosen epochs to reach the training point we observed in the previous graph. The final model's accuracy was calculated by generating the masks for the test data given by the competition and then submitting those masks to the competition website.

# CHAPTER 6

# RESULTS

## 6.1 Unet

### 6.1.1 Test and validation Scores

ResNet34 Validation data set on 8 epochs reaches an approximate accuracy of 0.50 and loss slightly increases to 0.7 as portrayed in Figure 6.1.

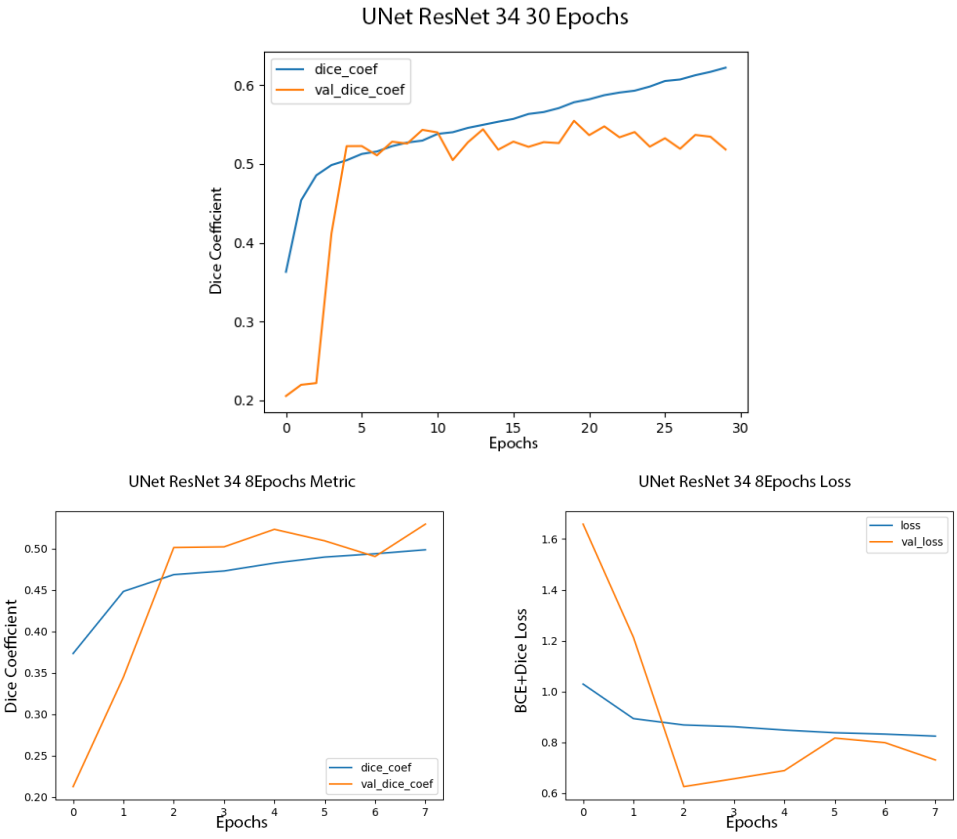


Figure 6.1: Resnet-34 backboned Unet 8 epochs.

ResNet101

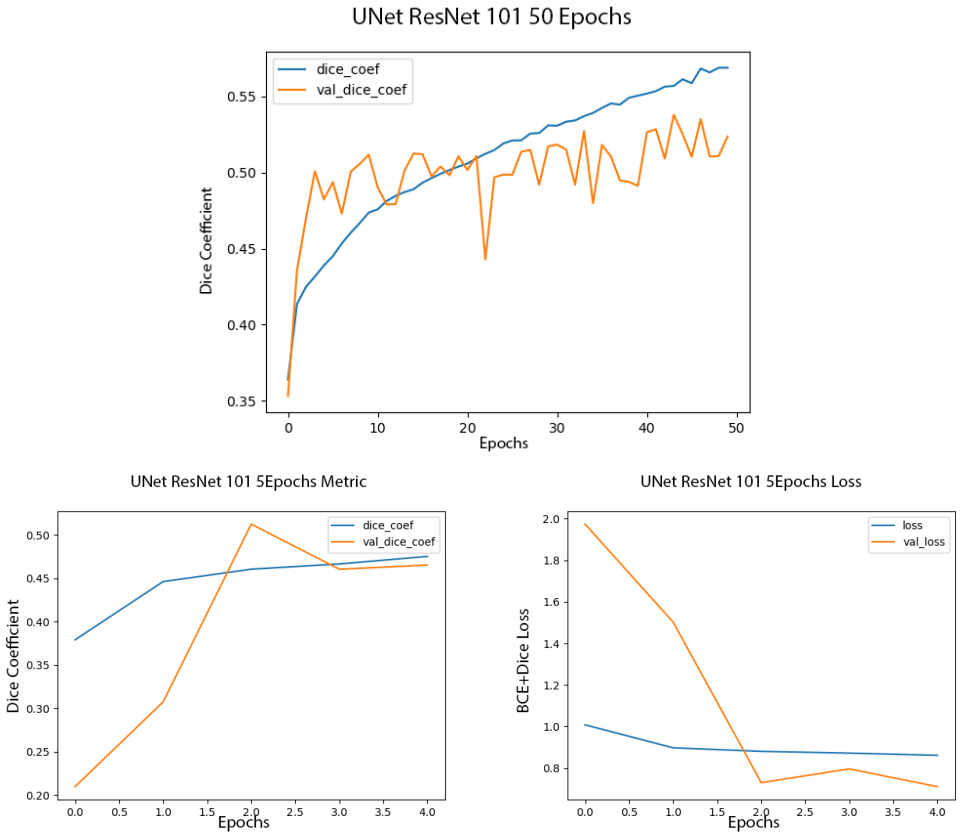


Figure 6.2: Resnet-101 backboned Unet 5 epochs.

