Experimental evaluation of ship detection using U-Net with various backbone networks

Vishalakshi Kamath

Master's Thesis

Master of Engineering - Big Data Analytics

2022

MASTER'S THESIS	
Arcada	
Degree Programme:	Big Data Analytics
Identification number:	8732
Author:	Vishalakshi Kamath
Title:	Experimental evaluation of ship detection using U-Net with various backbone networks.
Supervisor (Arcada):	Amin Majd, D.Sc. (Tech.) and PhD
Commissioned by:	

Abstract:
The growth in global trade has led to the growth in global ship traffic. Maritime security, safety and tracking has become more critical. Organizations and governments globally are developing applications to improve maritime surveillance. Among the widely used solutions, satellites are used to remotely scan and record images of waterways and shores. These image data feeds are processed using deep learning algorithms like Convolutional Neural Network (CNN) to detect ships with great precision. The challenges in this process are the image quality, size of ships, other varied noise within data and the computational requirements of handling and managing big data. The thesis work covers evaluating the U-Net model with popular transfer learning networks for semantic image segmentation on Airbus ship detection data. The backbones are pretrained on ImageNet dataset. Deep learning models were developed using Keras API with TensorFlow backend. The U-Net model with the selected backbones are compared with each other using standard metrics.

Keywords:	Ship detection, Transfer Learning, Deep Learning, CNN, Satellite images, Semantic Image segmentation, U-Net architecture, Maritime surveillance, ResNet-50, InceptionV3, MobileNetV2, EfficientNetB2 and DenseNet-121.
Number of pages:	50
Language:	English
Date of acceptance:	

CONTENTS

Figures

Tables

ABBREVATIONS AND TERMS

ANN	Artificial Neural Network
CNN	Convolutional Neural Network
ReLU	Rectified Linear Unit
RoI	Region of Interest
R-CNN	Region Based Convolutional Neural Network
Mask R-CNN	Mask Region Based Convolutional Neural Network
Cascade R-CNN	Cascade Region Based Convolutional Neural Network
Faster R-CNN	Faster Region Based Convolutional Neural Network
ResNet	Residual Network
GPU	Graphical Processing Unit
IoU	Intersection of Union
TP	True Positive
FP	False Positive
FN	False Negative

FOREWARD

I would like to express my gratitude to the Big Data Analytics team at Arcada University of Applied Sciences including my study colleagues and the teachers to name a few Dr. Magnus Westerlund, Dr. Leonardo Espinosa Leal and Andrej Scherbakov-Parland without whom the journey was not possible. Special thanks to my supervisor Dr. Amin Majd for his constructive feedback and insightful supervision throughout the project.

Last but not the least, I would like to thank my family especially my spouse, my son and friends for supporting me both spiritually and mentally throughout this journey. Without them, I would never have accomplished this.

I feel incredibly grateful for all of you.

Helsinki

May 2022

Vishalakshi Kamath

1 INTRODUCTION

Today's global economy and trade possibilities allow manufacturers around the world to manufacture products, ship and market the products worldwide. The global trade growth has increased shipping traffic and the need for ship detection and tracking. This study covers some of the Deep Learning, transfer learning methods for ship detection using satellite images with different encoders and loss functions to train segmentation models.

1.1 Background

As the market and consumption becomes global, the products needs to be shipped to consumers using the most economic logistics. Maritime transport logistics is critical for global trade and countries, regions are getting into large agreements for improving multilateral trade possibilities.

As global trade and shipping traffic grows, the infractions on international waterways like ship accidents, piracy and illegal cargo trade are on rise. Maritime security, safety, and tracking requirements have increased. Several companies and government bodies have developed systems to support safe navigation and security defining the border control zones and regulations. These ship detection applications are important also for defense to monitor country territorial shores for any possible enemy ship intrusions.

With the advancement in satellite imaging, it is now possible for satellites to continuously provide image data feeds. Due to remote satellite imaging and sensing constraints, these images require complex processing to enable detection of ships with good precision, avoiding the noise in the image data. Deep learning and advancements in data handling and computing has been successful in improving the object detection models.

1.2 Motivation and Outcome

This thesis is based on the Airbus Ship Detection Challenge using the satellite image data available in Kaggle data sources (Airbus ship detection challenge, 2018). The semantic

segmentation models have been successfully used for various types of images and have been used for ship detection. The study compares the U-Net model with available transfer learning methods like ResNet50, InceptionV3, MobileNetV2, DenseNet-121, VGG-16 and EfficientNetB2. U-Net with these pretrained models were further trained and validated to get standard metrics.

1.3 Thesis Structure

The thesis has the following main chapters:

- Introduction – Provides the business context, motivation and outcomes for research
- Literature review and Theoretical Background
- Materials and Methods
- Results and conclusions
- Future improvements

2 BACKGROUND THEORY

With the increase in global trade and shipping traffic, the navigation, tracking, border control applications are in demand. Various organizations and government bodies are working on developing solutions to ensure safe navigation and security to maritime activities limiting pollution and avoiding illegal cargo movements.

2.1 Ship Detection

Ships are being widely used in global trade, defense, fishing, travel, etc., so ship detection has become critical for various purposes like security, border control, traffic, defense and fishing. Using object detection methods ships can be precisely monitored using images from the space or satellite-based imaging. (Kanjir, Greidanus and Oštir, 2018)

Earlier costal radars and shore based Automatic Identification Systems (AIS) were extensively used, but these limited the monitoring near the shore. Satellite imaging-based ship detection allows more wider monitoring of the ocean. (Oligschläger, 2020)

Ship detection is a special case of object detection, where the background has the characteristics of the sea surface. Ship detection becomes challenging due to various real-time changes of the background - sea surface, lighting, pollution, ship size and orientation of the ships. The ship object variation and the background affect the image clarity and makes ship detection more difficult than other general object detection. (Li et al., 2020)

2.2 Challenges in Ship Detection

Computer vision is an emerging application, as it can be widely used in many military, social and industrial aspects, such as intelligent video surveillance in traffic management, security monitoring, content-based image retrieval and person re-identification. This process of object detection involves locating and classifying objects in images or video frames often received using remote sensing. The challenges in object detection are due to variations in quantities, positions, forms and sizes of objects, and to identify these from the varying background.

Ship detection is just a specific case of the object detection problem. Unlike other general object detection, the remote sensing-based image quality will be degraded by daylight, mists, clouds, ocean waves and other backgrounds like islands, harbor or coastline. The remote sensing satellite images for ships are mainly affected by three factors – weather conditions, imaging conditions and target property. Weather conditions include the sea state & the cloud conditions. Sea states like quite or stormy large waves will affect the imaging. Clouds conditions may vary no clouds, thin or thick cloud. These affect the grey scale distribution in images and the target-ship features. Imaging conditions mainly the lighting like morning, dusk, noon or night. Also, the weather seasons affect the brightness, contrast due to change in target exposure which impacts the image quality. Target property are the ship objects to be monitored. Ships are moving targets on sea and have variable wakes based on the ship size, speed, sea states and weather conditions. These

wakes are connected to ship and can be incorrectly detected as extension of target ship. Ships are in varying sizes and course of movements can be varying like rotational, so the detection algorithms need to handle scale and rotation. In ship detection the low contrast can cause algorithms to miss the objects and the broken clouds, sea waves, ship wakes, islands and other background objects can cause false ship detection. (Yao, Jiang and Zhang, 2016)

