PIXEL CLASSIFICATION USING U-NET
(Semantic Segmentation)

A Thesis to be Submitted in Partial Fulfilment
of the Requirements for the Degree of

Master of Technology

by

Ayush Chouhan
Roll No : CS2001

under the supervision of

Dr. Ashish Ghosh
Professor, Machine Intelligence Unit

Indian Statistical Institute
Kolkata-700108, India

CERTIFICATE

This is to certify that the dissertation titled "**PIXEL CLASSIFICATION USING U-NET (Semantic Segmentation)**" submitted by **Ayush Chouhan** to the Indian Statistical Institute, Kolkata, in partial fulfillment for the degree of **Master of Technology in Computer Science (Specialized in Data Science)** is a bonafide record of work carried out by him under my supervision and guidance. The dissertation has fulfilled all the requirements as per the regulations of the institute.

A. Ghosh.
11. 7. 2022

Ashish Ghosh
Professor,
Machine Intelligence Unit,
Indian Statistical Institute,
Kolkata-700108, India

Acknowledgement

I wish to express my sincere appreciation to my supervisor, Prof. Ashish Ghosh. I am delighted to have this project under his careful guidance. Special thanks go to the Machine Intelligence Unit (MIU) of ISI Kolkata for providing technical support and other essential resources. Senior research scholars specially Subhadip Boral, Anwesha Law gave me quality of time from their busy academic schedule so that I can complete this project conveniently on time. Last but not the least, I want to thank my parents, who constantly encouraged me in the study from my childhood and motivated me even in my bad days.

Ayush Chouhan
CS2001,
M.Tech in Computer Science,
Indian Statistical Institute,
Kolkata-700108, India

Abstract

The rapid advances in Deep Learning (DL) techniques have allowed rapid detection, localisation, and recognition of objects from images or videos. DL techniques are now being used in many different applications related to agriculture and farming and medical Science Images. In this work we are using Deep Learning techniques such as unet,pretrained unet and apply on CWIF data set for Anomaly Detection and anomaly is weed and on Electron Microscopy Dataset we are detecting mitochondria in hippocampus region of the brain we evaluate our model using different losses and evaluation metrics at the same time also telling the drawback and advantages of different models. If we can detect the images in the crops we can use different machines that can be used for real time detection and removal of weed from the field Our technology can distinguish between crop and weed plants in commercial fields where crop and weed grow near to one another and can tolerate plant overlap. Automated crop/weed discrimination allows for targeted weed treatment in weed management tactics to reduce expense and adverse environmental effects.
The images of hippocampus region of the brain to detect mitochondria in the images and give lable to each pixel will it belong to mitochondria or not

Keywords *Weed Detection; Segmentation ;Unet; Autoencoder; VGG16; Resnet; Inception; Focal Loss; Dice coefficient; Mean IOU; Jacard coefficient; Transfer Learning*

Contents

List of Figures

1 Introduction

In order to manage and remove weeds effectively and increase yields, recognition and prevention of weeds can be crucial. Although both weeds and crops have similar colours ('green-on-green'), and because their shapes and textures can be extremely similar throughout the growth period, weed detection in crops from pictures is naturally a difficult and tough challenge. Also, a crop in one setting can be considered a weed in another. Our task is to detect weed in given input vegetation mask and classifying EACH PIXEL of the input image to be weed or crop. Here we are using supervised as well as semi supervised learning techniques. For this segmentation task we are using the U-net architecture as our base model and different metrics as our evaluation metrics. In the second technique we are using attention U-net and apply model on two different datasets to get to know the power of attention U-net and how attention is useful in dealing with the pixel belonging to rare class. In the third technique we are using weights of pre- trained network like VGG16,Resnet,Inception as a part of encoder in u-net and doing segmentation task In forth technique we are using pre-trained autoencoder which is trained on our specific task(Images) and used that as apart of encoder in the u-net architecture.

Now based on our observation of different models and evaluation we are concluding that using a pre-trained autoencoder as a part of u-net architecture will definitely beneficial for the task of weed detection and detetcting mitochondria in brain microscopy images

Section 2 contains a formal problem definition

Section 3 contains the terminologies and various definition for explanation of the keywords

Section 4 contains a brief explanation of the dataset. *section 5* Data preprocessing and Augmentation *section 6* contains a brief discussion of some relevant research works.

Section 7 our contribution for the given problem definition

Section 8 -9 performance evaluation metrics and loss function used for the task

section 10 Contains all the results and performance of different models(Technique)

section 11 contains all the references of this thesis.

2 Problem Definition

Given images which contains crops(Carrot) and a weed in a single image and the ground truth of the input images which will actually tell which part belong to weed and which part belong to crops ,with the help of ground truth we will use supervised learning techniques to evaluate our model performance which will tell us that which technique is good for classification between crop and weed in images at pixel level, we have to detect which part of the image belong to crop and which part will belong to weed and the remaining will be background

For this we will trained a u-net where input is a image and output is a vegetation mask Now we have the vegetation mask we will give this vegetation mask as a input to a different U-net trained specifically for segmenting task and now we will do our prediction and based on that and evaluate the model performance

For the performance we are using evaluation metrics as f1-score and Iou-Score as well as Mean-Iou at Some points

In the next task we are taking the images of hippocampus region of brain and detecting mitochondria in it and doing the pixel level classification(PLA)

3 Terminologies

- **Segmentation**: In plain English, segmentation is the process of giving pixels labels. Each pixel or piece of a picture assigned to the same category has a unique label.

- **Weed Detection**:Unmechanized weed control is one of the current issues in agriculture, and weed detection systems are significant solutions. Weed identification also contributes to the reduction or elimination of pesticide use, minimising the negative effects of agriculture on the environment and human health, and enhancing sustainability.

- **U-net**: It is currently one of the methods used the most frequently for semantic segmentation tasks. It is a fully convolutional neural network built with a smaller training sample size in mind. It is an upgrade over the current FCN, which Jonathan designed as "Fully convolutional networks for semantic segmentation." Long et al. in (2014).

- **Multiclass Segmentation**: we instead of binary we have multiple class to segment or when given a pixel we have more then 2 classes to segment

- **VGG16** : A convolutional neural network with 16 layers is called VGG-16. The ImageNet database contains a pretrained version of the network that has been trained on more than a million photos. The trained network can identify 1000 different object types in pictures.

- **Resnet34**: Resnet34 is a modern image classification model that consists of a 34 layer convolutional neural network. This model was previously trained using the ImageNet dataset, which contains more than 100,000 images across 200 different classifications.. However, it is different from traditional neural networks in the sense that it takes residuals from each layer and uses them in the subsequent connected layers (similar to residual neural networks used for text prediction)

- **Inception-V3**: On the ImageNet dataset, it has been demonstrated that the picture recognition model Inception v3 can achieve higher than 78.1 percent accuracy.

The model is the result of numerous concepts that have been established by various researchers over the years.

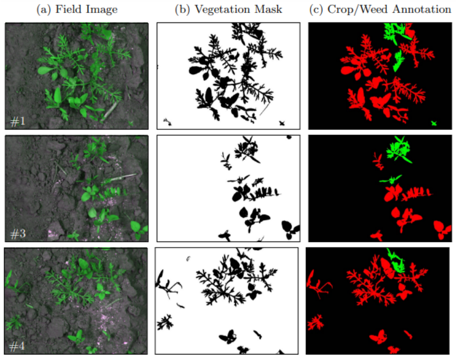
- **Autoencoder**: A particular kind of neural network called an autoencoder is capable of learning a compressed representation of raw data. Encoder and decoder sub-models make up an autoencoder. The input is compressed by the encoder, and the decoder makes an effort to reconstruct the input from the encoder's compressed form.

- **Transfer Learning**: Transfer learning (TL) is a machine learning (ML) research subject that is concerned with the storage of knowledge obtained while resolving one problem and its subsequent application to another similar but unrelated problem. For instance, understanding acquired in learning to recognise. one type of crop could apply when trying to recognize other type of crop.

4 Dataset Explaination

4.1 Dataset-1

Figure below displays example images from the dataset together with all annotations. The following section describes the content of the dataset, the acquisition parameters as well as the exact format of the image data and metadata.[**2**]



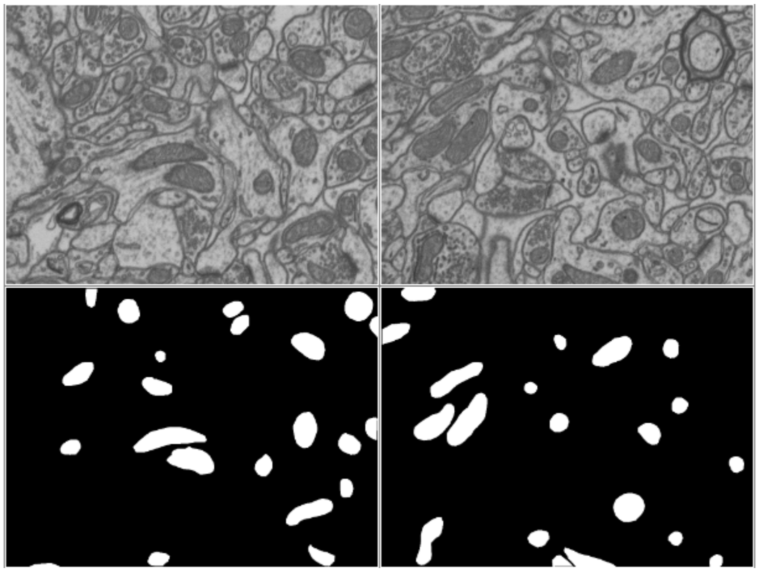
(a)

Figure 1: Crops,Vegetation mask and ground truth containing weed(Yellow) and carrot(Red)

The dataset contain total 60 images and corresponding vegetation masks and also corresponding Ground truth images which contain back colour pixel belongs to background red pixel belong to crop and yellow pixel belong to weed part

4.2 Dataset-2

Their are total images size 1065x2048x1536 volume of tif stake of Electron microscopy and the corresponding mask images which are total 1980 images(After breaking them into patches) The dataset available for download on this webpage represents[**1**] a 5x5x5µm section taken from the CA1 hippocampus region of the brain.Although our line of research was primarily motivated by the need to accurately segment mitochondria and synapses, other structures are of interest for neuroscientists such as vesicles or cell boundaries.



(a)

Figure 2: Above images is brain images and the below part is the corresponding masks in which white part belong to mitochondria

5 Data Preprocessing And Augmentation

5.1 Dataset 1 - Augmentation and preprocessing

As we are doing our task of pixel classification with use of deep learning techniques and deep learning task required a lot of data but data for weed detection contain only 60 images and corresponding masks so we need to do data augmentation,for this task we are flipping and rotating the images and corresponding ground truth images so that their will be no discrepancy when we are training our supervised models. By doing this we will be generating 500 images and corresponding masks so total we have 1000 images to work with

5.2 Dataset 2 - Augmentation and preprocessing:

In this data set we have total volume of $1065 \times 2048 \times 1536$ and have a similar tif stake of masks. we will divide the images into 256×256 with a step size of 256 so we have no overlapping, we also doing the same for ground truth

By this process we have 1980 images of size 256×256 and masks in total which is good for deep learning tasks.

6 Related Work

When focusing solely on leaves, techniques based on shape, colour, and texture characteristics have shown promise in differentiating between various leaf kinds. With an average accuracy of 85%, Beghin et al. [1] categorise leaves from 10 species based on form and texture. . Kumar et al. [2] developed a Northeastern United States smartphone application for categorising tree leaves These methods have a common input—an image of a flat leaf taken against a largely uniform background—with other works[**beghin2010shape**]. Machine vision can be used in agriculture for intelligent weed control, however often, commercial field settings do not call for the use of single leaf images.In order to control the application of herbicides, remote sensing has been effectively used to predict weed densities and dispersion on a field level [5]. However, a far finer scale of plant classification is required for the precision agricultural activities that are taken into consideration here [6]. Camera-based sensing can be used to classify and identify individual plants as well as groupings of a few plants on the ground. A robot and computer vision system developed by Hemming et al. [7] reliably detects 51 to 95 percent of plants based on the colour and shape of segmented plants in top-down images.They draw the conclusion that segmenting plants into distinct entities is challenging and requires more research. Similar characteristics are used by Astrand Baerveldt [8] to classify segmented plants. They test their system in greenhouse tests with big plants (approximately 5 cm in diameter), but they don't give quantitative information on categorization performance. To facilitate crop / weed distinction, leaf segmentation has also been studied in field settings [9, 10].Neto et al. [11] introduce a method for segmenting leaves that works well for convex leaves but not for other leaf shapes (like carrots considered here). They come to the conclusion that these circumstances require greater inquiry..