Validation data set on 5 epochs reaches an approximate accuracy of 0.45 and loss decreases to 0.7 as portrayed in Figure 6.2.

MobileNetv2

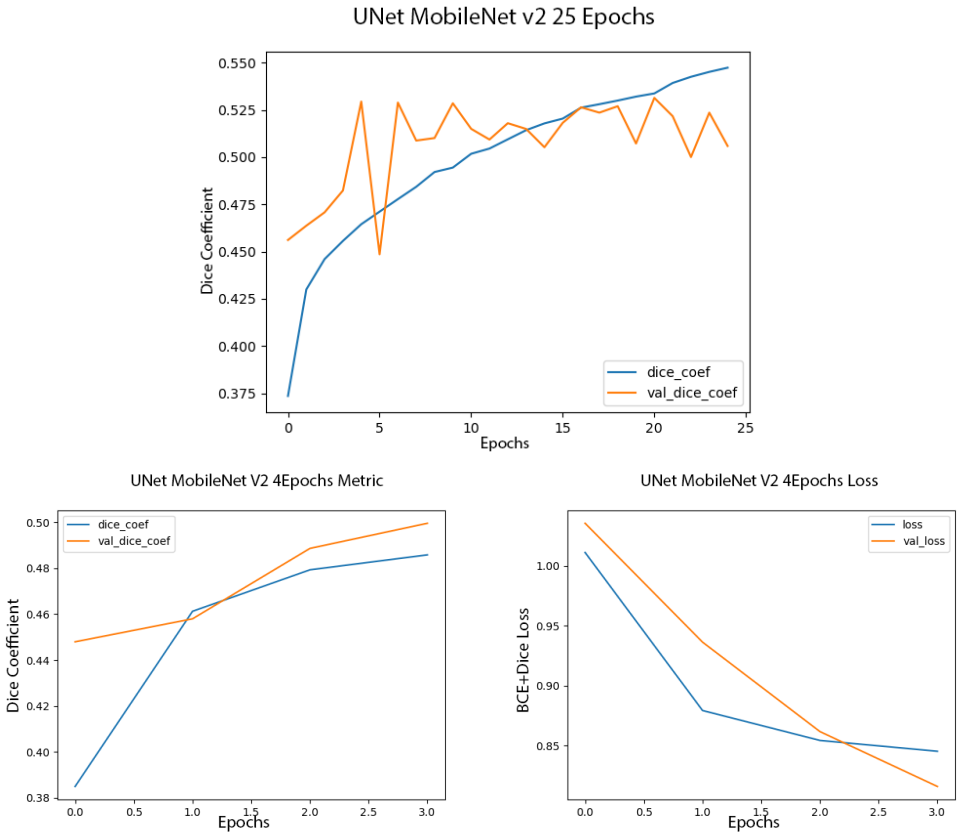


Figure 6.3: MobileNet-v2 backboned Unet 4 epochs.

Validation data set on 4 epochs reaches an approximate accuracy of 0.50 and loss decreases to 0.75 as shown in Figure 6.3.

InceptionResNetv2

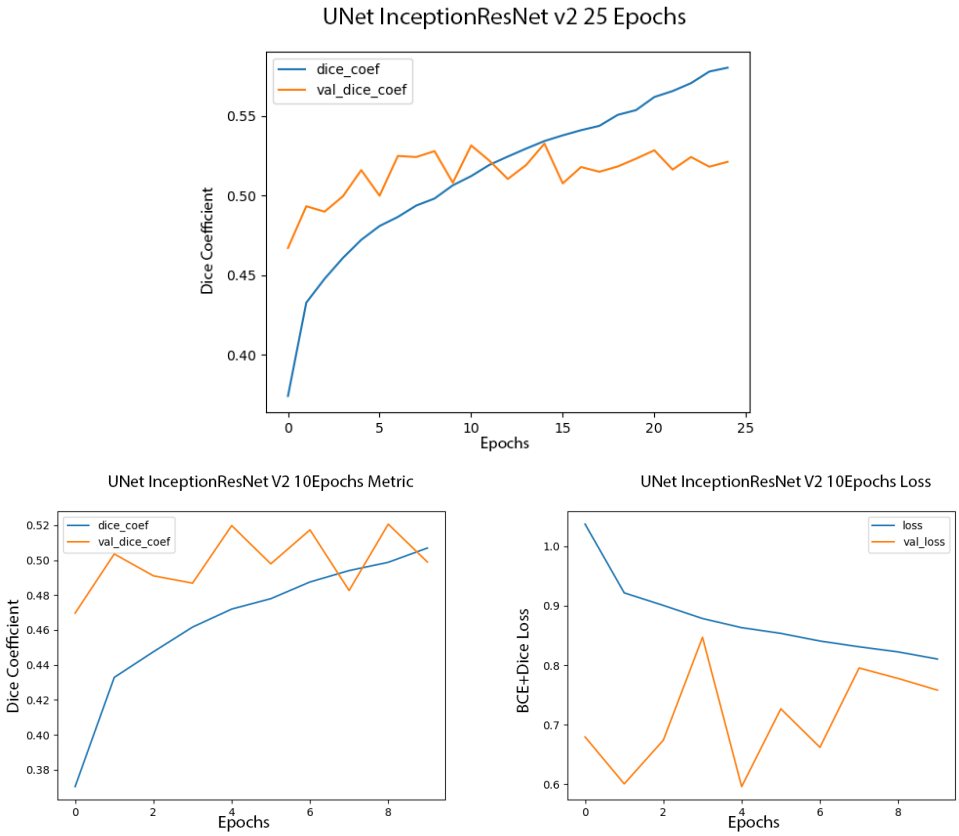


Figure 6.4: InceptionResNetv2 backboned Unet 10 epochs.