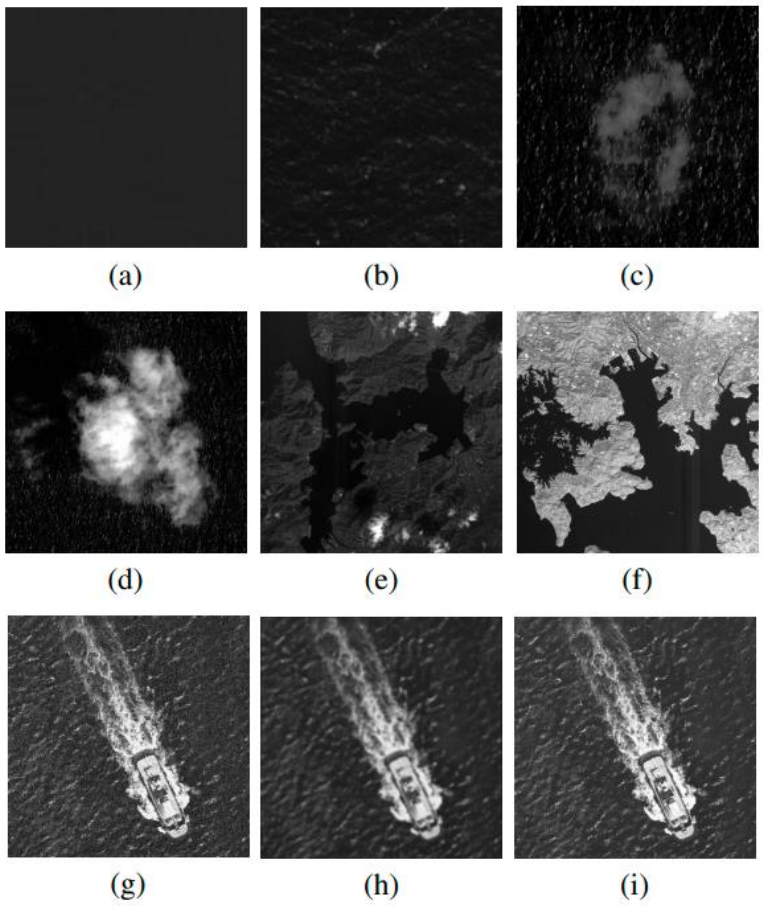


Figure 1. Optical remote sensing images under different conditions.

(a)-(c) – Sea states, (d)-(f) – Cloud state, (g)-(i) – target ship image quality with wakes.

(Yao, Jiang and Zhang, 2016)

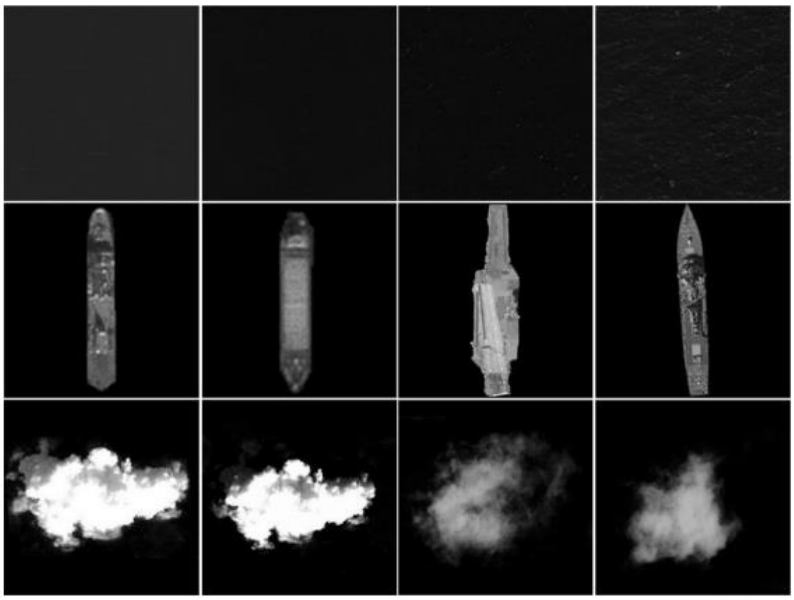


Figure 2. Sea surface, ship target and cloud samples

(Yao, Jiang and Zhang, 2016)

3 LITERATURE REVIEW

3.1 Deep Learning

Deep learning is machine learning using neural network with three or more layers. The neural networks try to learn from large amount of data. A single layer neural network can make approximate predictions and additional layers then improve the accuracy of predictions.

Deep learning is used in many artificial learning (AI) based automation applications where human involvement can be avoided by self-learning automated systems. Deep learning has many applications across industries like digital assistants, automated self-driving vehicles, etc. (IBM Cloud Education, 2020)

3.2 Artificial Neural Networks (ANN)

The artificial neural network (ANN) approach of machine learning is based on simulating the human brain. The Big data analysis needs machine learning to learn large properties within data & their inter-relations to derive an accurate machine model. ANN can model complex non-linear relations within data and are fast, fault tolerant, scalable networks. (Zou, Han and So, 2008)

Simulating the human neuron, a perceptron is an artificial neuron taking various inputs and generating a weighted summation output. The weights are learned during the model training process. (Shanmugamani, 2018)

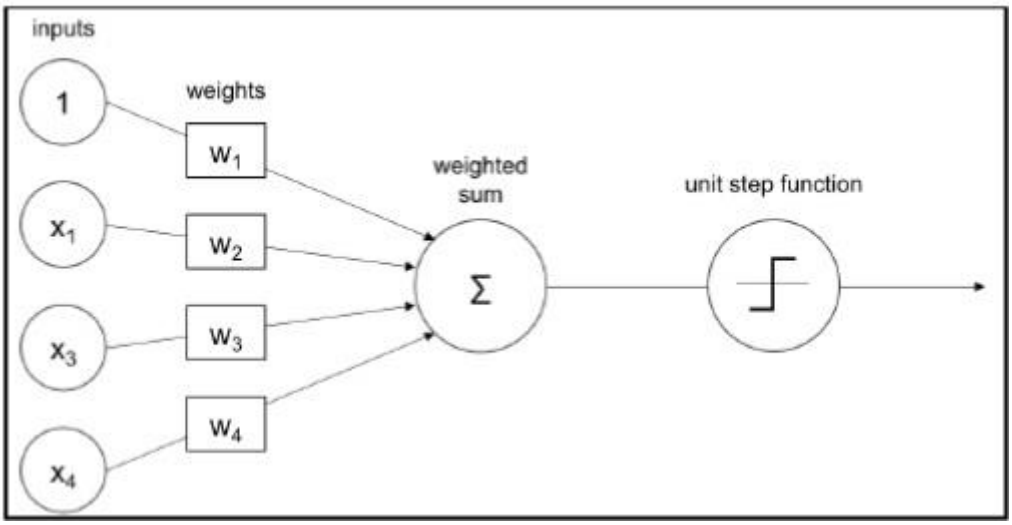


Figure 3. Single layer Perceptron / Artificial neuron

(Shanmugamani, 2018)

A collection of perceptrons and activation functions together form the artificial neural network (ANN). The connected perceptrons form hidden layers or units that map the input to output on non-linear basis.