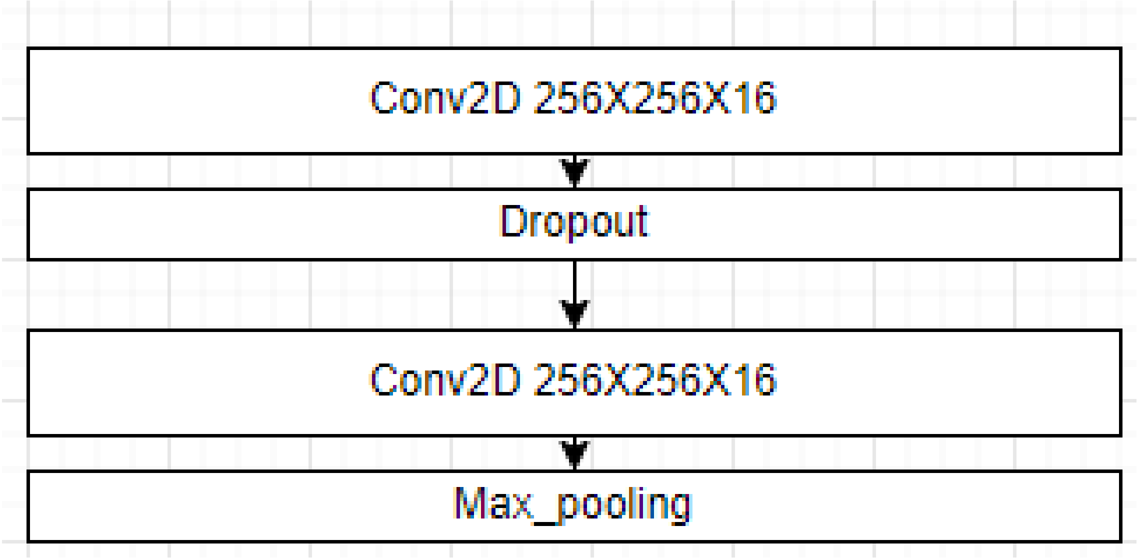
Another work done by Sebastian Haug and Andreas Michaels They evaluate their approach using a collection of pictures taken by a field robot operating autonomously on an organic carrot farm. Cross-validating with a leave one out strategy while using photos from our dataset as input results in an average classification accuracy of 93.8.%.

The data of electron microscopy to detecting mitochondria is a open source data for different uses so no pixel level classification accuracy is given

7 Our Contribution

7.1 Segmentation using U-net

One of the main advantages of using U-Net is its ability to yield relatively good results on pixel-labelling tasks with limited dataset images.
One Layer of Encoder block will be shown below:



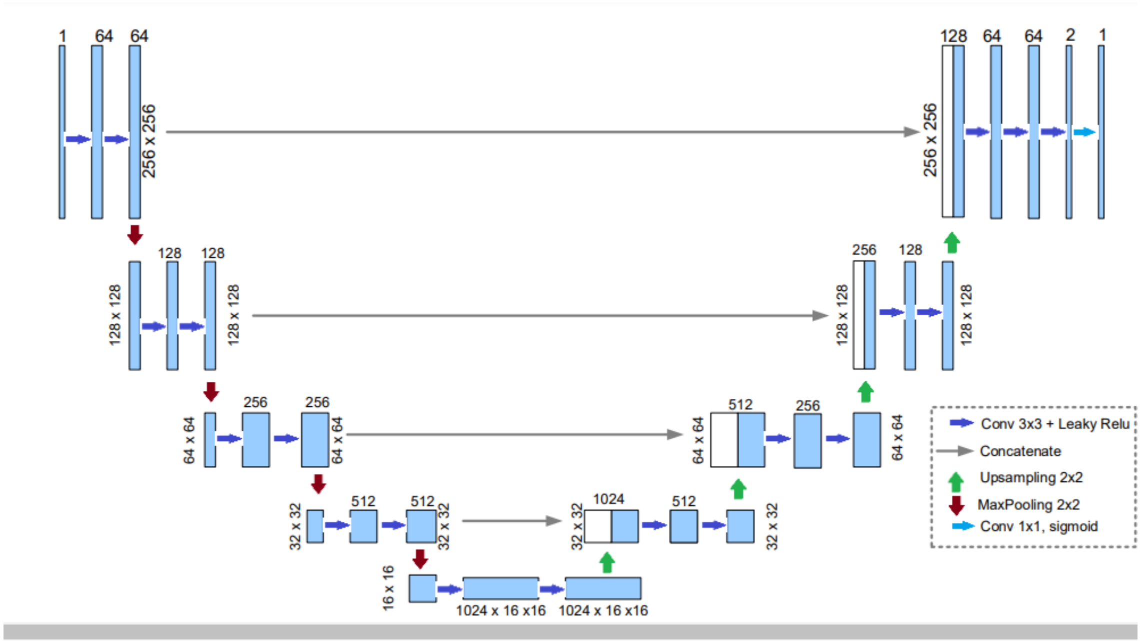
(a)

Figure 3: One Layer of encoder block

$$256X256 \rightarrow 128X128 \rightarrow 64X64 \rightarrow 32X32 \rightarrow 16X16 \rightarrow 32X32 \rightarrow 64X64 \rightarrow 128X128 \rightarrow 256X256$$

The actual Unet architecture is shown above, when we are dealing with multiclass Classification we use **Softmax** in the last layer of Decoder block of Unet.When we are dealing with Binary classification we will be using **Sigmoid** this we shoud we have to remember in every time when we are training our model

Classical U-Net architecture includes the down-sampling and up-sampling paths. The down-sampling path has four blocks. Each block does two 3x3 unpadded convolutions, a rectified linear unit (ReLu)[Relu], a 2x2 max pooling operation with stride 2, and a rectified linear unit (ReLu)[Relu]. However, Leaky ReLu with a 0.3 value was utilised in its



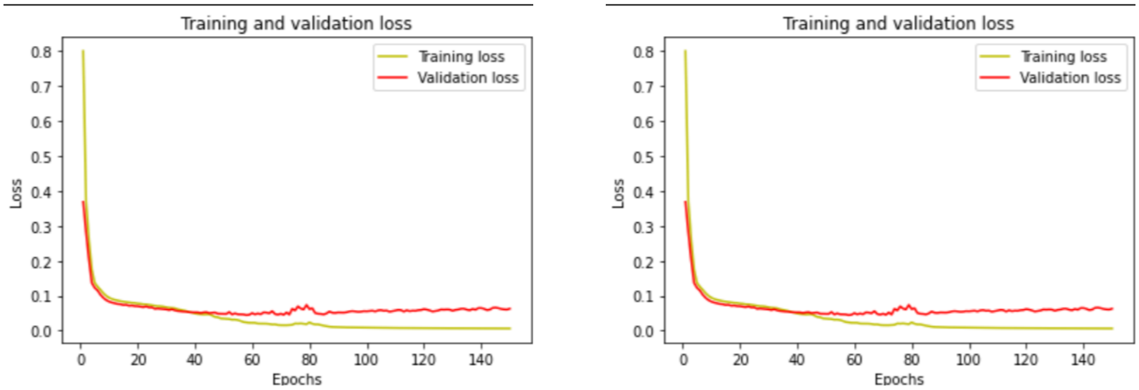
(a)

Figure 4: (a)U-net architecture used in experiment

place. ReLu unit may reach a point where it solely outputs zeros, which would prohibit some neural network components from learning. The number of feature channels doubles with each block.. The up-sampling path has four blocks as well. Every block along this route upsamples a feature map, performs a 2x2 convolution to cut the number of feature channels in half, concatenates the upsampled feature map with the downsampled feature map, and then does two 3x3 unpadded convolutions and a leaky ReLu. In this method, a 256x256 image with a 256x256 mask is sent to the input layer. The final layer of the network maps the feature vector to two classes using 1x1 convolution and the softmax function. Architecture has a total of 29 levels. There are 20 convolutional layers total, followed by Leaky ReLu, 4 max-pooling down-sampling layers, 4 up-sampling up-sampling layers, and an output layer. Adam optimizer was used for optimization, with categorical cross-entropy as the loss function..

Some times we also used Focal Loss and negative of jacard coefficient and negative of Dice coefficient and also used Binary focal loss when we are dealing with binary classification

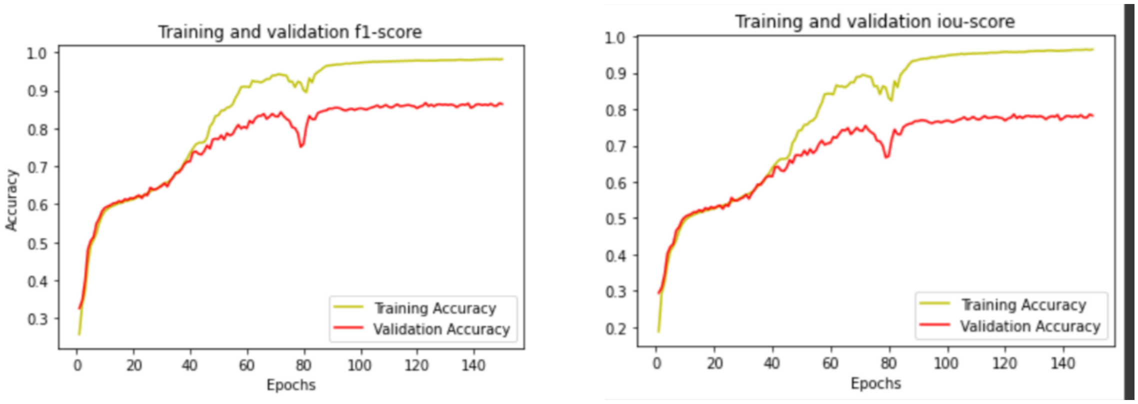
Results :



(a) Training and Validation loss

(b) Training and Validation Accuracy

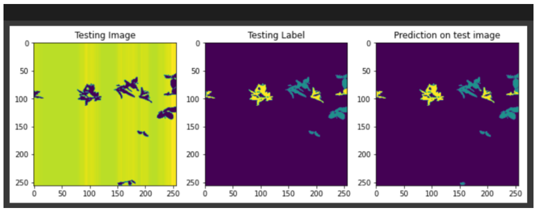
Figure 5: Training and Validation loss and Accuracy on weed detection



(a) Training and Validation f1-score

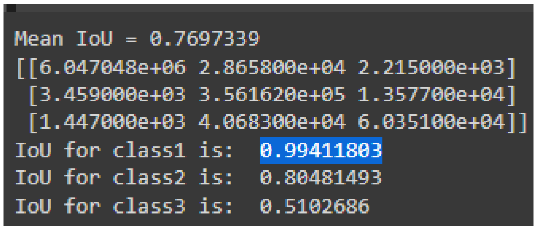
(b) Training and Validation Iou-score

Figure 6: Training and Validation f1-score and Iou-score on weed detection



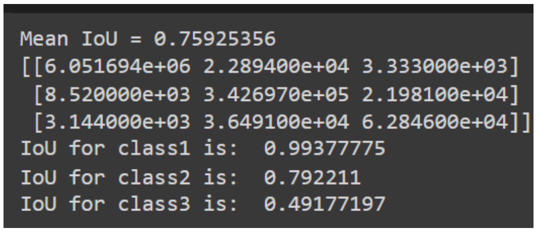
(a) Prediction

Figure 7: Prediction of Multi class segmentation model on weed detection



```
Mean IoU = 0.7697339
[[6.047048e+06 2.865800e+04 2.215000e+03]
 [3.459000e+03 3.561620e+05 1.357700e+04]
 [1.447000e+03 4.068300e+04 6.035100e+04]]
IoU for class1 is:  0.99411803
IoU for class2 is:  0.80481493
IoU for class3 is:  0.5102686
```

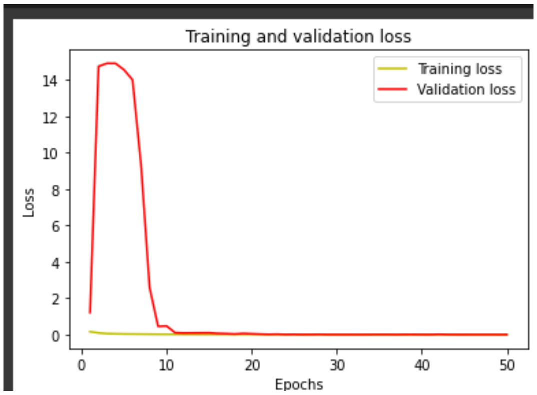
(a) Mean Iou $\gamma = 2$



```
Mean IoU = 0.75925356
[[6.051694e+06 2.289400e+04 3.333000e+03]
 [8.520000e+03 3.426970e+05 2.198100e+04]
 [3.144000e+03 3.649100e+04 6.284600e+04]]
IoU for class1 is:  0.99377775
IoU for class2 is:  0.792211
IoU for class3 is:  0.49177197
```