Validation data set on 10 epochs reaches an approximate accuracy of 0.50 and loss decreases to 0.75 as shown in Figure 6.4.

### 6.1.2 Unet Epochs and Accuracy

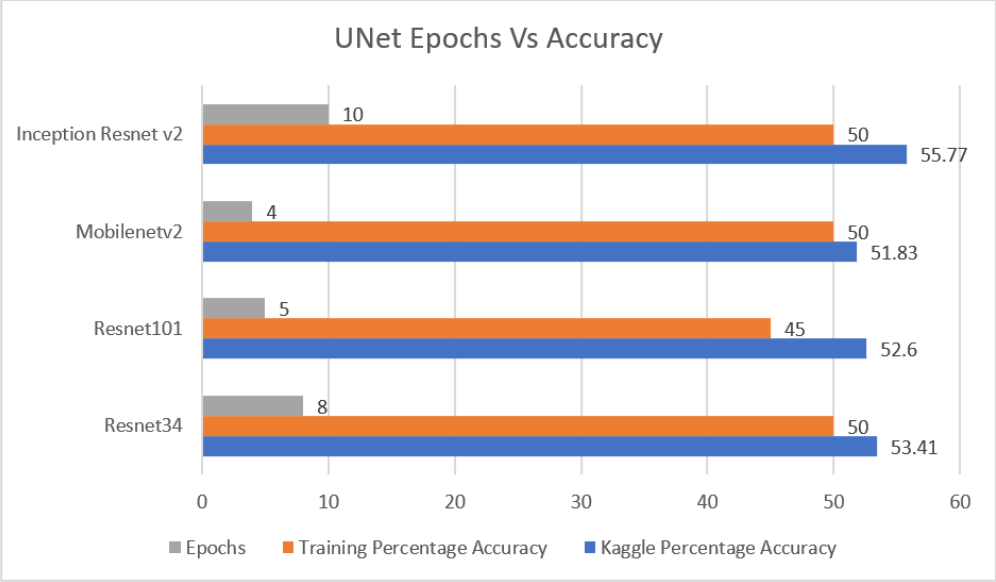


Figure 6.5: Unet Comparison graph epochs vs accuracy.

According to Kaggle's score the winner is UNet architecture with Inception-Resnet-v2 followed by Resnet-34 on the test data (Figure 6.5). We also see from Figure 6.1, 6.2, 6.3 and 6.4 that the loss never went below 0.7, which means that more training might be needed for the networks or the data set is not containing similarly mapped features.

## 6.2 PSPnet

### 6.2.1 Test and validation Scores

ResNet34

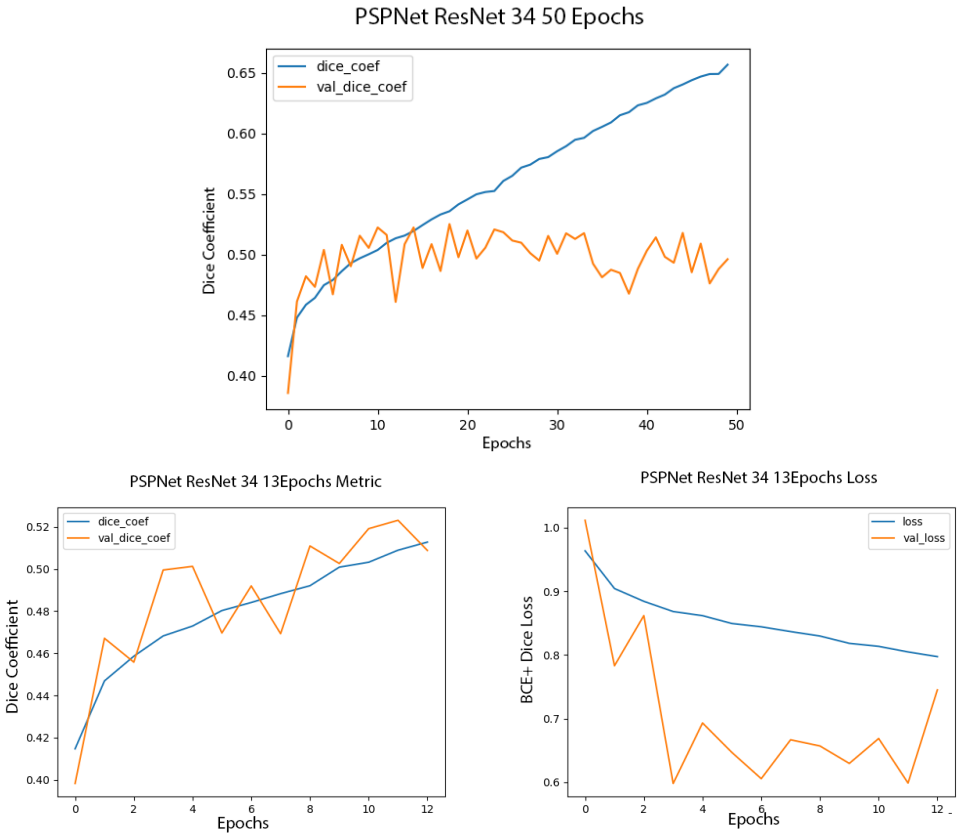


Figure 6.6: ResNet-34 backboned PSPNet 13 epochs.