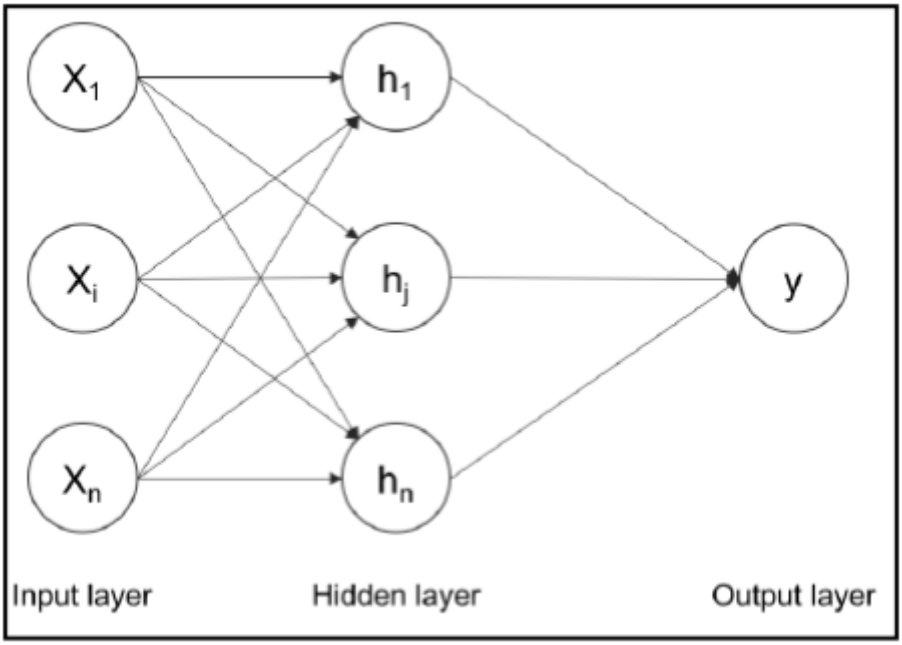


Figure 4. Multi-Layer Perceptron

(Shanmugamani, 2018)

3.3 Convolutional Neural Network (CNN)

The Convolutional Neural Network (CNN) are multilayer perceptron in regularized form. Here each neuron in one layer is linked to all neurons of next layer. With the regularized form CNNs are less complex and have many applications like face or action detection, text classification, natural language processing, image classification and computer vision functions. CNN uses feed forward architecture. (Alghazo et al., 2021)

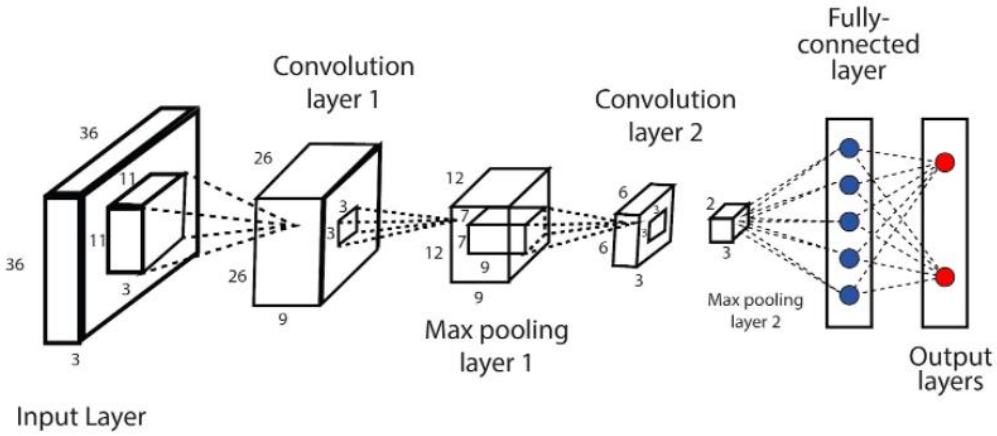


Figure 5. Architecture of Convolutional Neural Network (CNN)

(Balodi, 2021)

3.3.1 Forward Propagation

In forward propagation neural networks is based on mapping function $y = f(x; \theta)$, the model learns the value of θ as a best function approximation. (Schulz and Behnke, 2012) In forward propagation model, there are no feedback connections. In cases where the forward propagation neural networks are extended to include feedback connections then these are called recurrent neural networks. (Goodfellow, Bengio and Courville, 2016)

3.3.2 Backward Propagation

In backward propagation the information flows from Output layer to Input layer preferably used for corrections or feedback to adjust weights to minimize the loss function. (Seth, 2021)

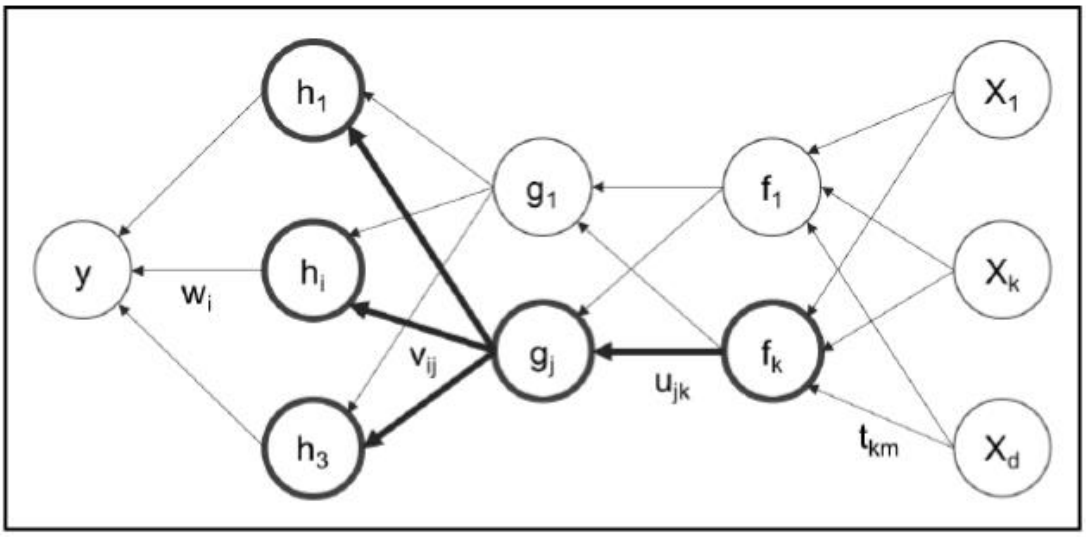


Figure 6. Backward propagation

(Shanmugamani, 2018)

3.4 Semantic Segmentation

Semantic Segmentation approach of computer vision identifies the contents of image for each pixel. It differs from image recognition where entire image is identified.