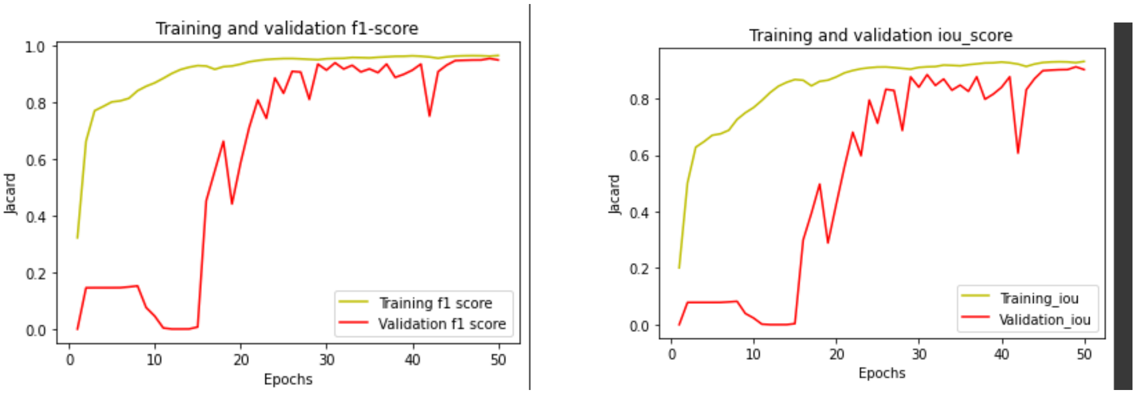
(b) Mean Iou $\gamma = 5$

Figure 8: Mean iou at $\gamma = 2and\gamma = 5$ on weed detection



(a) Training and Validation loss

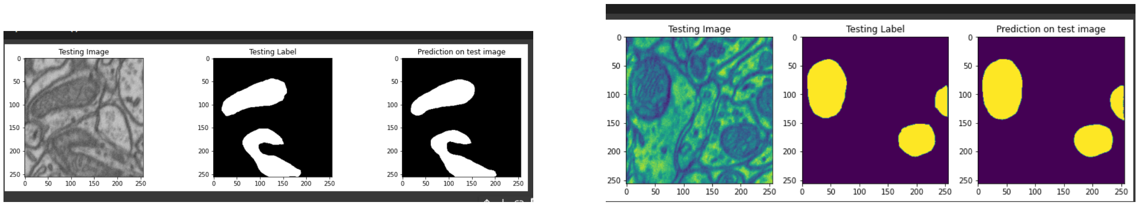
Figure 9: Training and Validation loss Electron Microscopy Dataset



(a) Training and Validation f1-score

(b) Training and Validation Iou-score

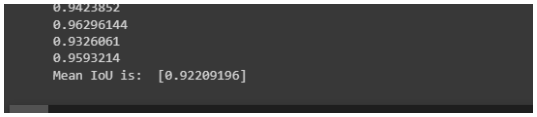
Figure 10: Training and Validation f1-score and Iou-score on Electron Microscopy Dataset



(a) prediction

(b) prediction

Figure 11: Final prediction on electron microscopy images



```
0.9423852
0.96296144
0.9326061
0.9593214
Mean IoU is:  [0.92209196]
```

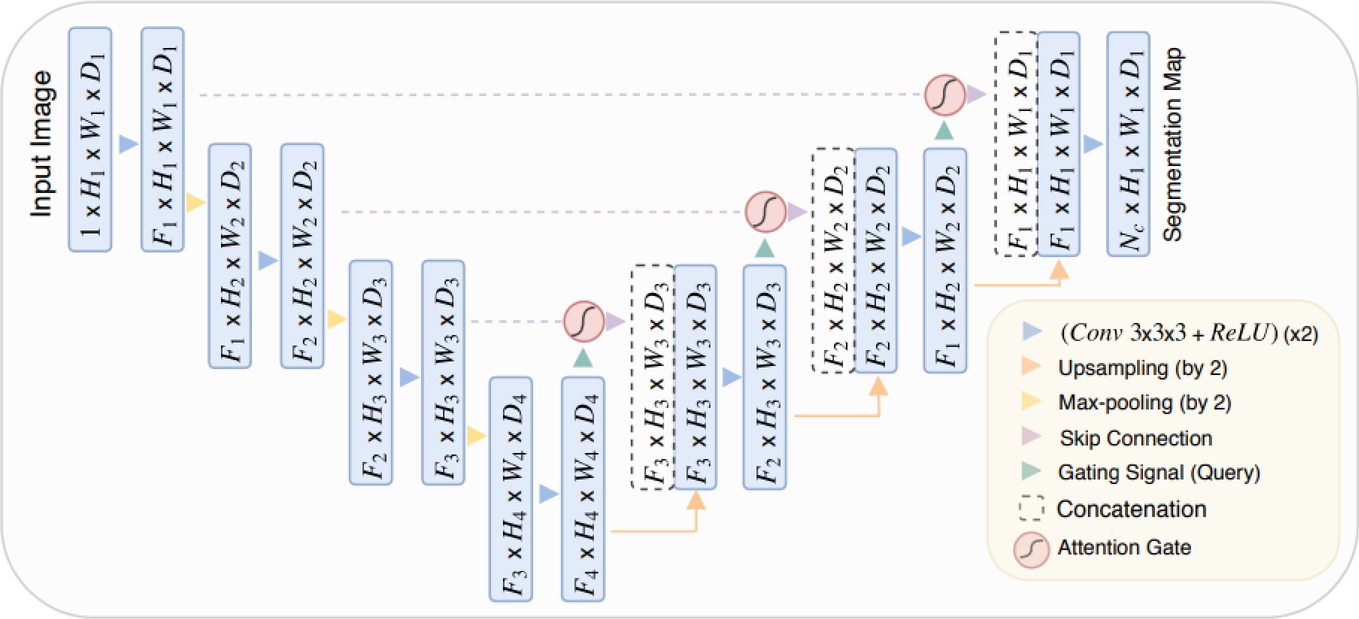
(a) Mean Iou $\gamma = 2$

Figure 12: Mean iou at $\gamma = 2$ on Electron Microscopy Dataset

7.2 Attention-Unet-Learning Where to Look

Attention Gates:For tasks like image captioning [1], machine translation [2, 30], and classification [11, 31, 32], AGs are frequently employed in natural image analysis, knowledge graphs, and language processing (NLP). Prior research investigated attention-maps by analysing the gradient of output class scores in relation to the input image. On the other hand, trainable attention is imposed by design and divided into hard- and soft-attention. Model training is more challenging because hard attention [21], such as iterative region proposal and cropping, is frequently non-differentiable and requires on reinforcement learning for parameter updates. [36] use recursive hard attention to find anomalies in chest X-ray data. Contrarily, soft attention uses normal back-propagation and is probabilistic without the need for Monte Carlo sampling.For instance, additive soft attention has recently been utilised to classify images [11, 32] and is used in sentence-to-sentence translation [2, 29]. In [10], which was the top performer in the ILSVRC 2017 image classification challenge, channel-wise attention is used to emphasise significant feature dimensions. To end the reliance on outside gating information, self-attention approaches [11, 33] have been suggested. For instance, [33] uses non-local self attention to identify long-range dependencies. Self-attention is used to achieve class-specific pooling in [11, 32], leading to more precise and reliable image classification performance.

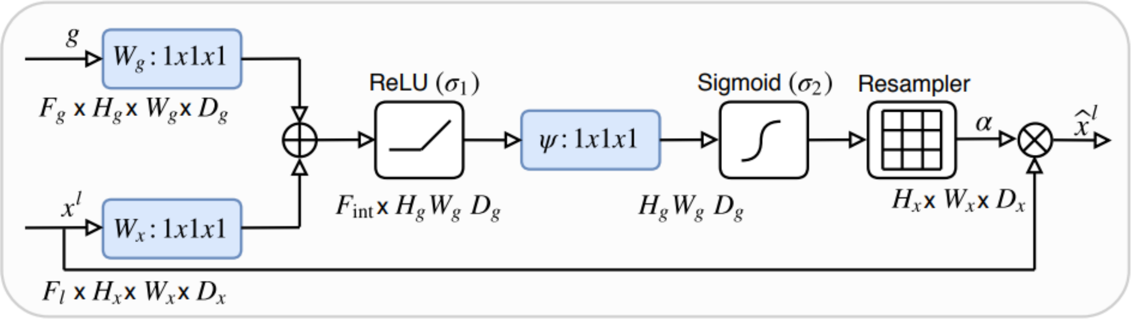
In this research, we present a novel self-attention gating module that may be used for dense label predictions in CNN-based standard image analysis models. We also look at how AGs can be used for medical image analysis, namely for picture segmentation. The following is a summary of this work's contributions::


Input Image
$1 \times H_1 \times W_1 \times D_1$
$F_1 \times H_1 \times W_1 \times D_1$
$F_1 \times H_2 \times W_2 \times D_2$
$F_2 \times H_2 \times W_2 \times D_2$
$F_2 \times H_3 \times W_3 \times D_3$
$F_3 \times H_3 \times W_3 \times D_3$
$F_3 \times H_4 \times W_4 \times D_4$
$F_4 \times H_4 \times W_4 \times D_4$
$F_3 \times H_3 \times W_3 \times D_3$
$F_3 \times H_3 \times W_3 \times D_3$
$F_2 \times H_3 \times W_3 \times D_3$
$F_2 \times H_2 \times W_2 \times D_2$
$F_2 \times H_2 \times W_2 \times D_2$
$F_1 \times H_2 \times W_2 \times D_2$
$F_1 \times H_1 \times W_1 \times D_1$
$F_1 \times H_1 \times W_1 \times D_1$
$N_c \times H_1 \times W_1 \times D_1$
Segmentation Map
(Conv 3x3x3 + ReLU) (x2)
Upsampling (by 2)
Max-pooling (by 2)
Skip Connection
Gating Signal (Query)
Concatenation
Attention Gate

(a)

Figure 13: (a)A block diagram of the proposed Attention U-Net segmentation model. Input image is progressively filtered and downsampled by factor of 2 at each scale in the encoding part of the network $(e.g.H4 = \frac{H_1}{8})$. N_c denotes the number of classes.Attention gates (AGs) filter the features propagated through the skip connections.Schematic of the AGs is shown in Figure below. Feature selectivity in AGs is achieved by use of contextual information (gating) extracted in coarser scales.

Attention Gates in U-Net Model: To draw attention to important features that are sent over skip connections, the suggested AGs are implemented into the conventional U-Net design (see Figure 5). In order to distinguish between irrelevant and noisy responses in skip connections, information retrieved from coarse scale is used in gating. To merge just pertinent activations, this is done just prior to the concatenation step. Additionally, both during the forward pass and the reverse pass, AGs filter neuron activations. During the backward pass, gradients coming from background regions are given a lower weight.



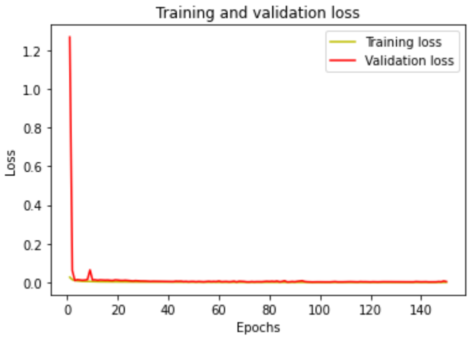
(a)

Figure 14: (a)Schematic of the proposed additive attention gate (AG). Input features(x^l) are scaled with attention coefficients (α) computed in AG. Spatial regions are selected by analysing both the activations and contextual information provided by the gating signal (g)which is collected from a coarser scale. Grid resampling of attention coefficients is done using trilinear interpolation.

This enables shallower layer model parameters to be changed primarily depending on spatial areas that are important to a specific task. The following can be used to develop the update rule for layer "l" - 1's convolution parameters:

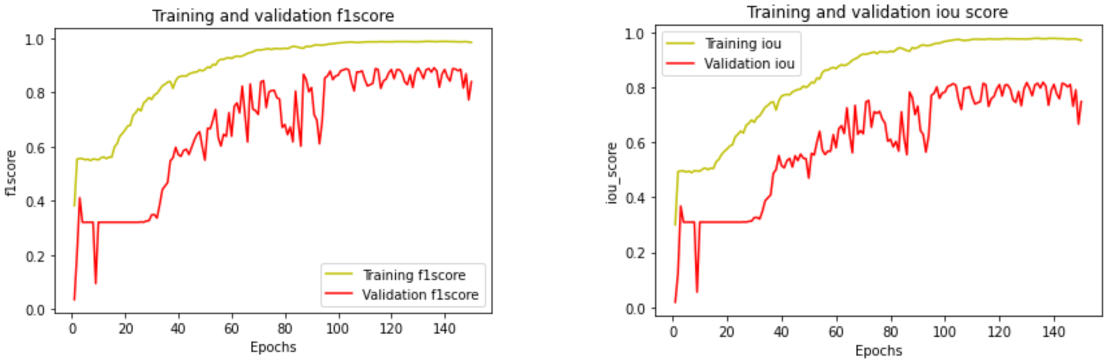
$$\frac{\partial(\hat{x}_i^l)}{\partial(\Phi^(l-1))} = \frac{\partial(\alpha_i^l f(x^(l-1)_i;\Phi^(l-1)))}{\partial(\Phi^(l-1))} = \alpha_i^l \frac{\partial(f(x^(l-1)_i;\Phi^(l-1)))}{\partial(\Phi^(l-1))} + \frac{\partial(\alpha_i^l)x_i^l}{\partial(\Phi^(l-1))}$$