Validation data set on 13 epochs reaches an approximate accuracy of 0.55 and loss slightly increases to 0.7 as shown in Figure 6.6.

ResNet101

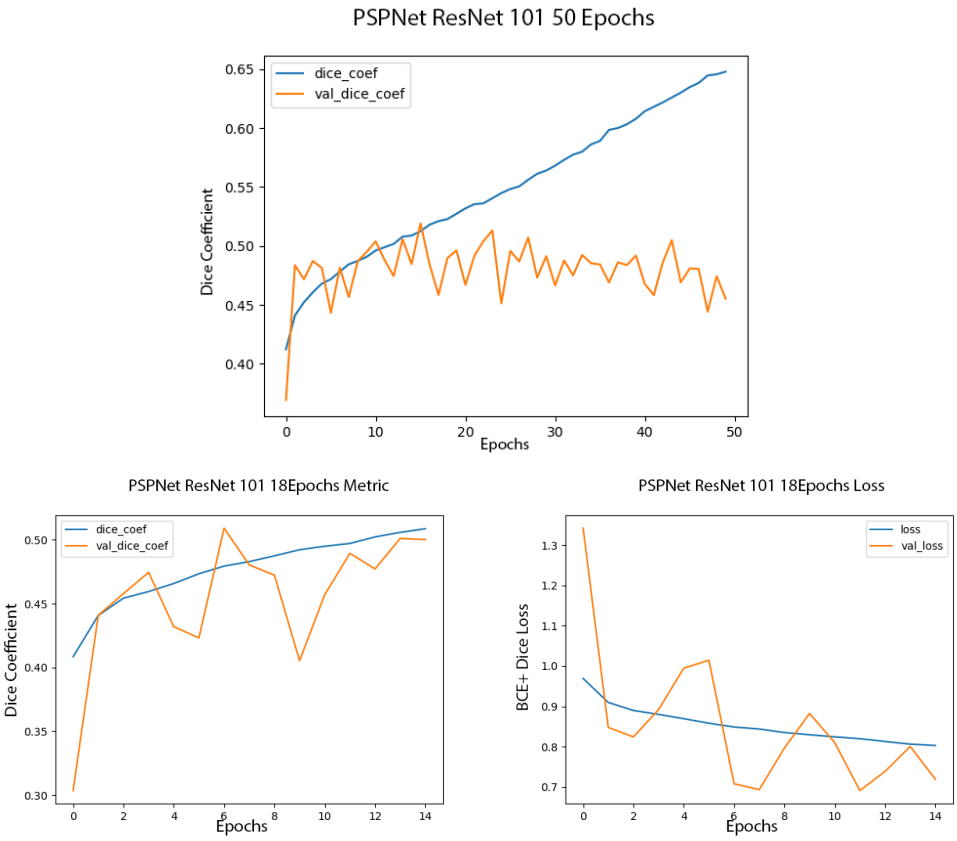


Figure 6.7: ResNet-101 backboned PSPNet 15 epochs.

Validation data set on 15 epochs reaches an approximate accuracy of 0.5 and loss continuously decreases to 0.7 as shown in Figure 6.7.

MobileNetv2

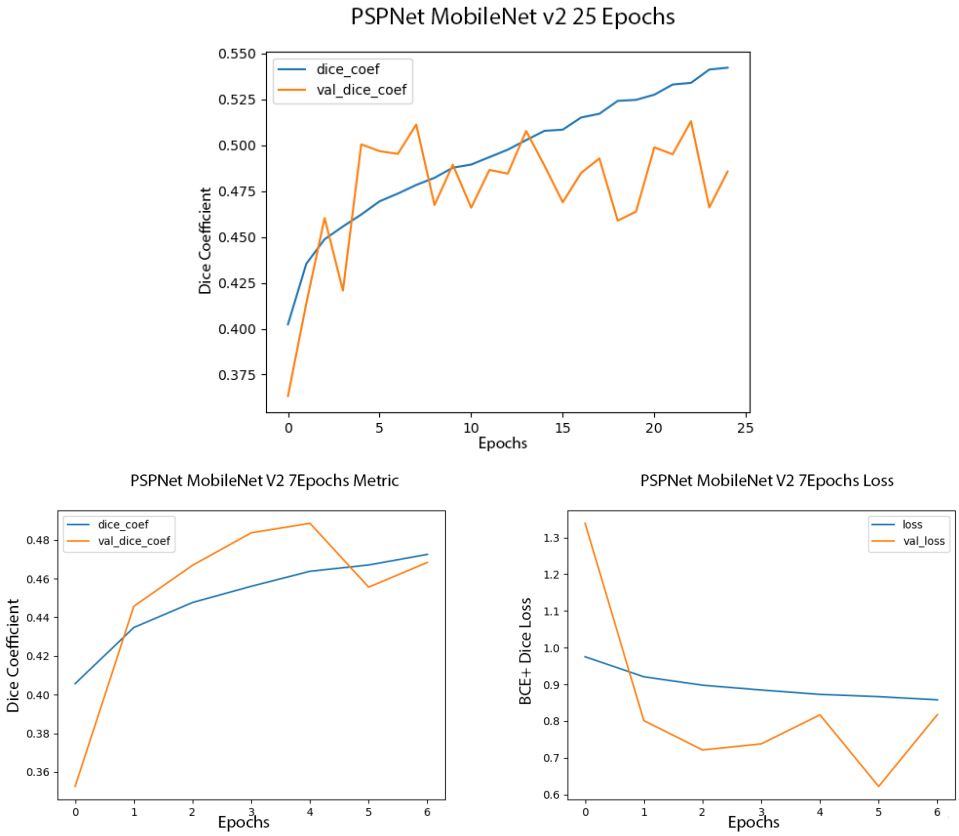


Figure 6.8: Mobilenetv2 backboned PSPNet 7 epochs.