Some of its primary applications are in autonomous vehicles, human-computer interaction, robotics, and photo editing/creativity tools, where surrounding scene understanding plays a vital role. (Image Segmentation Guide, 2021)

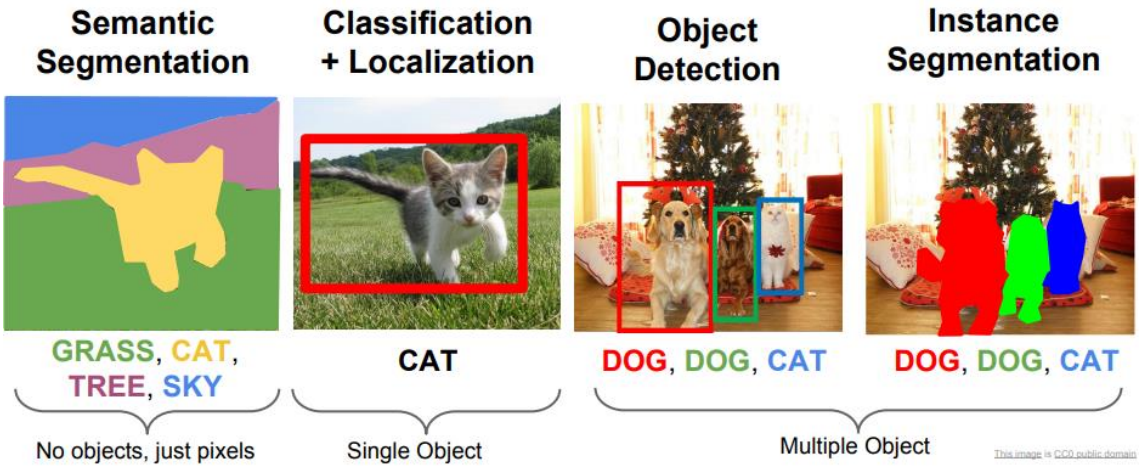


Figure 7. Computer vision tasks

(Li, Johnson & Yeung, 2017)

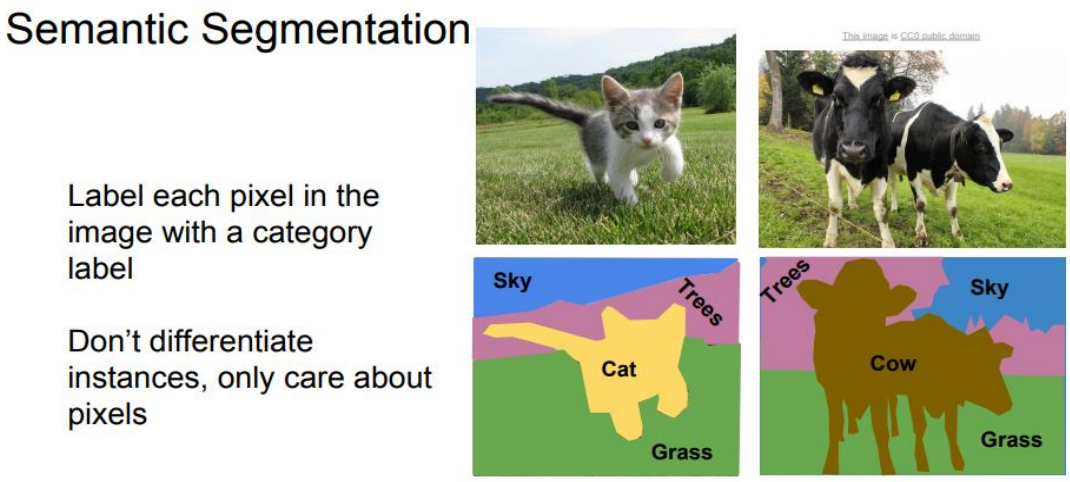


Figure 8. Semantic segmentation example

(Li, Johnson & Yeung, 2017)

Semantic segmentation: Fully convolutional

This has encoder-decoder networks. The encoder module that learns the feature maps and collects semantic information. The decoder module then collects the spatial information. (Chen et al., 2018)

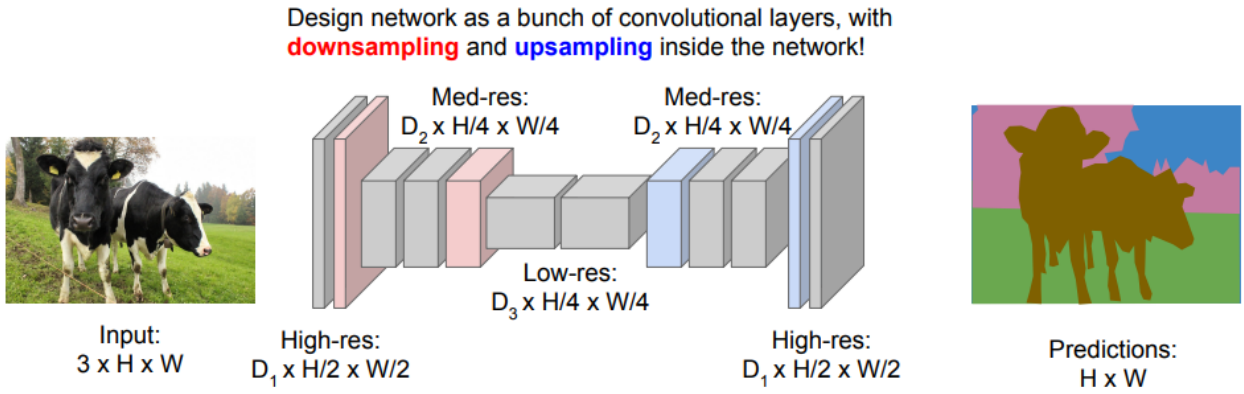


Figure 9. Semantic segmentation: fully convolutional

(Li, Johnson & Yeung, 2017)

3.5 Transfer Learning

Transfer Learning method provides feature representation with pre-trained models such that training effort and time is minimized. Pre-trained models are trained using large datasets like ImageNet, these models can then be tuned for custom applications using case specific image data. This approach reduces the training time and generalization error. Pre-trained models perform poorly when the features learned through the initial training data do not match the problem dataset. Popular pre-trained architectures include VGG family, ResNet, Inception, MobileNet, EfficientNet and DenseNet. These models are trained on ImageNet dataset and can be implemented with frameworks like TensorFlow, Keras, and Pytorch.

ImageNet Dataset – is a large dataset of annotated photographs which are used for computer vision modelling. This dataset has about 14 million images from more than 21000 groups or classes of images and have more than 1 million images that are annotated with bounding box. The ImageNet large scale vision recognition challenge for deep learning required models to be developed to classify these images out into 1000 object categories. (Lendave, 2021)

3.5.1 Commonly used Transfer Learning Models

Inception: Compared to VGGNet, Inception Networks (GoogLeNet/Inception v1) are more computationally efficient, generating fewer number of parameters requiring less memory and other resources. This was proposed in the paper Rethinking the Inception Architecture for Computer Vision, published in 2015, co-authored by Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, and Jonathon Shlens. (Szegedy et al., 2014)

Inception networks need to be optimized to allow easier adaptation, these include factorized convolutions, regularization, dimension reduction, and parallelized computations. (Kurama, 2020)