The first gradient term on the right-hand side is scaled with α_i^l. In case of multi-dimensional AGs, corresponds to a vector at each grid scale. In each sub-AG, complementary information is extracted and fused to define the output of skip connection. To reduce the number of trainable parameters and computational complexity of AGs, the linear transformations are performed without any spatial support (1x1x1 convolutions) and input feature-maps are downsampled to the resolution of gating signal, similar to non-local blocks [33]. The corresponding linear transformations decouple the feature-maps and map them to lower dimensional space for the gating operation. As suggested in [11], low-level feature-maps, i.e. the first skip connections, are not used in the gating function since they do not represent the input data in a high dimensional space. We use deep-supervision



(a) Training and Validation loss

Figure 15: Attention Training and Validation loss on weed detection Dataset



(a) Training and Validation f1-score

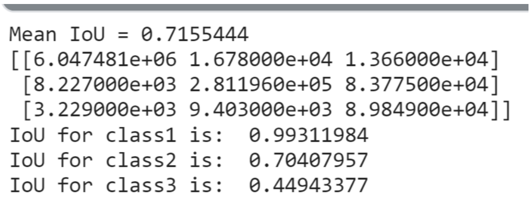
(b) Training and Validation Iou-score

Figure 16: Attention Training and Validation f1-score and Iou-score on weed detection Dataset

[16] to force the intermediate feature-maps to be semantically discriminative at each image scale. This helps to ensure that attention units, at different scales, have an ability to influence the responses to a large range of image foreground content. We therefore prevent dense predictions from being reconstructed from small subsets of skip connections

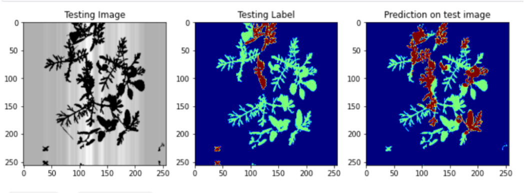
Observations : With the use of attention gates we will be increasing the time for a single epoch to complete but due to attention we will be getting the same f1 score(as in simple Unet) in less no.of epochs,so with the help of attention gets model is learning where to look for weed and mitochondria in respective images in less no of epochs so here we are saving our time and convergence will be faster

Results :



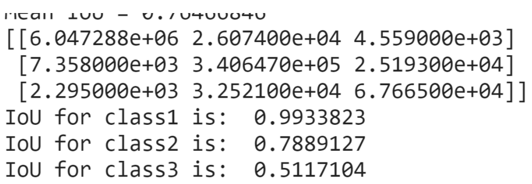
```
Mean IoU = 0.7155444
[[6.047481e+06 1.678000e+04 1.366000e+04]
 [8.227000e+03 2.811960e+05 8.377500e+04]
 [3.229000e+03 9.403000e+03 8.984900e+04]]
IoU for class1 is:  0.99311984
IoU for class2 is:  0.70407957
IoU for class3 is:  0.44943377
```

(a) Mean Iou $\gamma = 2$



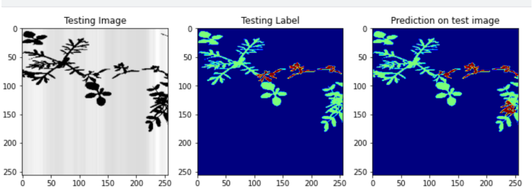
(b) Prediction

Figure 17: Mean iou and prediction at $\gamma = 2onweeddetectionDataset$



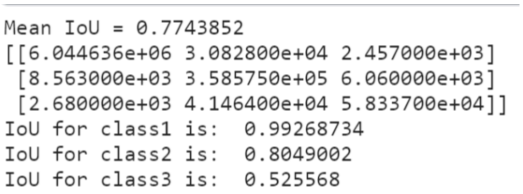
```
Mean IoU = 0.76400840
[[6.047288e+06 2.607400e+04 4.559000e+03]
 [7.358000e+03 3.406470e+05 2.519300e+04]
 [2.295000e+03 3.252100e+04 6.766500e+04]]
IoU for class1 is:  0.9933823
IoU for class2 is:  0.7889127
IoU for class3 is:  0.5117104
```

(a) Mean Iou $\gamma = 3$



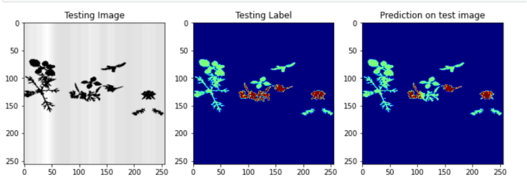
(b) Prediction

Figure 18: Mean iou and prediction at $\gamma = 3onweeddetectionDataset$



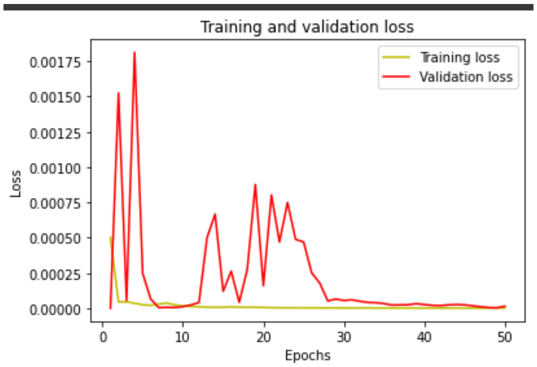
```
Mean IoU = 0.7743852
[[6.044636e+06 3.082800e+04 2.457000e+03]
 [8.563000e+03 3.585750e+05 6.060000e+03]
 [2.680000e+03 4.146400e+04 5.833700e+04]]
IoU for class1 is:  0.99268734
IoU for class2 is:  0.8049002
IoU for class3 is:  0.525568
```

(a) Mean Iou $\gamma = 5$

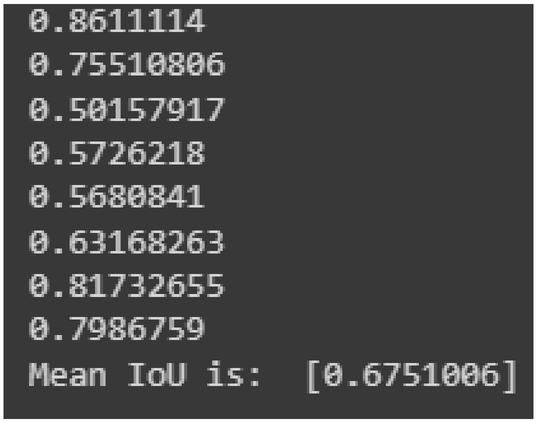


(b) Prediction

Figure 19: Mean iou and prediction at $\gamma = 5$ on weed detection Dataset



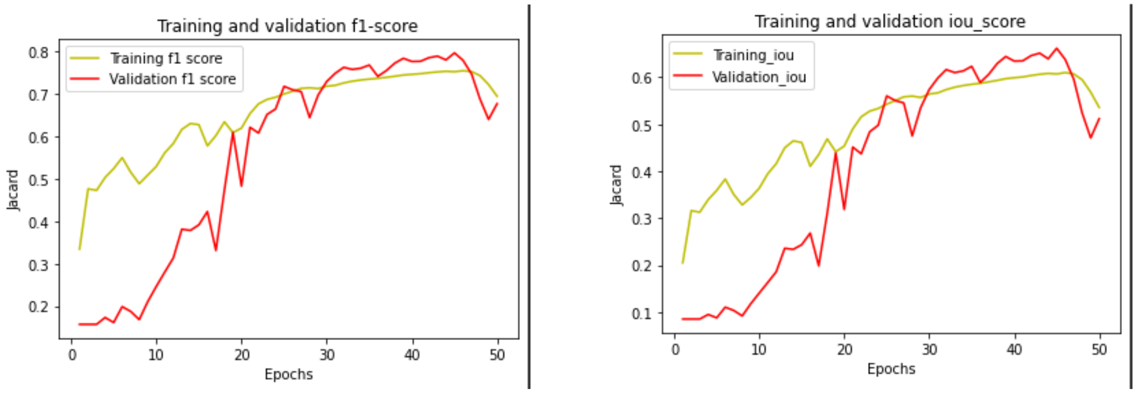
(a) Training and Validation loss



```
0.8611114
0.75510806
0.50157917
0.5726218
0.5680841
0.63168263
0.81732655
0.7986759
Mean IoU is:  [0.6751006]
```

(b) mean-iou

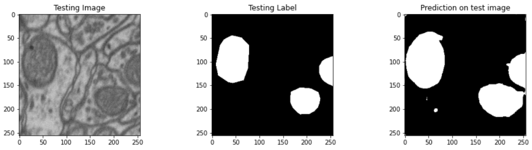
Figure 20: Training and Validation loss and Mean-iou on electron microscopy Dataset



(a) Training and Validation f1-score

(b) Training and Validation Iou-score

Figure 21: Training and Validation f1-score and Iou-score on electron microscopy Dataset



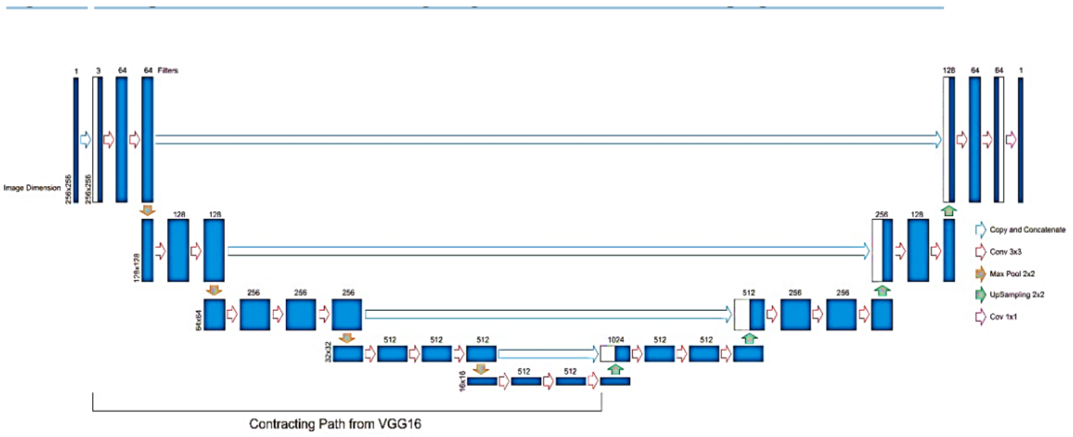
(a) Prediction

Figure 22: Prediction on electron microscopy using attention

7.3 VGG16 with Unet

One of the CNN architectures used exclusively for picture segmentation is called U-Net. The complexity of U-Net (this research contains 31,031,685 parameters) affects the execution time, and the U-Net design cannot be executed on some systems due to their low specification. To solve this issue, we suggested a novel model that combines the U-Net design with the VGG16 architecture to lower the layer and parameter of U-Net. VGG16 was chosen because it has fewer parameters than U-Net and is similar to the contracted layer of U-Net. Additionally, VGG16 already contains weights for easily accessible parameters, so we apply these weights to our new model. The expanding layer and contracting layer are typically present in several segmentation models. In this study, the VGG16 architecture was altered to mimic the U-Net architecture by including an expanding layer at the end of the VGG16 architecture that included numerous upsampling layers and convolution layers. This is carried out up until the model's overall architecture is symmetrical and takes the form of the letter U. As a result, the UNet-VGG16 model's design will consist of two layers: the VGG16, which serves as the contracting layer, and the expansion layer, which will be introduced later. The parameters will be decreased with this new architecture to 17,040,001 with about 2,324,353 trainable parameters. The UNet-VGG16 model with the Transfer Learning approach will be used to train the MRI brain tumour picture. By freezing the contraction layer in UNet-VGG16, this technique prevents the weighted layer from being modified when training data is executed. Instead, we make advantage of the VGG16 model's convolution layer weight. The objective is to shorten the computation time and shorten the model's training period. Figure 6 depicts the process visualisation of image segmentation under the new proposed architecture, whereas Figure 5 displays the architecture of the proposed model, namely UNet-VGG16 with Transfer Learning.