Validation data set on 7 epochs reaches an approximate accuracy of 0.46 and loss increases to 0.8 as shown in figure 6.8.

InceptionResNetv2

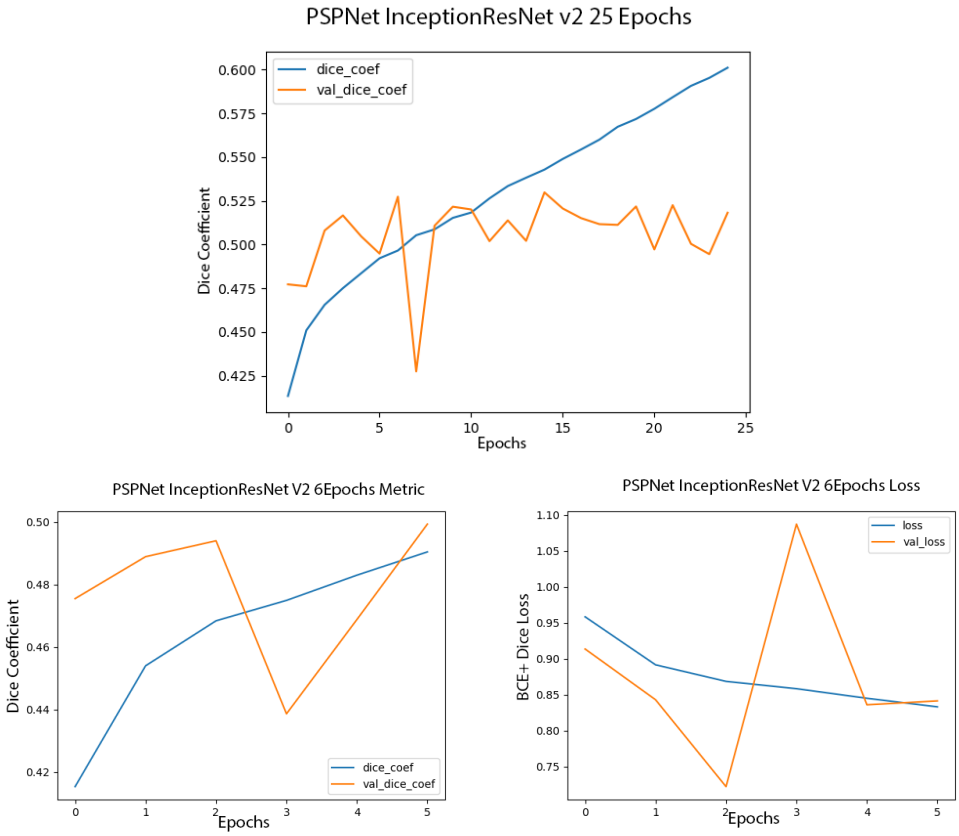


Figure 6.9: InceptionResNetv2 backboned PSPNet 6 epochs.

Validation data set on 6 epochs reaches an approximate accuracy of 0.56 and loss is stable at 0.85 as shown in Figure 6.9.

### 6.2.2 PSPnet Epochs and Accuracy

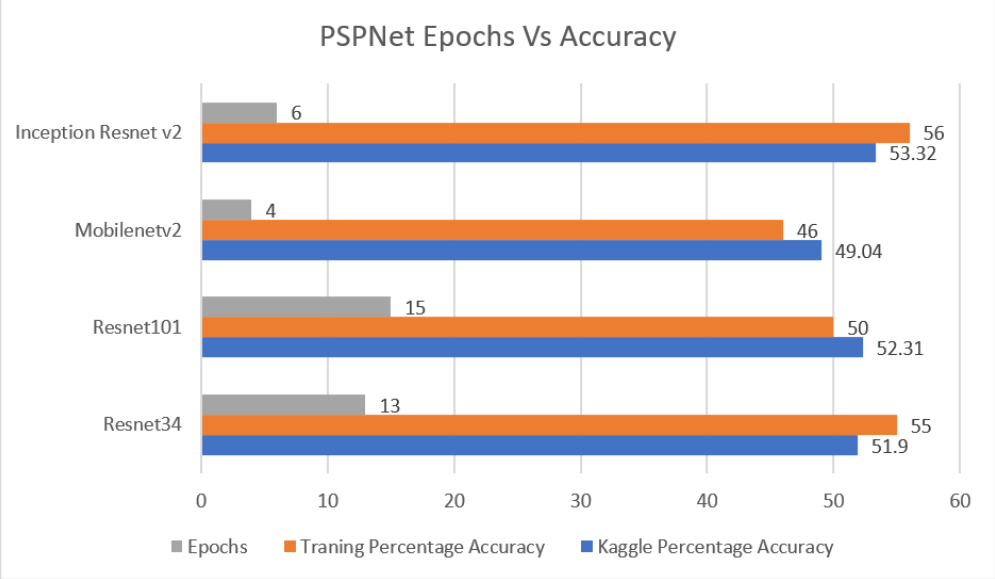


Figure 6.10: PSPNet Comparison graph epochs vs accuracy.