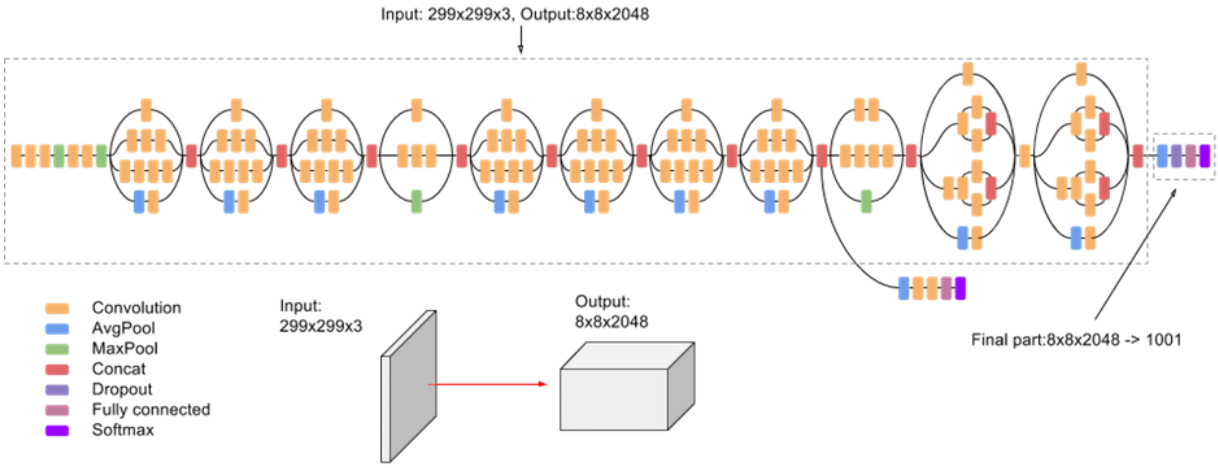


Figure 10. Inception model architecture

(Kurama, 2020)

EfficientNet: This CNN family has been built by Google, it provides better accuracy and efficiency by reducing the number of parameters in comparison to other models. EfficientNet-B0 model is a baseline architecture trained on the ImageNet dataset. EfficientNet is built to provide an effective compound scaling method to achieve maximum accuracy gains. The figure below provides the visualization of scaling possibilities. (Jordan, 2018)

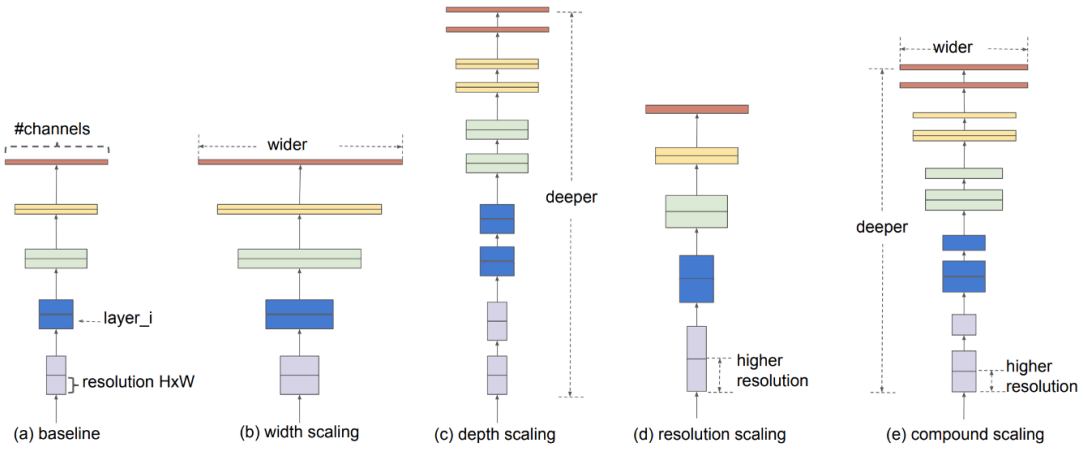


Figure 11. Model Scaling

(Tan and Le, 2019)

(a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) Is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

VGG Family: VGG stands for Visual Geometry Group; it is a standard deep Convolutional Neural Network (CNN) architecture with multiple layers. The "deep" refers to the number of layers with VGG-16 or VGG-19 consisting of 16 and 19 convolutional layers**.**

The VGG architecture is the basis of ground-breaking object recognition models. Developed as a deep neural network, the VGGNet also surpasses baselines on many tasks and datasets beyond ImageNet. Moreover, it is now still one of the most popular image recognition architectures. (Leonardblier, 2016)

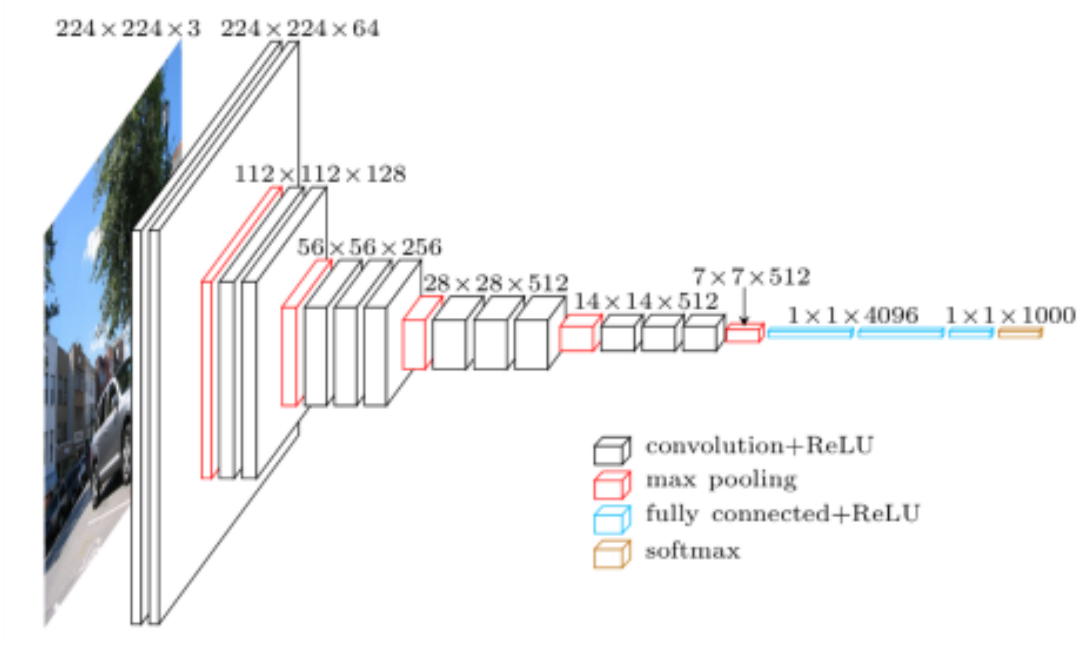


Figure 12. VGG Neural Network Architecture

(Leonardblier, 2016)

ResNet: This model differed from the previous sequential networks, is based on microarchitecture modules or building blocks used to create a new network. This Residual Network model was introduced by Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun in paper "Deep Residual Learning for Image Recognition". In deep learning networks as the number of layers increase there is problem of Vanishing/Exploding gradient, causing the gradient to become 0 or too large, causing increase in error with increase in

layers. The residual network architecture uses skip connection technique. These skip connections or shortcuts improves performance by regularization.