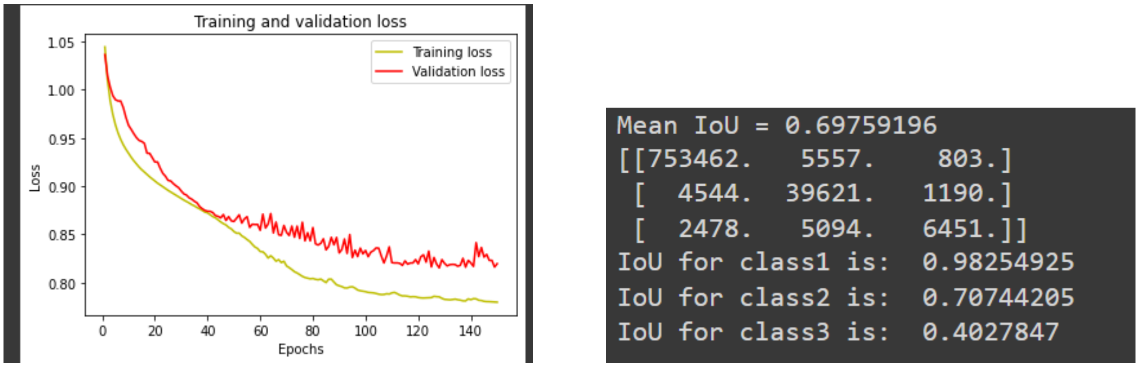
Observation: With the use of transfer learing and using vgg 16 pretarined weights which is trained on imagenet dataset so vgg 16 is locally optimized for a pearticular task ,here we are increasing the parameters but we are saving time in converging to best f1 score because of the concept of transfer learning we are converging more faster than we have attention gates plus Unet. here we are again saving more time with respect to no of epochs



(a)

Figure 23: The architecture of UNet-VGG16 with transfer learning

Results :

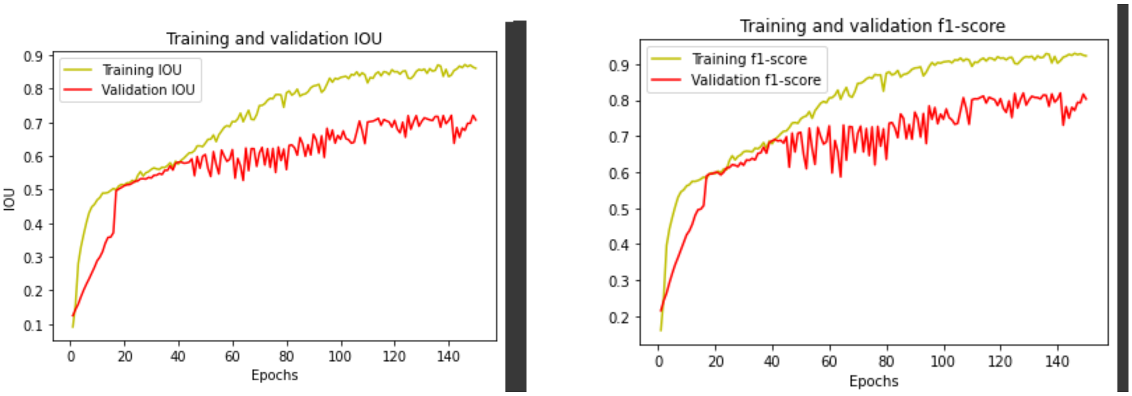


```
Mean IoU = 0.69759196
[[753462.   5557.    803.]
 [  4544.  39621.   1190.]
 [  2478.   5094.   6451.]]
IoU for class1 is:  0.98254925
IoU for class2 is:  0.70744205
IoU for class3 is:  0.4027847
```

(a) Training and Validation Loss

(b) Mean Iou-score

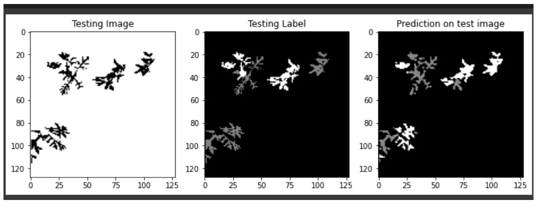
Figure 24: Training and Validation Loss and Mean iou



(a) Training and Validation Iou-score

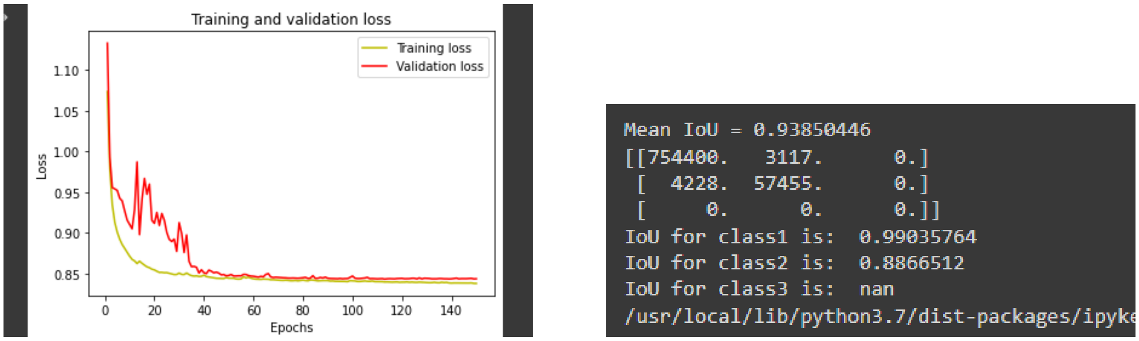
(b) Training and Validation f1-score

Figure 25: Training and Validation f1-score and Iou-score



(a) Predictions

Figure 26: Final Predictions

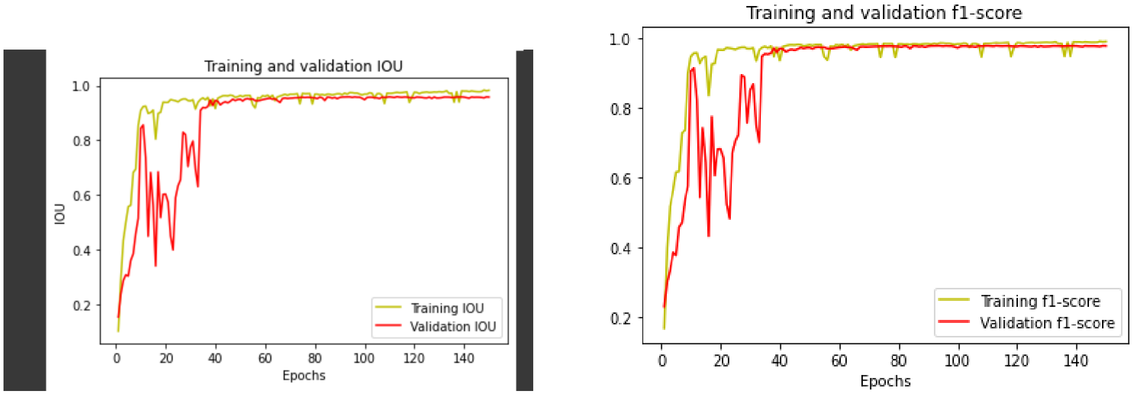


```
Mean IoU = 0.93850446
[[754400.   3117.      0.]
 [  4228.  57455.      0.]
 [     0.      0.      0.]]
IoU for class1 is:  0.99035764
IoU for class2 is:  0.8866512
IoU for class3 is:  nan
/usr/local/lib/python3.7/dist-packages/ipyke
```

(a) Training and Validation Loss

(b) Mean Iou-score

Figure 27: Training and Validation Loss and Mean iou on Electron Microscopy dataset



(a) Training and Validation Iou-score

(b) Training and Validation f1-score

Figure 28: Training and Validation f1-score and Iou-score on Electron Microscopy dataset

7.4 Resnet with Unet

One of the problems ResNets solve is the famous known vanishing gradient. This is because when the network is too deep, the gradients from where the loss function is calculated easily shrink to zero after several applications of the chain rule. This result on the weights never updating its values and therefore, no learning is being performed.

With ResNets, the gradients can flow directly through the skip connections backwards from later layers to initial filters.

Take H(x) as an underlying mapping that can be fitted by a few stacked layers (but not necessarily the complete net), with x indicating the inputs to the first of these layers. It is analogous to hypothesising that numerous nonlinear layers may asymptotically approximate the residual functions, i.e., $H(x) - x$, if one believes that they can do so for difficult functions (assuming that the input and output are of the same dimensions). Therefore, we explicitly allow these layers to approach a residual function $F(x) := H(x) - x$ rather than assuming that they will approximate H(x). Thus, the initial function becomes $F(x) + x$. Although (as expected) both forms should be able to asymptotically approximate the desired functions, the learning curve for each form may be different.

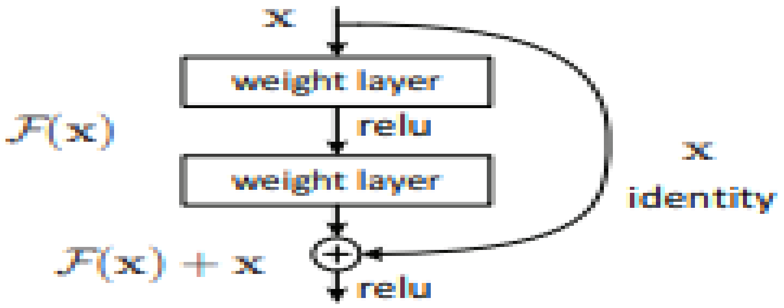

x
weight layer
relu
$\mathcal{F}(\mathbf{x})$
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 29: Residual learning: a building block. [**15**]

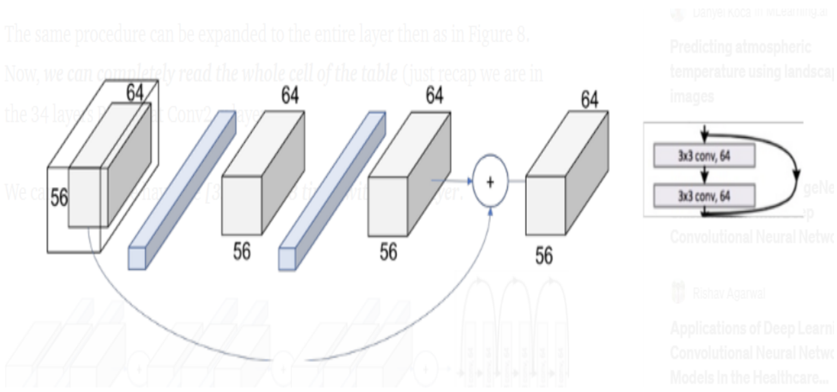
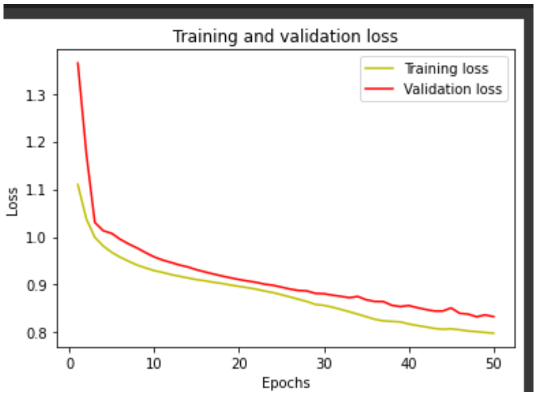


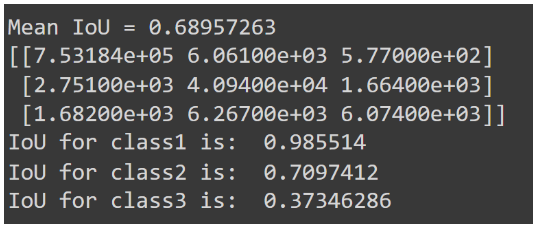
Figure 30: Residual learning: a building block. [**15**]

From this above observation taking a little bit further on that. Based on the plain network fig we insert shortcut connections which turn the network into its counterpart residual version. The identity shortcuts can be directly used when the input and output are of the same dimensions. In fig We can observe that the ResNet is made up of four layers with similar behaviour after one convolution and pooling stage (on orange). The same pattern is followed by each layer. By passing the input every two convolutions, they perform a $3x3$ convolution with fixed feature map dimensions (F) of 64, 128, 256, and 512, respectively. Additionally, during the entire lay, the measurements of width (W) and height (H) stay unchanged.

we are using resnet as a part of encoder in unet and using in unet so due to Pre trained weights in the encoder block of unet (resnet as backbone) and with the concept of transfer



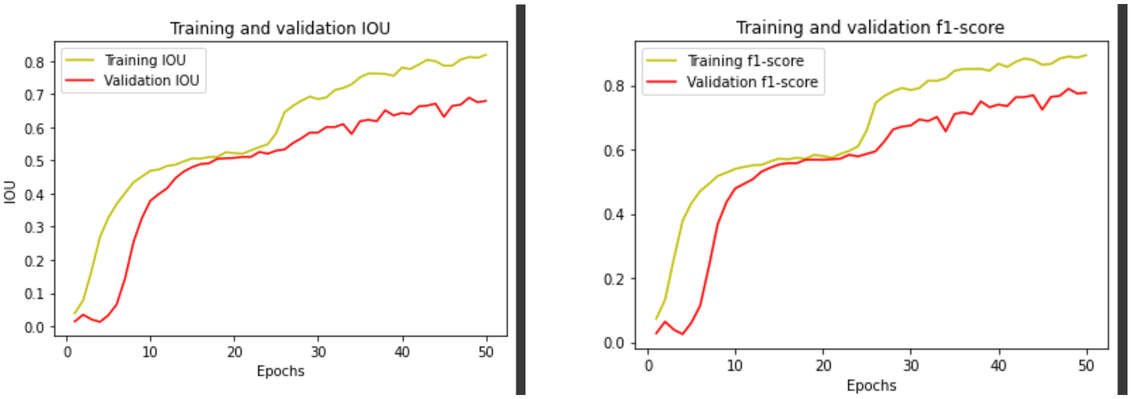
(a) Training and Validation Loss



```
Mean IoU = 0.68957263
[[7.53184e+05 6.06100e+03 5.77000e+02]
 [2.75100e+03 4.09400e+04 1.66400e+03]
 [1.68200e+03 6.26700e+03 6.07400e+03]]
IoU for class1 is:  0.985514
IoU for class2 is:  0.7097412
IoU for class3 is:  0.37346286
```