According to Kaggle's score the winner in PSPNet architecture is Inception-ResNet-v2 followed by ResNet 101 (Figure 6.10). Additionally we see that the loss never decreased below 0.7 and actually reached a new high at 0.85, suggesting that either we need more training or the data set is not containing similar featured mapping.

## 6.3 FPN

### 6.3.1 FPN Test and validation Scores

ResNet34

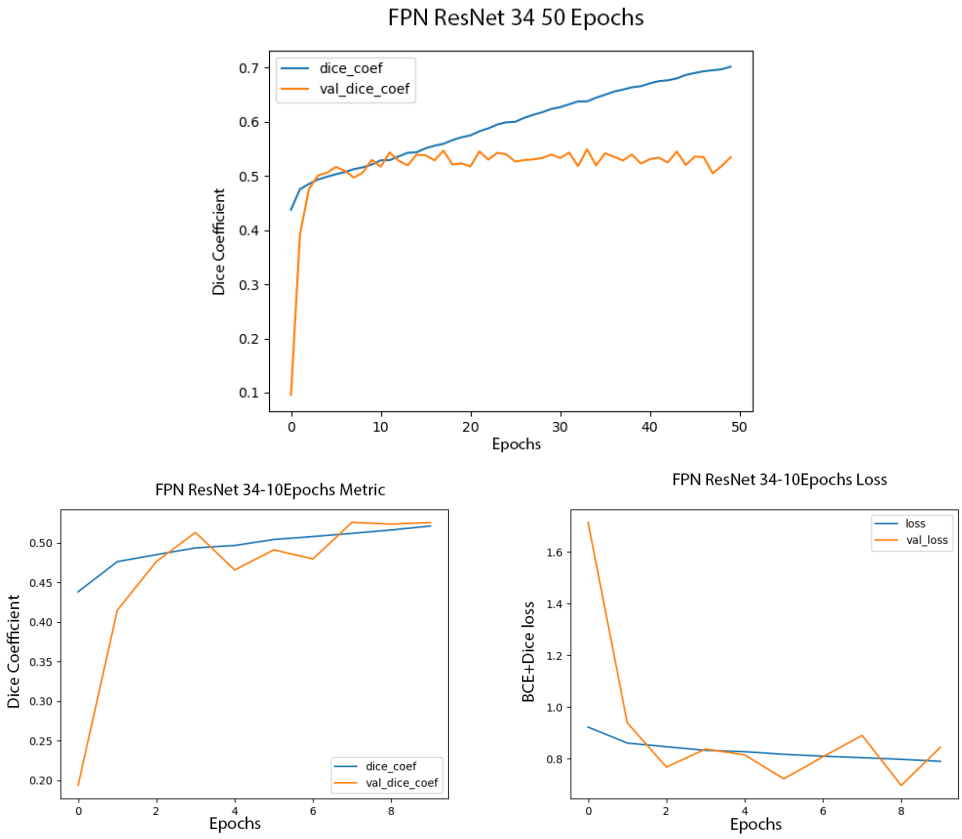


Figure 6.11: ResNet-34 backboned FPN 10 epochs

Validation data set on 10 epochs reaches an approximate accuracy of 0.50 and loss fluctuating at 0.8, shown in Figure 6.11.

ResNet101

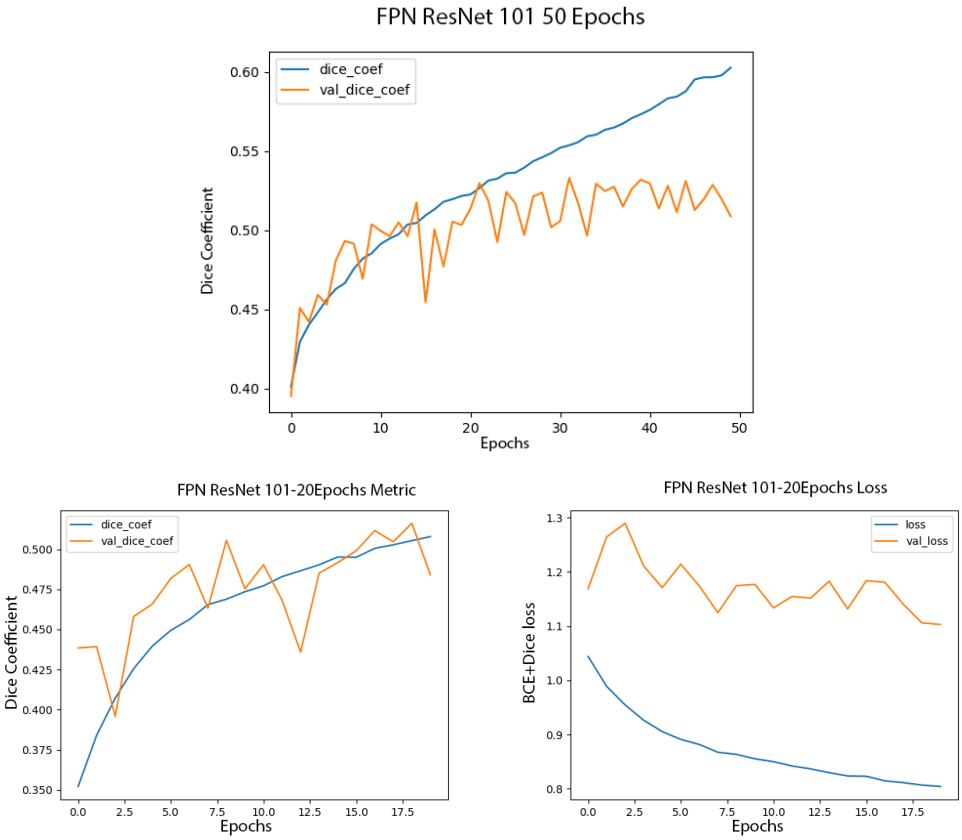


Figure 6.12: ResNet-101 backboned FPN 20 epochs.