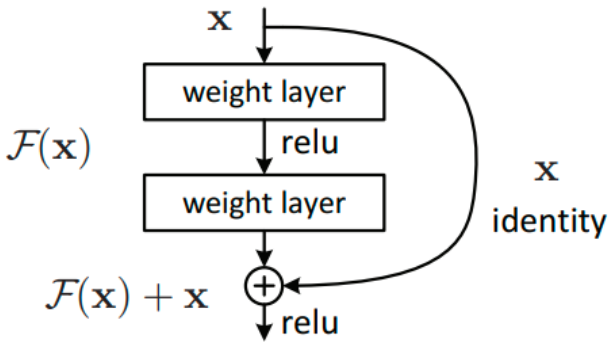


Figure 13. Skip (shortcut) connections in ResNet

(Tomar, 2021a)

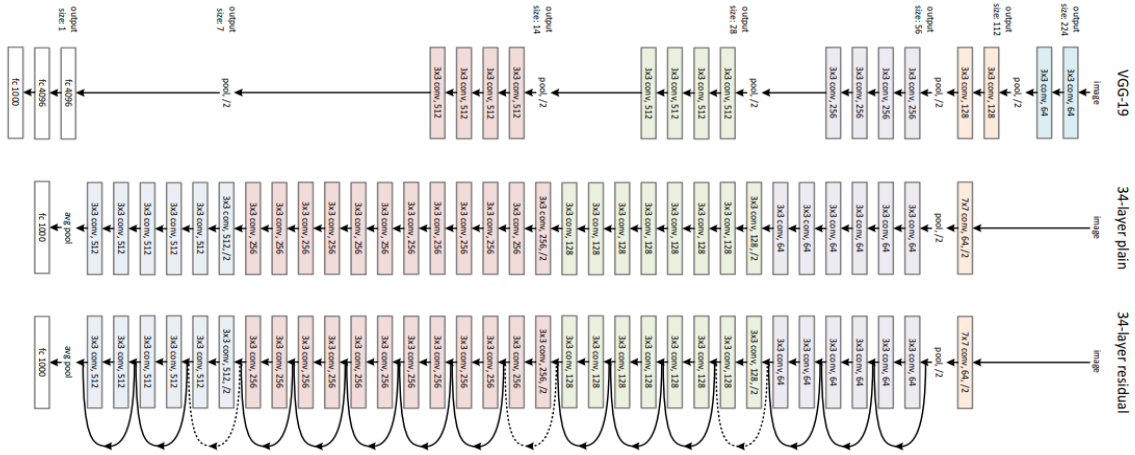


Figure 14. ResNet-34 architecture

(Tomar, 2021a)

The ResNet-34 architecture uses a 34-layer plain architecture inspired by VGG-19 with shortcut connections added. The shortcut connections convert architecture into residual network. (Tomar, 2021a)

MobileNet: This is TensorFlow's first mobile computer vision model designed for mobile applications.

MobileNet is based on depthwise separable convolutions which reduces the number of parameters in comparison with regular convolutions networks, and so is a lightweight deep neural network. MobileNet is open sourced by Google giving a starting point for training the classifiers. (Pujara, 2020)

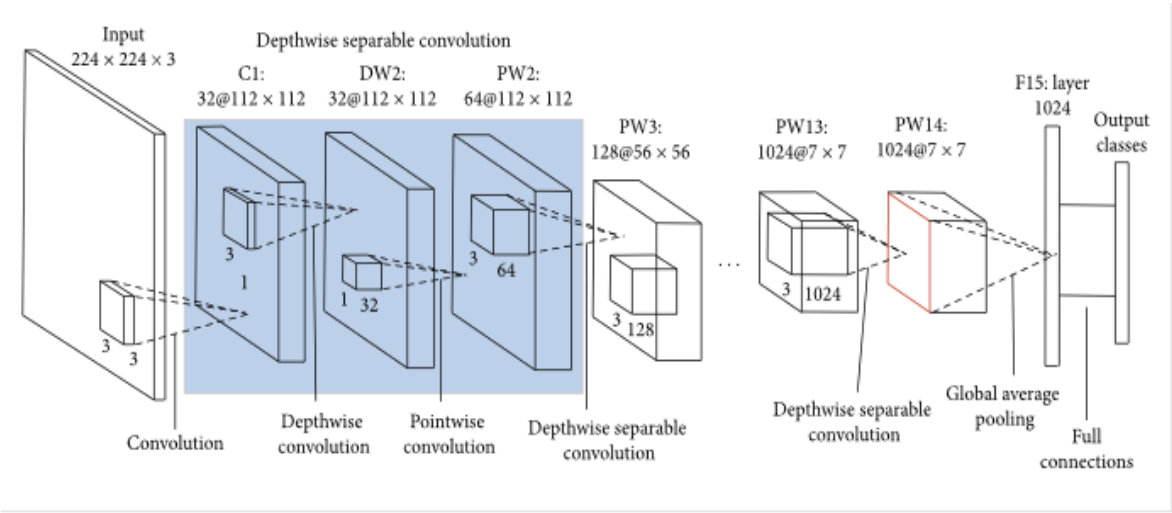
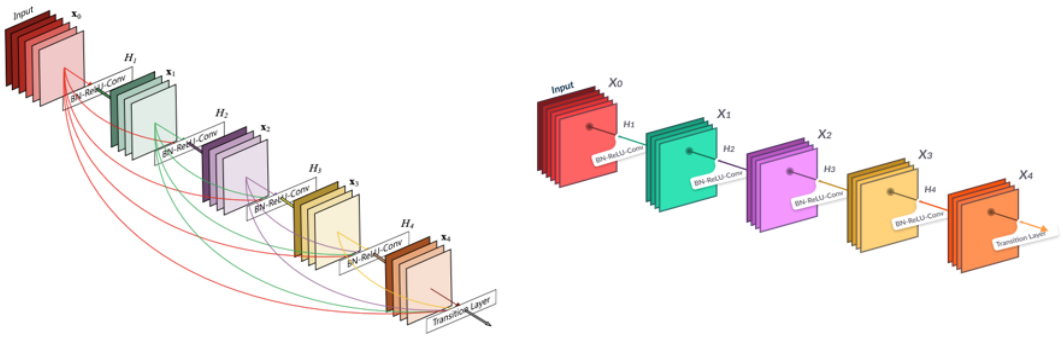


Figure 15. MobileNet architecture

(Pujara, 2020)

DenseNet: This is densely connected-convolutional networks and like a ResNet. They differ as ResNet uses an additive method taking a previous output as an input for a future layer, whereas the DenseNet takes all previous output as an input for a future layer. DenseNet is developed to improve accuracy over vanishing gradient in high-level neural networks. (Huang *et al.*, 2016)



DenseNet Structure

ResNet Structure

$$a^{[l]} = g\left([a^{[0]}, a^{[1]}, a^{[2]}, \dots\dots\dots\dots, a^{[l-1]}\right)$$

$$a^{[l]} = g\left(z^{[l+1]} + a^{[l]}\right)$$

Figure 16. DenseNet Architecture VS ResNet Architecture

(Singhal, 2020)

3.6 Satellite imagery

Satellites with different sensors collect data for remote sensing applications. There are active & passive sensors based on the sensing approach. Passive sensors collect the earths reflected radiation received from Sun. The Active sensors can emit radiation to earth and collect reflection.