(b) Mean Iou-score

Figure 31: Training and Validation Loss and Mean iou on weed detection dataset

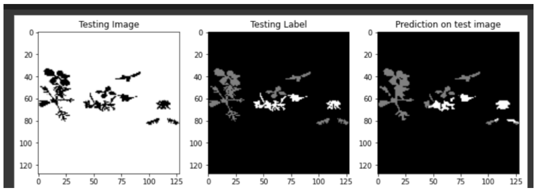


(a) Training and Validation Iou-score

(b) Training and Validation f1-score

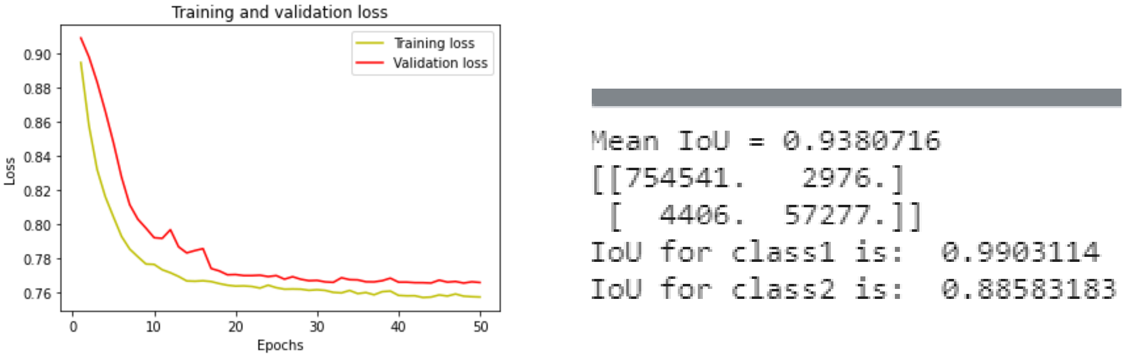
Figure 32: Training and Validation f1-score and Iou-score on weed detection dataset

learning the no. of epochs required to give the best f1 score we need less no of epochs



(a) Predictions

Figure 33: Final Predictions on weed detection dataset

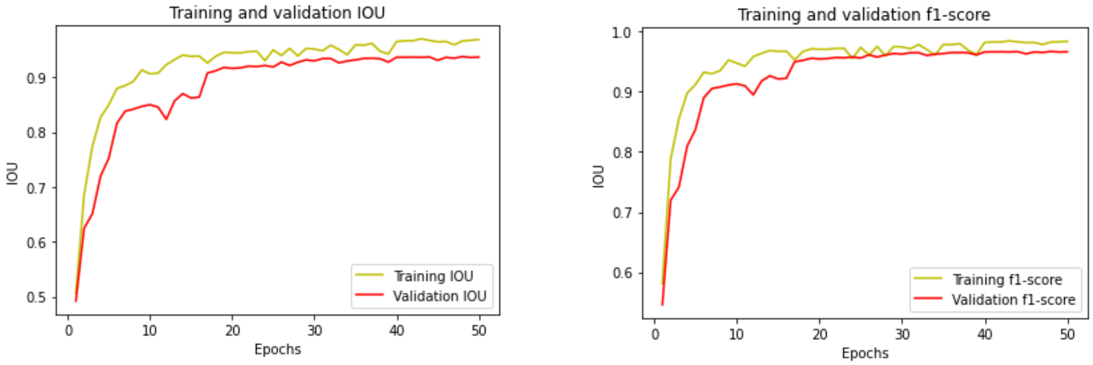


```
Mean IoU = 0.9380716
[[754541.   2976.]
 [  4406.  57277.]]
IoU for class1 is:  0.9903114
IoU for class2 is:  0.88583183
```

(a) Training and Validation Loss

(b) Mean Iou-score

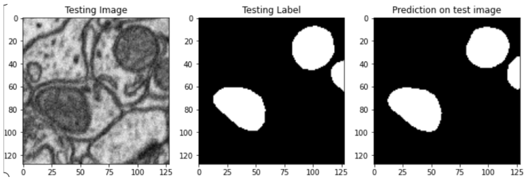
Figure 34: Training and Validation Loss and Mean iou on electron microscopy dataset



(a) Training and Validation Iou-score

(b) Training and Validation f1-score

Figure 35: Training and Validation f1-score and Iou-score on electron microscopy dataset



(a) Predictions

Figure 36: Final Predictions on electron microscopy dataset

7.5 Inception with Unet

Stage 1 and 2 The network starts with an image size of 224x224x3. Then it goes through a 1x1 Conv, 3x3 MaxPool, 1x1 Conv, 3x3 Conv, and a 3x3 MaxPool, and results in an image of size 192x28x28.

This part of the network is very similar to AlexNet, except that the image size in AlexNet is further reduced to 256x12x12.

Stage 3 As we see in Figure 6, stage 3 has two Inception blocks and in the end a Max Pool layer. But the inception blocks do not have the same channel allocation, as seen in the figure. Block 1 has 256 channels, whereas block 2 has 480 channels. So the input image of 192x28x28 now becomes 480x14x14, after the two inception blocks and the MaxPooling layer.

Stage 4 and 5 Stage 4 and 5 are very similar to stage 3. Stage 4 has 5 inception blocks followed by a MaxPool, and stage 5 has 2 inception blocks followed by a GlobalAveragePool.

So this is the Inception or GoogleLeNet architecture that was originally published. But later the architecture has been further improved in various different versions v2, v3, and v4.

Inception V2 —Add batch normalization

Inception V3 —Modified inception block (replace 5x5 with multiple 3x3 convolutions,replace 5x5 with 1x7 and 7x1 convolutions,replace 3x3 with 1x3 and 3x1 convolutions,have deeper stacks)

Observation: Inception,VGG16,Resnet34 as a encoder in Unet- In this experiment of transfer learning we are using we have observed that resnet34 will be acting as a good encoder for Unet and the model is giving its best accuracy in less no of epochs;henceforth we are converging faster

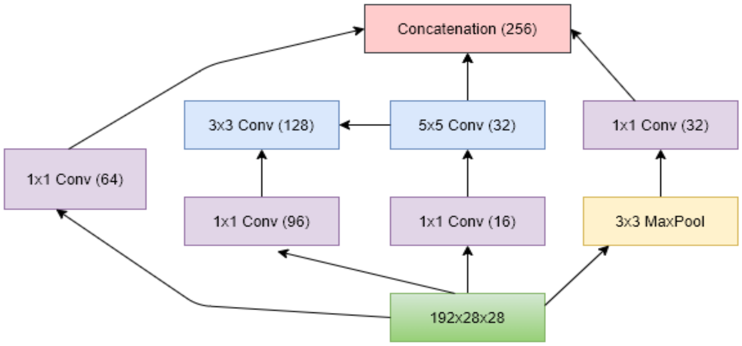
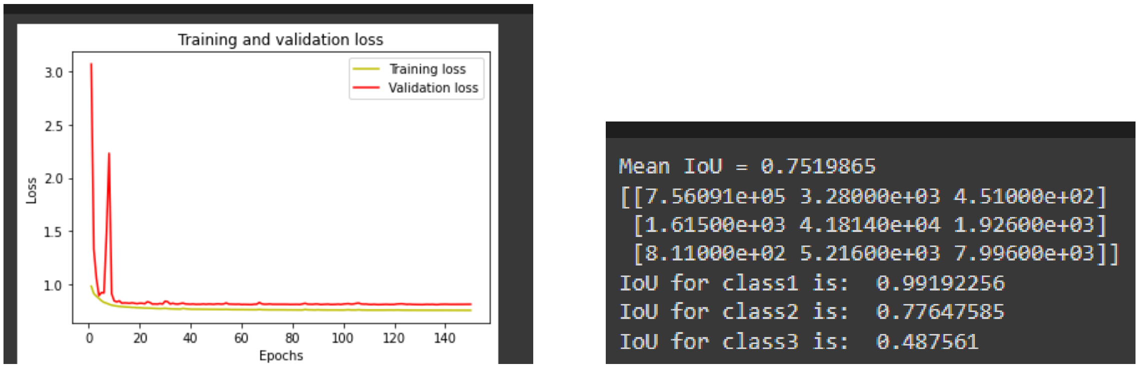


Figure 37: Inception Block

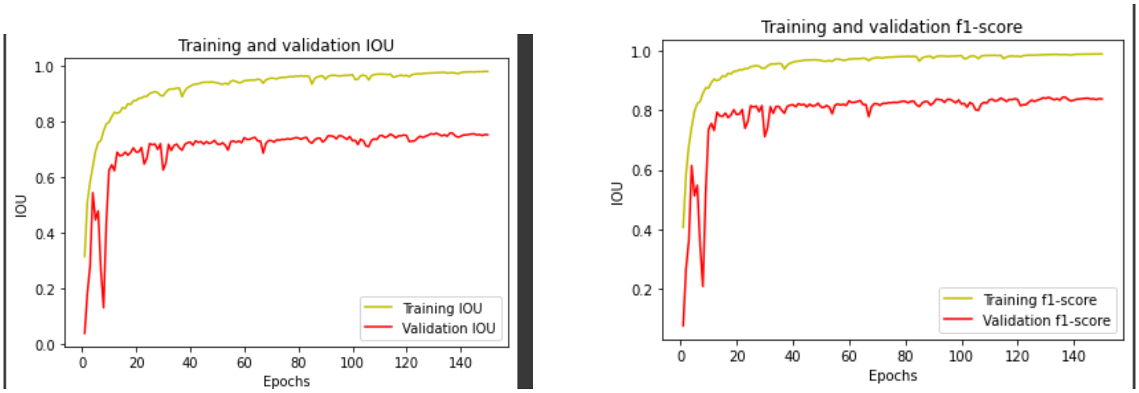


```
Mean IoU = 0.7519865
[[7.56091e+05 3.28000e+03 4.51000e+02]
 [1.61500e+03 4.18140e+04 1.92600e+03]
 [8.11000e+02 5.21600e+03 7.99600e+03]]
IoU for class1 is:  0.99192256
IoU for class2 is:  0.77647585
IoU for class3 is:  0.487561
```

(a) Training and Validation Loss

(b) Mean Iou-score

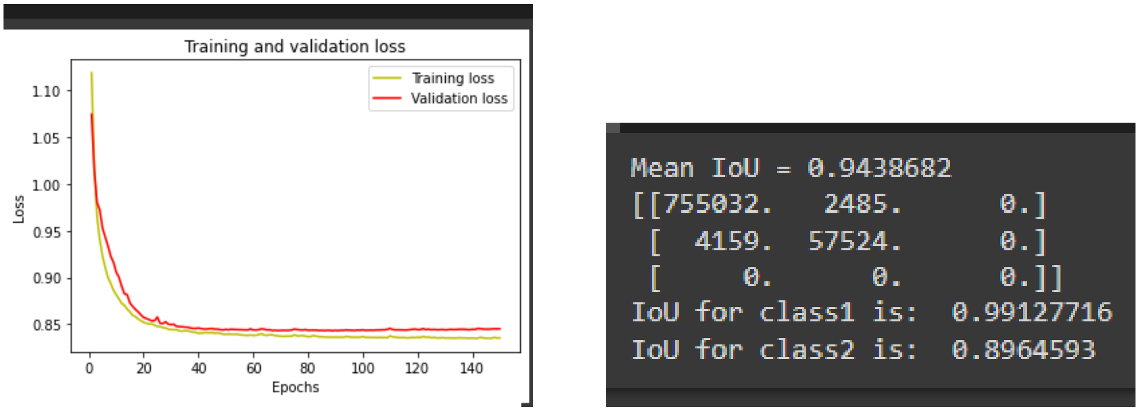
Figure 38: Training and Validation Loss and Mean iou on weed detection dataset



(a) Training and Validation Iou-score

(b) Training and Validation f1-score

Figure 39: Training and Validation f1-score and Iou-score on weed detection dataset

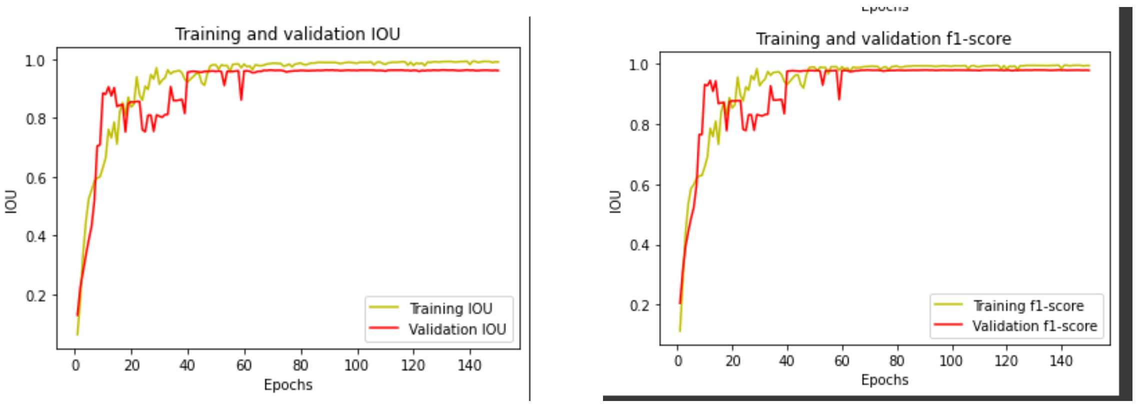


```
Mean IoU = 0.9438682
[[755032.   2485.      0.]
 [  4159.  57524.      0.]
 [     0.      0.      0.]]
IoU for class1 is:  0.99127716
IoU for class2 is:  0.8964593
```

(a) Training and Validation Loss

(b) Mean Iou-score

Figure 40: Training and Validation Loss and Mean iou on electron microscopy dataset



(a) Training and Validation Iou-score

(b) Training and Validation f1-score

Figure 41: Training and Validation f1-score and Iou-score on electron microscopy dataset

7.6 Autoencoder with Unet

Background: Autoencoder is an unsupervised artificial neural network that learns how to efficiently compress and encode data then learns how to reconstruct the data back from the reduced encoded representation to a representation that is as close to the original input as possible.