Validation data set on 20 epochs reaches approximate accuracy of 0.47 and loss finally reaching 1.1 (Figure 6.12).

MobileNetv2

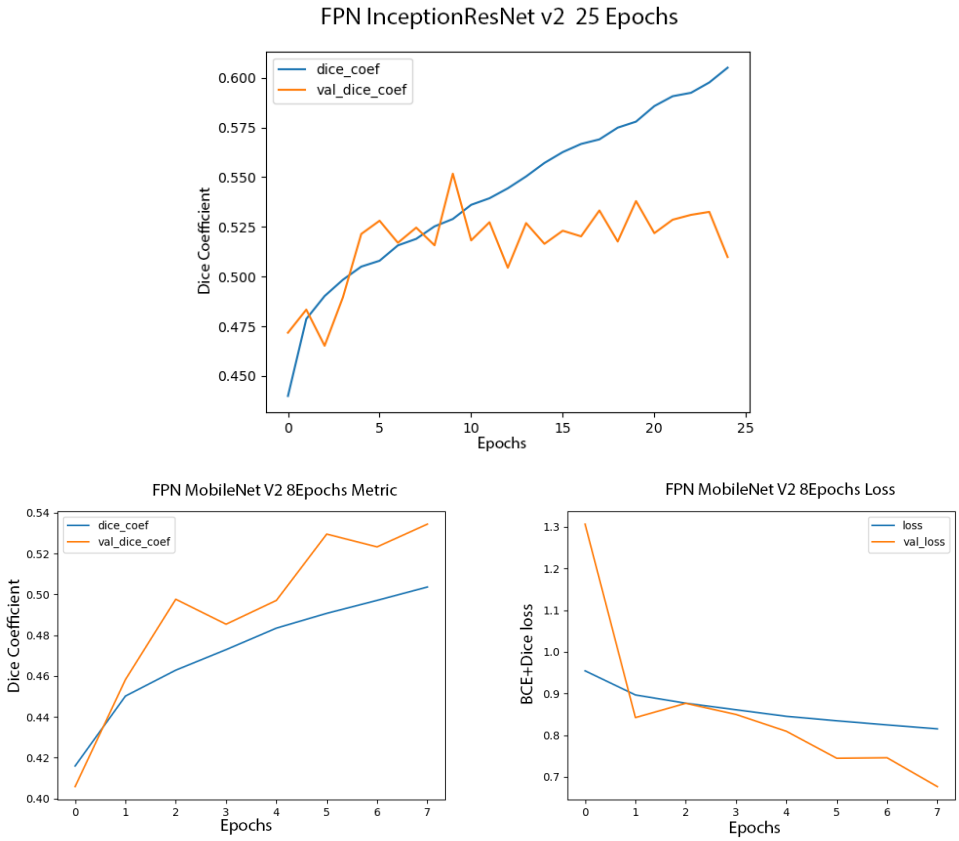


Figure 6.13: MobileNetv2 backboned FPN 8 epochs.

Validation data set on 8 epochs reaches an approximate accuracy of 0.54 and loss decreases to finally 0.65 as shown in figure 6.13.