Satellites and the EM Spectrum - Satellites carry sensors that can capture radiation covering broad EM spectrum. These could include human visible light, infrared light, ultraviolet light, or even microwaves. The satellite sensing data which are from non-visible spectrum are then mapped to visible colors to generate visible images. (How does satellite imaging work?, 2021)

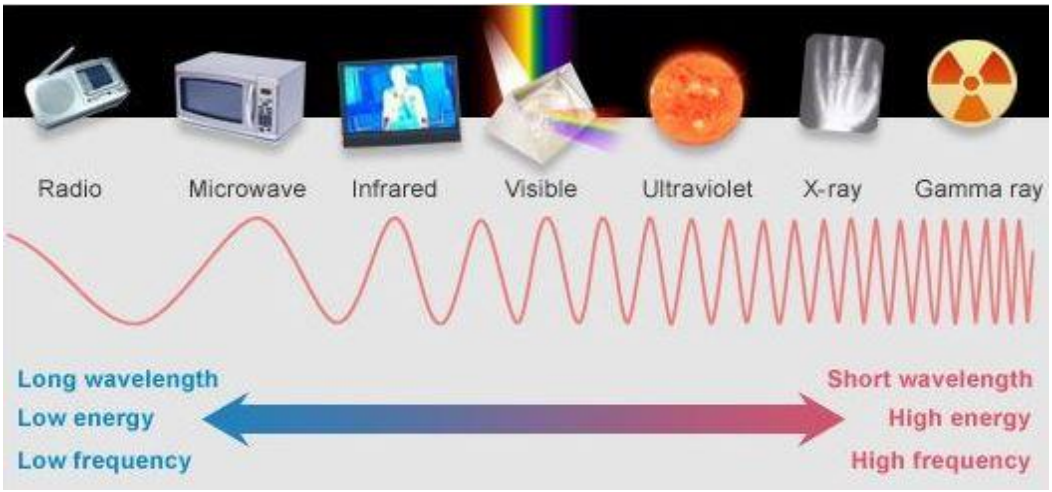


Figure 17. The EM Spectrum

(How does satellite imaging work?, 2021)

3.7 Related Work

The use of satellite images to detect ships has been studied as both segmentation as well as object detection problem statement. The CNN-based SAR ship detection can be divided into region-based methods and regression-based methods. a) In region-based detection process there are two stages. The first stage generates a set of sparse candidate proposals, then classifying these proposals into different categories. Few variants for this process are Faster RCNN, RFCN, Cascade RCNN. b) In regression-based approach, feature pyramids and multiscale prediction structure are used to improve accuracy and speed. It predicts the location and categories of the objects. (Han et al., 2020)

Han *et al.* in their paper "Multi-Size Convolution and Learning Deep Network for SAR Ship Detection from Scratch" provide a SAR ship detector from scratch. The experiments show the design to have good accuracy, speed and robustness. (Han et al., 2020)

Yuan Yao *et al*. in his paper "High-resolution optical satellite image simulation of ship target in large sea scenes" lists the usual image vulnerabilities and simulates the image sensing issues due to different sea states, cloud conditions, target types and imaging conditions which can support the evaluation of ship detection algorithms. (Yao, Jiang and Zhang, 2016)

There has been lot of research concerning the segmentation of ships from image and detection of ships from image.

4 METHODS AND MATERIAL

The programming language used for preprocessing the data and developing/training model was Python. Commonly used packages such as Numpy, Pandas were used. Also, deep learning models were developed using Keras API with TensorFlow backend. All the experiments were conducted on the CSC's puhti computing GPU partition with NVIDIA Volta V100.

4.1 Dataset

A public dataset provided on the Airbus Ship Detection Challenge website (Airbus ship detection challenge, 2018) is used in this thesis. The data set contains more than 100K images of size 768*768 with a total size exceeding 30Gb. Many images do not contain ships, and those that do may contain multiple ships. Ships within and across images may differ in size (sometimes significantly) and be located in open sea, at docks, marinas, etc. There is actually quite imbalance in the sense that only ≈ 1/4 of the data images have ships. Along with the images is a CSV file that lists all the images' ids and their corresponding pixels coordinates. These coordinates represent segmentation boxes of ships. Not having pixel coordinates means the image doesn't have any ships.

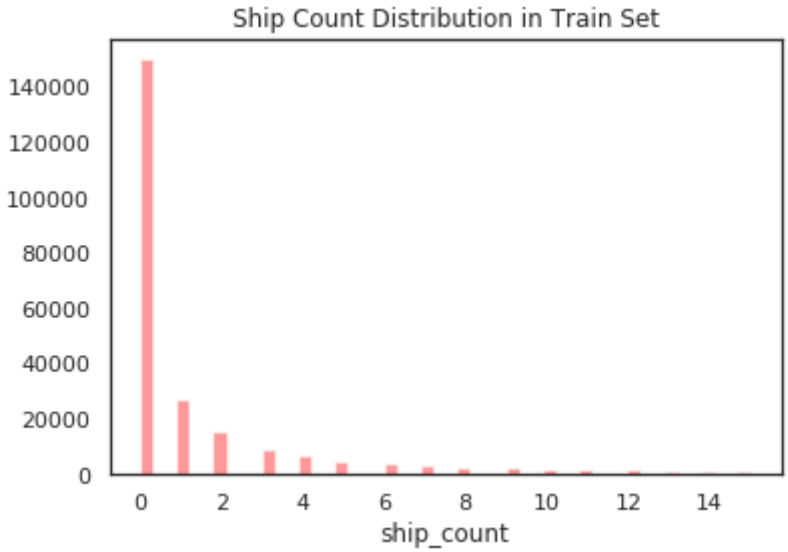


Figure 18. Ship counts in data

The challenge of detecting the ships in the images can be thought as a classification problem for pixels, where, for each image, we need to classify 768 × 768 pixels in one of two classes: ship and no-ships. Below graph shows the imbalance of classes considering the pixel level granularity, only 0.001% of the pixels are ships, while 99.9% of the pixels are no-ships. And dropping all the images with no ships in them the class imbalance is reduced, but it's still very high: this is, 0.5% of the pixels are ships while 99.5% are no-ships.