Autoencoder, by design, reduces data dimensions by learning how to ignore the noise in the data.

Autoencoder Components:

Autoencoders consists of 4 main parts:

1- Encoder: In which the model learns how to reduce the input dimensions and compress the input data into an encoded representation.

2- Bottleneck: which is the layer that contains the compressed representation of the input data. This is the lowest possible dimensions of the input data.

3- Decoder: In which the model learns how to reconstruct the data from the encoded representation to be as close to the original input as possible.

4- Reconstruction Loss: This is the method that measures measure how well the decoder is performing and how close the output is to the original input.

The training then involves using back propagation in order to minimize the network's reconstruction loss

Now we will train a auto encoder on our mask images and then we will use the weights of autoencoder and transfer these weights to the unet for the training and testing purpose by we can say that when we use vgg16,resnet and inception we are getting the same results in less no of epoches now if we train a auto encoder on our data set and train it and use weights in our unet architecture we can say that the model will converge faster

When we are using transfer learning at that time we are thinking that we will be transfering the weights of a model which is trained on imagnet dataset why we build a autoencoder and use the weights of encoder and populate it in the encoder of the Unet now train our unet for segmentation. Surperisingly the result we got after this process are not improved in terms of f1 score but we will be getting same value of f1-score in less no of epochs,so

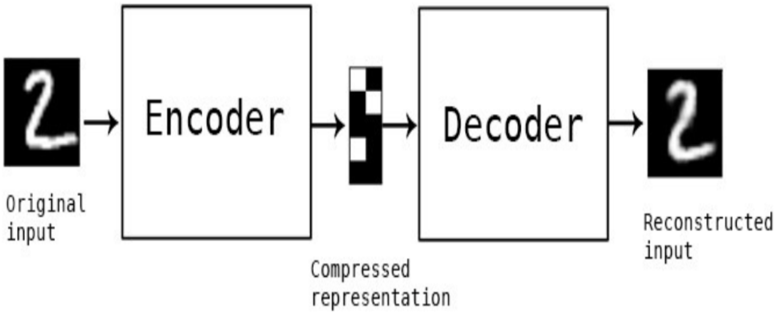


Figure 42: A autoenoder visualization

again we are saving time

Results :

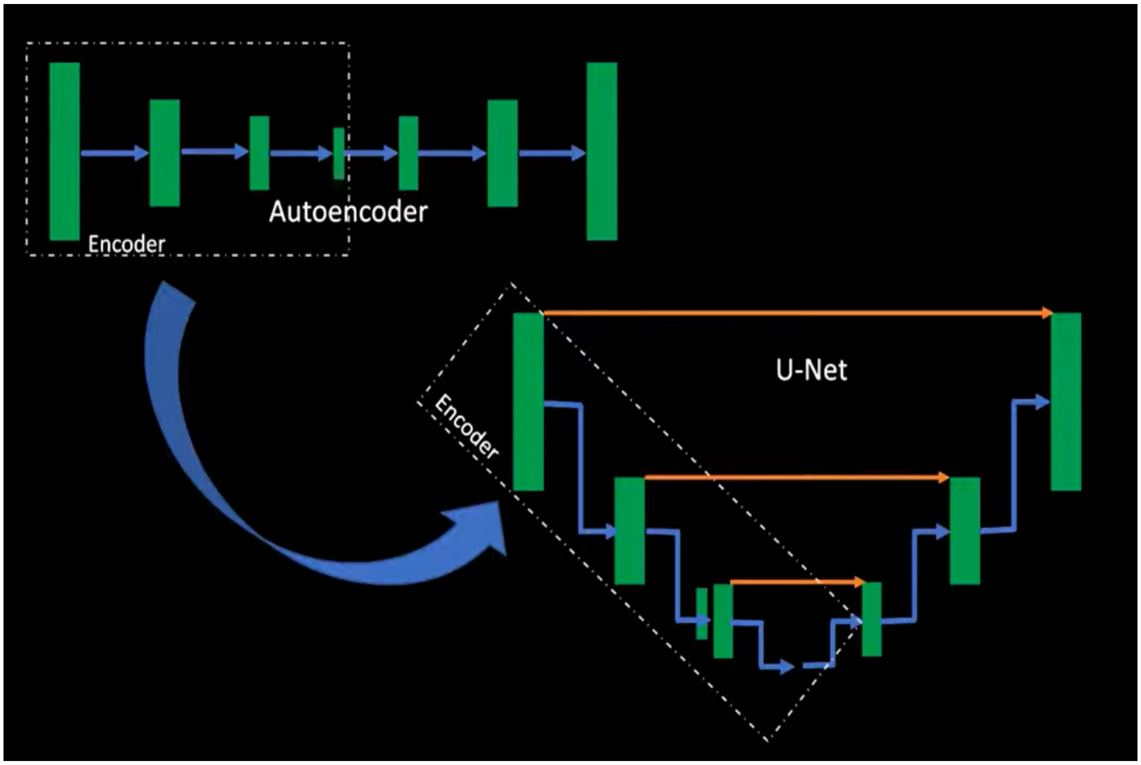
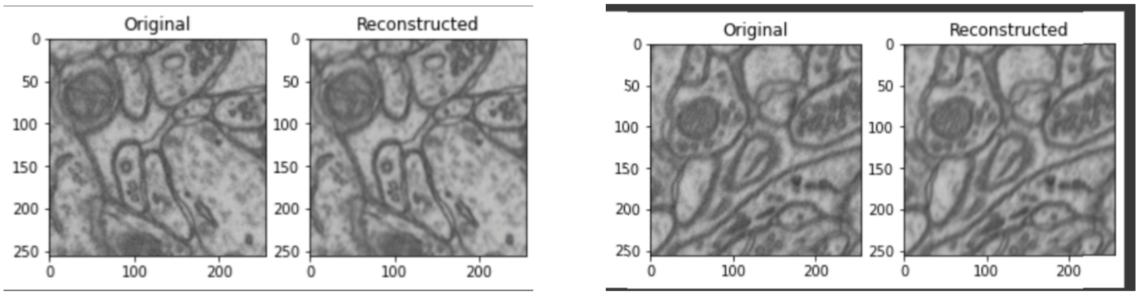


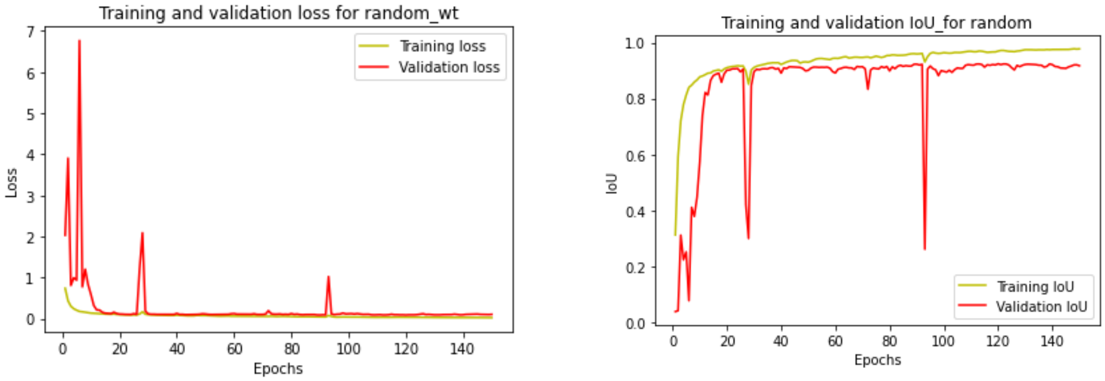
Figure 43: Encoder of autoencoder to use in Unet as a part of encoder



(a) Image Reconstruction using Autoencoder

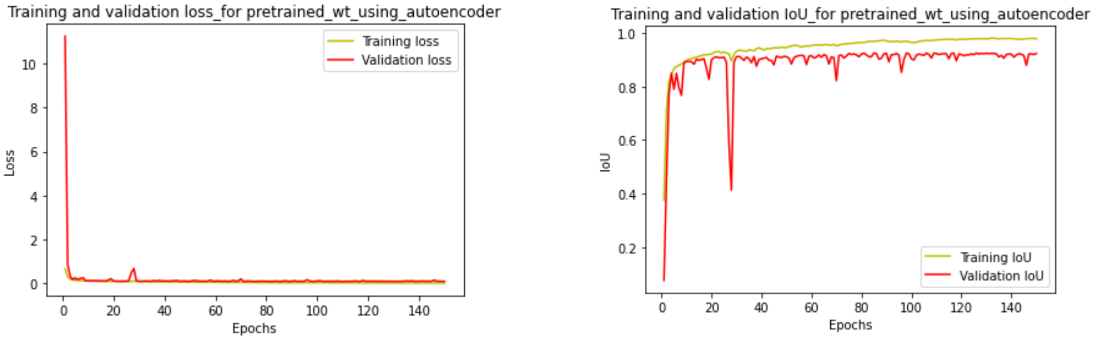
(b) Image Reconstruction using Autoencoder

Figure 44: Autoencoder images Reconstrution



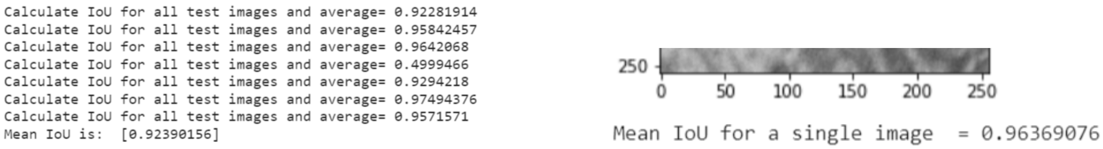
(a) Training and Validation Loss (b) Training and Validation f1-score

Figure 45: Loss and Iou Using Random Weights



(a) Training and Validation Loss (b) Training and Validation f1-score

Figure 46: Iou-score and F1-score Using Pretrained weights Weights



```
Calculate IoU for all test images and average= 0.92281914
Calculate IoU for all test images and average= 0.95842457
Calculate IoU for all test images and average= 0.9642068
Calculate IoU for all test images and average= 0.4999466
Calculate IoU for all test images and average= 0.9294218
Calculate IoU for all test images and average= 0.97494376
Calculate IoU for all test images and average= 0.9571571
Mean IoU is:  [0.92390156]
```

```
Mean IoU for a single image  = 0.96369076
```

(a) Mean Iou (b) Iou-score Calculated on single Image

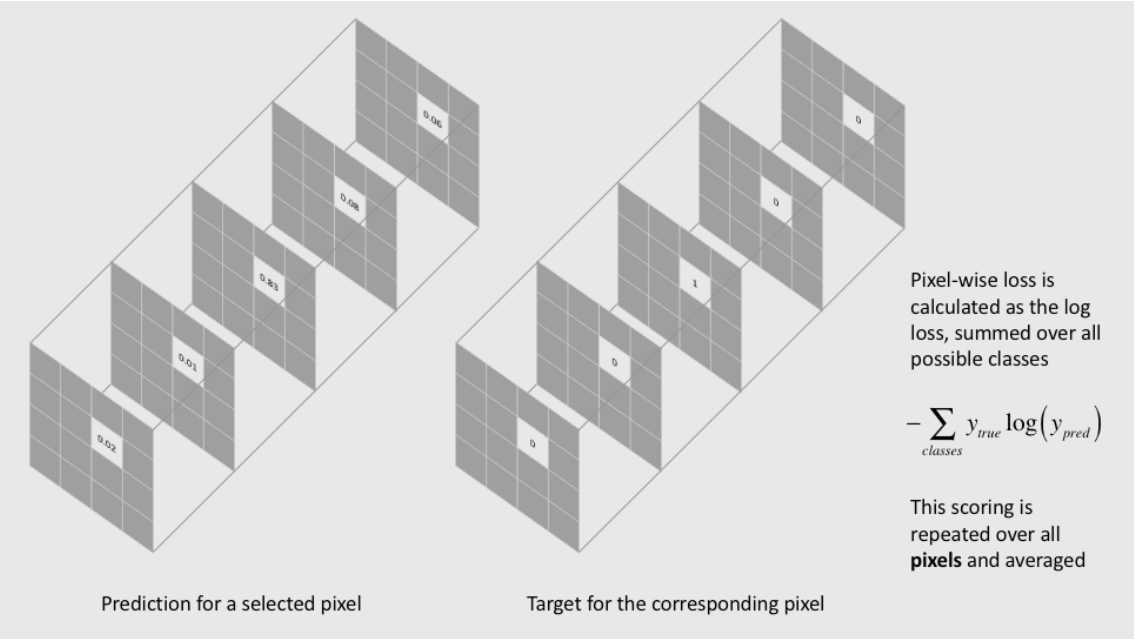
Figure 47: Iou-score on all images and a single image

8 Loss Function

8.1 Pixel-wise cross entropy loss

This loss compares the class predictions (a depth-wise pixel vector) against our one-hot encoded target vector for each particular pixel.