InceptionResNetv2

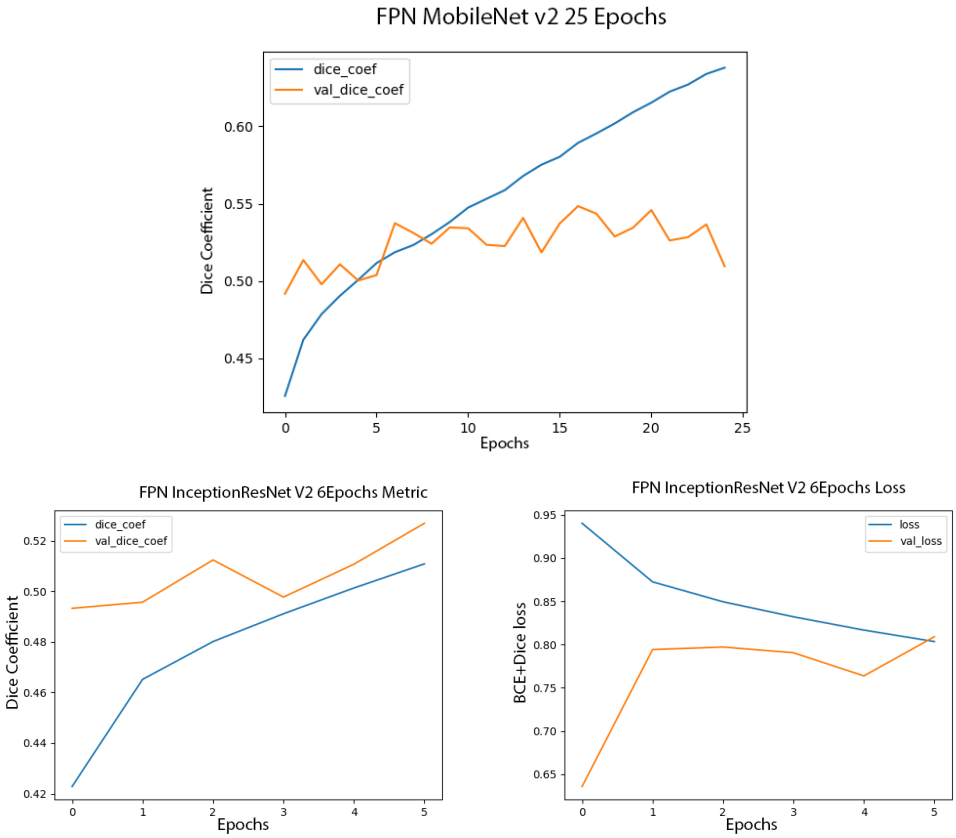


Figure 6.14: InceptionResNetv2 FPN PSPNet 6 epochs.

Validation data set on 6 epochs reaches an approximate accuracy of 0.53 and loss increases finally reaching 0.85 as shown in Figure 6.14.

### 6.3.2 FPN Epochs and Accuracy

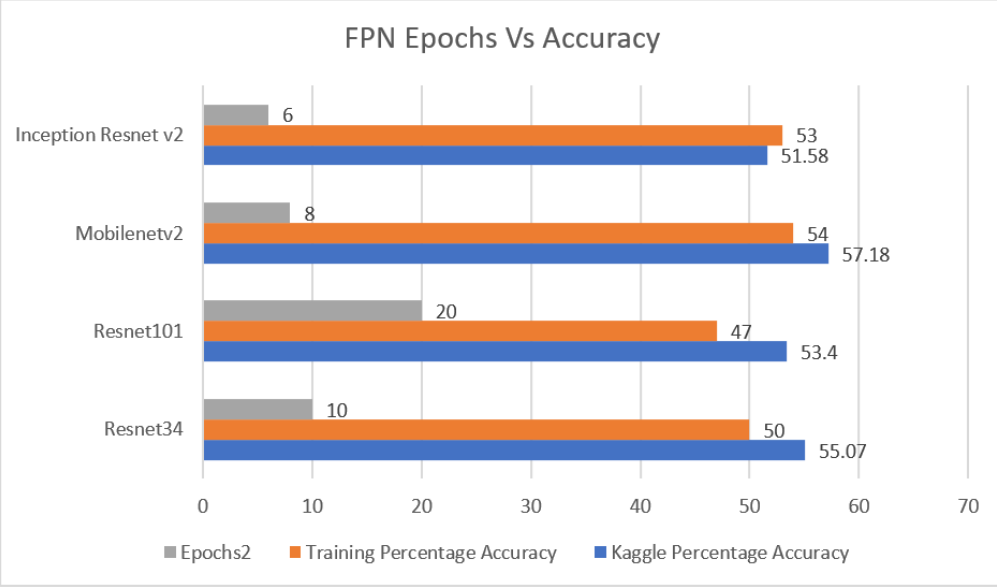


Figure 6.15: ResNet-34 backboned PSPNet 13 epochs.

According to Kaggle's score in FPN architecture the winner is MobileNetv2 followed by ResNet34 as seen in Figure 6.15. We also see from figure 6.11, 6.12, 6.13 and 6.14 that the loss fell below 0.7 unlike the previous models.

# CHAPTER 7

# CONCLUSIONS AND FUTURE WORK

After processing the normalized images on mesoscale cloud data, which were made available by the Max Plank institute of meteorology and Kaggle.com, we can conclude that the accuracy results were broadly similar. The three models UNet, FPN and PSPNet in combination with the four encoders ResNet-34, ResNet-101, MobileNet-v2 and Inception-ResNet-v2 performed almost the same.

In second place Inception-ResNet-v2 backbone stands, while reaching a high approximate accuracy rate of 55 % in UNet. The image data set is noisy by itself when we consider that each mask was drawn by approximate three participants and then the union of these masks were taken. Inception-ResNet-v2 claims to perform well on such data sets, which can be verified by this study.

In the first place stands FPN with MobileNet-v2 back end model. This infers to us that the pixel arrangement in the masks contains high semantic locality and a particular feature which was captured perfectly by the FPN. This result seems to verify FPN's structural model which we described in Chapter 2. Also having the MobileNet-v2 back end means that this models will train faster because of the depth wise separable convolution.

In future, the performance of all these models might be boosted by two to three percent by creating a precision recall and area under curve function to be tested for validation data set on each epoch. This function would have triggered a stop on the training right when the model reaches it's peak accuracy rather than when we visually identify model's peak. This would have also saved the extra computation time caused by running model twice to find out right epochs to avoid over fitting. Another thing that could improve model's perform

is removal of images containing significant small masks. Since the dice coefficient depend on the area of prediction, if we remove the images with small masks the model would only train on bigger masks. This will indirectly predict the bigger region more accurately in the test images leading to a boost in dice coefficient and accuracy score.

# BIBLIOGRAPHY

# BIBLIOGRAPHY


[1] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[2] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," 2016.

[3] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," 2016.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," 2018.

[6] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," 2016.

[7] S. Rasp, H. Schulz, S. Bony, and B. Stevens, "Combining crowd-sourcing and deep learning to explore the meso-scale organization of shallow convection," 2019.

[8] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras, 2015.

[9] P. Yakubovskiy, "Segmentation models," 2019. [Online]. Available: https://github.com/qubvel/segmentation_models