We are effectively asserting equal learning to each pixel in the image because the cross entropy loss analyses the class predictions for each pixel vector individually before averaging over all pixels. This can be a concern because training might be dominated by the class that is shown in the image the most frequently. For each output channel, Long et al. (FCN article) consider weighting this loss to account for the dataset's class imbalance. [**1**]


0.06
0.08
0.83
0.01
0.02
0
0
1
0
0
Prediction for a selected pixel
Target for the corresponding pixel

Pixel-wise loss is calculated as the log loss, summed over all possible classes

$$-\sum_{classes} y_{true} \log\left(y_{pred}\right)$$

This scoring is repeated over all **pixels** and averaged

(a)

Figure 48: (a) pixel-wise cross entropy loss

8.2 Dice coefficient

In essence, it is a measurement of sample overlap. The Dice coefficient, which spans from 0 to 1, represents perfect and total overlap. Initially designed for binary data, the Dice coefficient can be calculated as:

$$Dice = \frac{2\,|A \cap B|}{|A| + |B|}$$

where $|A \cap B|$ represents the common elements between sets A and B, and $|A|$ represents the number of elements in set A (and likewise for set B).

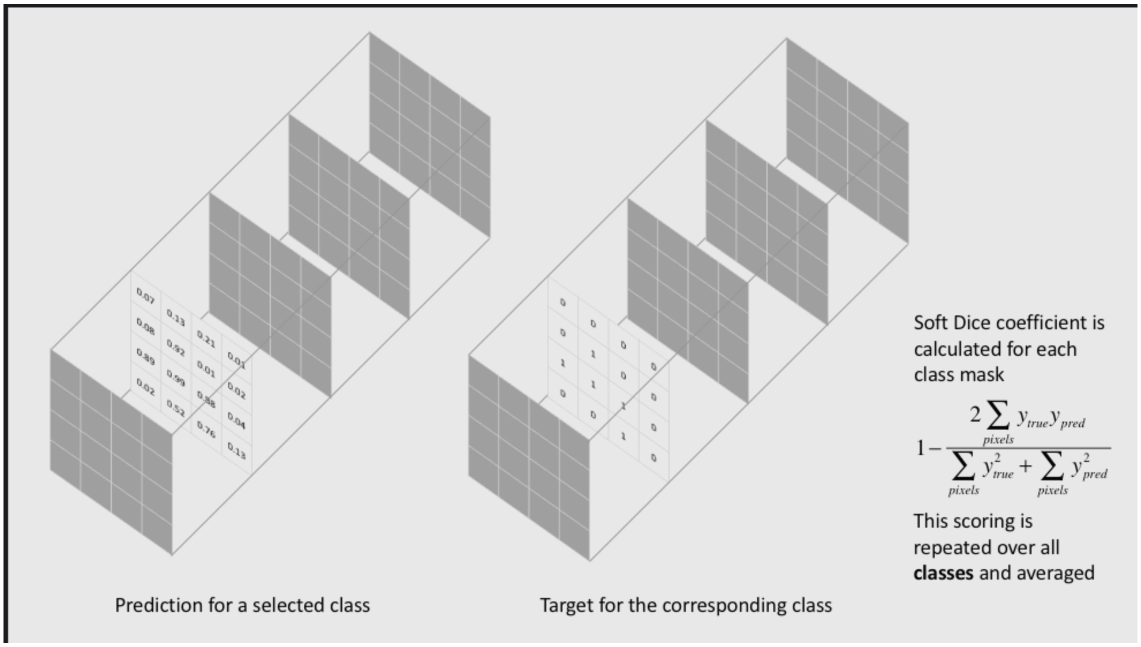
For the case of evaluating a Dice coefficient on predicted segmentation masks, we can approximate $|A \cap B|$ as the element-wise multiplication between the prediction and target mask, and then sum the resulting matrix.

Since our target mask is binary, any pixels from our prediction that are not "activated" in the target mask are effectively zeroed out. We effectively penalise low-confidence predictions for the remaining pixels; a greater value for this expression, which is in the numerator, results in a better Dice coefficient.

In order to quantify $|A|$ and $|B|$, Some scholars perform this calculation using the simple sum, while others favour using the squared sum. I'll let you test them both and decide which performs better because I lack the practical knowledge to know which empirically outperforms the other over a wide range of tasks.

In case you were wondering, the Dice coefficient is calculated with a 2 in the numerator since our denominator "twice counts" the elements that are shared by the two sets. In order to formulate a loss function which can be minimized, we'll simply use $1 - Dice$. Because we directly use the predicted probabilities rather than thresholding and turning them into a binary mask, this loss function is also referred to as the "soft Dice loss."

The numerator of the neural network output reflects the shared activations between our prediction and the target mask, whilst the denominator reflects the total number of activations across each mask individually. So that the soft Dice loss does not have trouble picking up new information from classes that have less spatial representation in an image, this has the effect of normalising our loss in accordance with the size of the target mask.


0.07 0.13 0.21 0.01
0.08 0.92 0.01 0.02
0.89 0.99 0.88 0.04
0.02 0.52 0.76 0.13
0 0 0 0
0 1 0 0
1 1 1 0
0 0 1 0
Soft Dice coefficient is calculated for each class mask
$1 - \frac{2 \sum_{pixels} y_{true} y_{pred}}{\sum_{pixels} y_{true}^2 + \sum_{pixels} y_{pred}^2}$
This scoring is repeated over all **classes** and averaged
Prediction for a selected class
Target for the corresponding class

(a)

Figure 49: (a) Dice Coefficient

8.3 Focal Loss

Another way to think of focal loss (FL) is as a variant of binary cross-entropy. It allows the model to put greater emphasis on learning challenging examples by de-weighting the contribution of simple examples. It functions well in situations with extreme class imbalance, as seen in Fig. 1. Let's examine the design of this focus loss. To understand how Focal loss is generated from cross-entropy, we will first examine binary cross entropy loss.

$$CE = \begin{cases} -log(p) & if\ y = 1 \\ -log(1-p) & otherwise \end{cases}$$

To make convenient notation, Focal Loss defines the estimated probability of class as:

$$P_t = \begin{cases} p & if\ y = 1 \\ 1-p & otherwise \end{cases}$$

Therefore, Now Cross-Entropy can be written as

$$CE(p, y) = CE(p_t) = -log(p_t)$$

Focal Loss proposes to down-weight easy examples and focus training on hard negatives using a modulating factor$((1-p)t)^{\gamma}$ as shown below

$$FL(p_{\gamma}) = -\alpha_{\gamma}(1-p_t)^{\gamma} lop(p_t)$$

Here, $\gamma > 0$ and when $\gamma = 1$ Focal Loss works like Cross Entropy loss function. Similarly, generally range from [0,1], It can be set by inverse class frequency or treated as a hyperparameter.

9 Evaluation Metrics

9.1 Pixel Accuracy

The concept of pixel precision is likely the simplest to grasp. It is the proportion of correctly categorised pixels in your image. Even though it is simple to understand, this metric is far from ideal.

It could be challenging to detect the problem with this measure at first look. Here is a situation that will reveal this measure for what it actually is: Let's imagine that you put your segmentation model to use on the image to the left. The ground truth, or annotation, is shown in the picture to the right (what the model is supposed to segment). Our model is here attempting to identify ships in a satellite image..

Simply said, one class made up 95% of the original picture. Therefore, if the model assigns every pixel to that class, 95% of the pixels are correctly classified while the remaining 5% are not. Because of this, even though your accuracy is a stunning 9%, your model is giving a prediction that is absolutely meaningless. This is intended to demonstrate that higher segmentation skills aren't always implied by high pixel precision.

Class imbalance is the term for this problem. When our classes are severely out of balance, one or more classes dominate the picture while other classes make up a very minor fraction of it. Unfortunately, class imbalance can't be disregarded because it can be seen in many real-world data sets.

9.2 Intersection-Over-Union (IoU, Jaccard Index)

The Intersection-Over-Union (IoU), also known as the Jaccard Index, is one of the most commonly used metrics in semantic segmentation… and for good reason. The IoU is a very straightforward metric that's extremely effective.

Look at the illustration to the left before continuing to the next sentence. According to the graphic on the left, the IoU is the area of union between the predicted segmentation and the ground truth divided by the area of overlap between the predicted segmentation and the ground truth. This metric ranges from 0–1 (0–100%) with 0 signifying no overlap and 1 signifying perfectly overlapping segmentation.

The mean IoU of the picture is determined for binary (two classes) or multi-class segmentation by averaging the IoU of each class. (The code does it a little bit differently.)

Now let's use the identical example as section 1 to try to understand why this statistic is superior to pixel precision. Let's assume for the sake of simplicity that the ships (coloured boxes) all belong to the same class.

9.3 Dice Coefficient (F1 Score)

The IoU and the Dice coefficient are quite similar. They are positively associated, thus if one claims that model A is superior to model B at picture segmentation, the other will also claim the same. They both have a range of 0 to 1, like the IoU, with 1 denoting the greatest resemblance between expected and truth.

IoU and the Dice Coefficient are the two metrics for semantic segmentation that are most frequently utilised.

10 Final Observations:

In our All experiments we are concerned about the time to converge the unet So we here realise that with the help of transfer learning and pre trained auto encoder on our data set can greatly reduces the no. of epochs to converge

11 References

[1] https://www.epfl.ch/labs/cvlab/data/data-em/

[**2a**] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J.Kress, I. C. Lopez, and J. V. Soares. Leafsnap: A computer vision system for automatic plant species identification. In Computer Vision–ECCV 2012, pages 502–516 Springer, 2012.

[**3a**] J.-X. Du, X.-F. Wang, and G.-J. Zhang. Leaf shape based plant species recognition. Applied mathematics and computation, 185(2):883–893, 2007. 2

[**4**] P. J. Komi, M. R. Jackson, and R. M. Parkin. Plant classification combining colour and spectral cameras for weed control

[**5**] K. Thorp and L. Tian. A review on remote sensing of weeds in agriculture. Precision Agriculture, 5(5):477–508, 2004. 2

[**6**] S. Christensen, H. T. Søgaard, P. Kudsk, M. Nørremark,I. Lund, E. Nadimi, and R. Jørgensen. Site-specific weed control technologies. Weed

[**7**] J. Hemming and T. Rath. Computer-vision-based weed identification under field conditions using controlled lighting. Journal of Agricultural Engineering Research, 78(3):233–243, 2001. 2, 3

[**8**] B. Astrand and A.-J. Baerveldt. An agricultural mobile robot with vision-based perception for mechanical weed control. Autonomous robots, 13(1):21–35, 2002. 2

[**9**] J. C. Neto, G. E. Meyer, and D. D. Jones. Individual leaf extractions from young canopy images using gustafson–kessel clustering and a genetic algorithm. Computers and Electronics in Agriculture, 51(1):66–85, 2006. 2

[**relu**] Deep Learning using Rectified Linear Units (ReLU) Atention References

[**10**] Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and vqa. arXiv preprint arXiv:1707.07998 (2017) [**11**] Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)

[**12**] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł.,Polosukhin, I.: Attention is all you need. In: NIPS.pp. 6000–6010 (2017)

[**12a**] Jetley, S., Lord, N.A., Lee, N., Torr, P.: Learn to pay attention. In: International Conference on Learning Representations (2018),https://openreview.net/forum?id=HyzbhfWRW

[**13**] Velickovi ˇ c, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention ´networks. arXiv preprint arXiv:1710.10903(2017)

[**14**] Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., Tang, X.: Residual attention network for image classification. In: IEEE CVPR. pp. 3156–3164 (2017)

[**15**] Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)

[**16**] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NIPS. pp. 6000–6010(2017)

[**17**] Mnih, V., Heess, N., Graves, A., et al.: Recurrent models of visual attention. In: Advances in neural information processing systems. pp. 2204–2212 (2014)

[**18**] Ypsilantis, P.P., Montana, G.: Learning what to look in chest X-rays with a recurrent visual attention model. arXiv preprint arXiv:1701.06452 (2017)

[**21**] Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. arXiv:1709.01507 (2017)

[**22**] Lee, C.Y., Xie, S., Gallagher, P., Zhang, Z., Tu, Z.: Deeply-supervised nets. In: Artificial Intelligence and Statistics. pp. 562–570 (2015)

[**25**] Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. arXiv preprint arXiv:1711.07971 (2017)