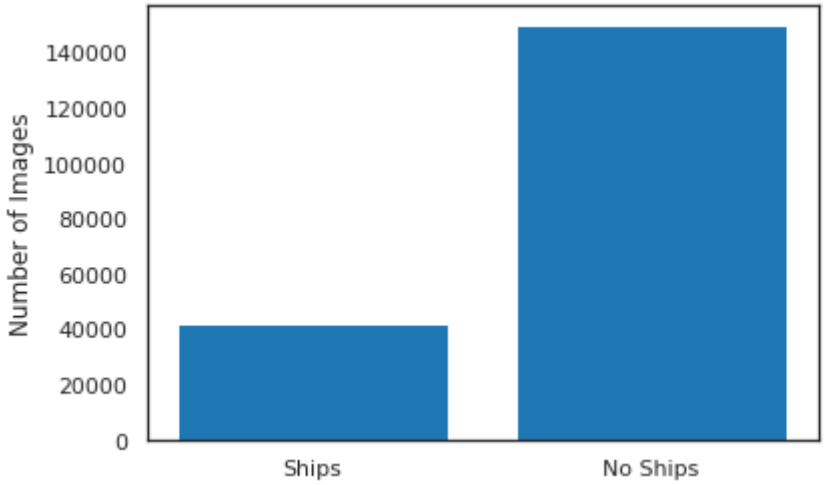


Figure 19. Image count with valid ships

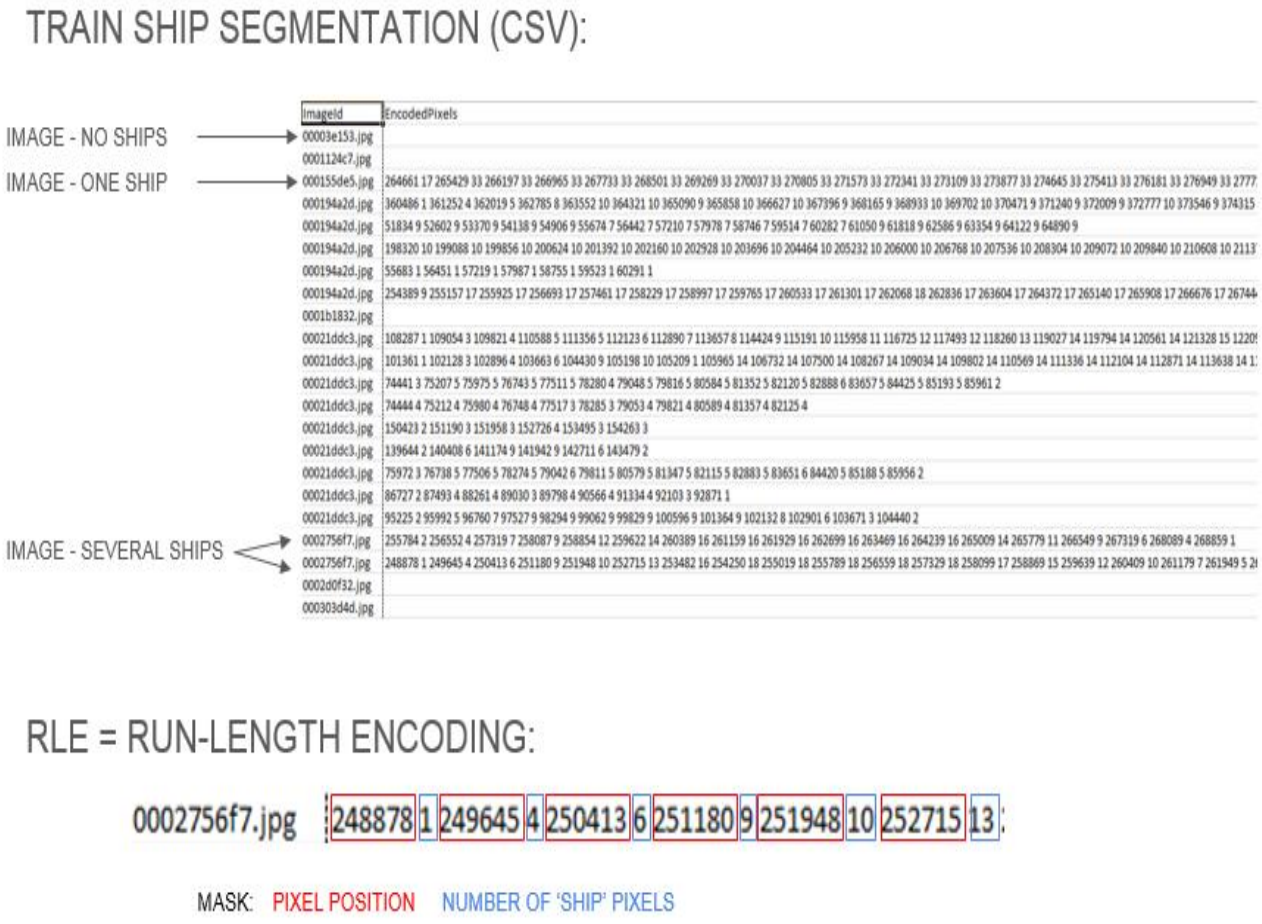


Figure 20. Image data in RLE

Figure 20 shows the masks in run length encoding format. It shows a list of start pixels and how many pixels after each of those starts is included in the mask. There are no masks for images without ships. In the csv file there are many images with more than one mask, which means there are more than one ship in that image.

In the below Figure we can see the image and the masks for ships in the image and overlay.

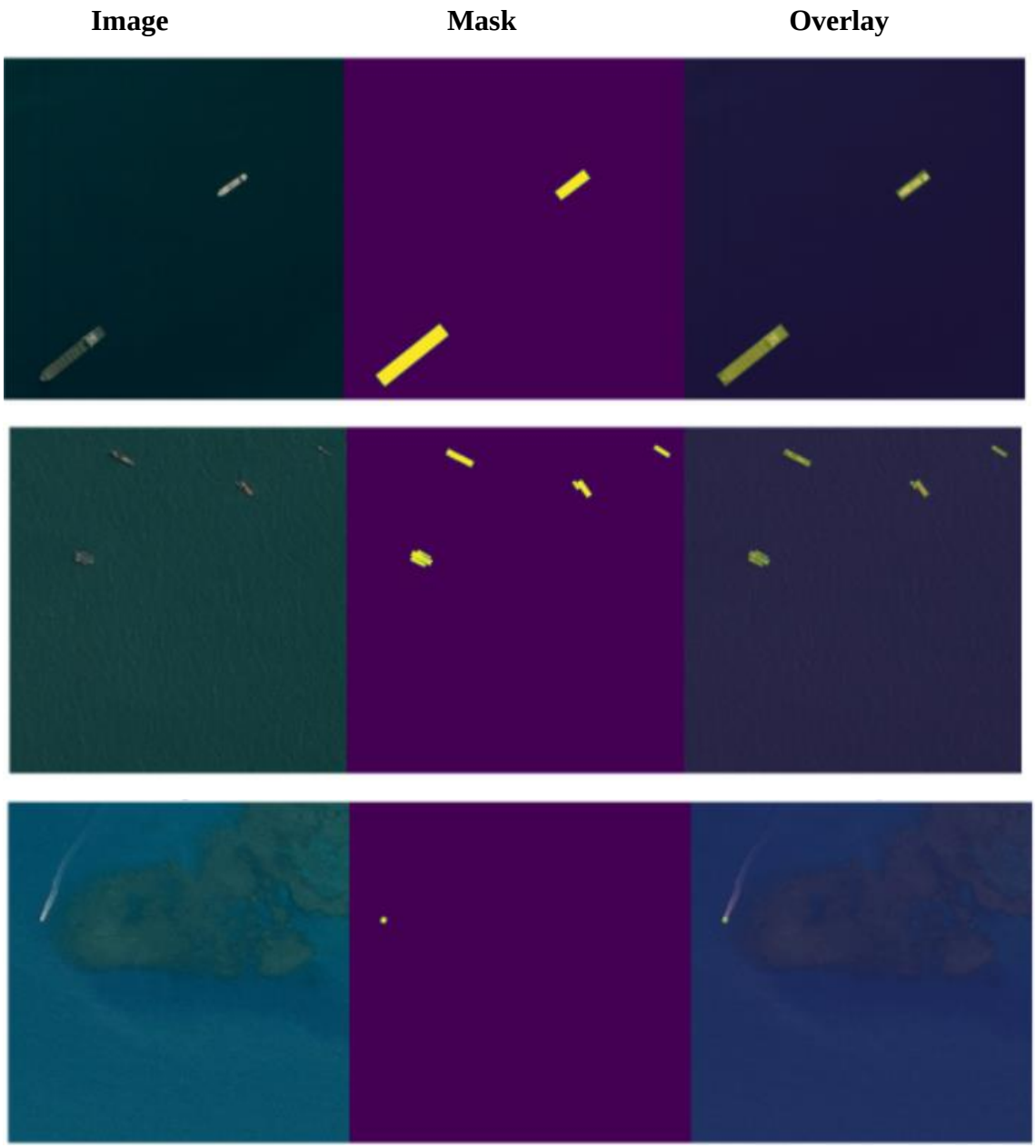


Figure 21. Images with masks

4.2 U-Net: Convolutional Networks for Image Segmentation

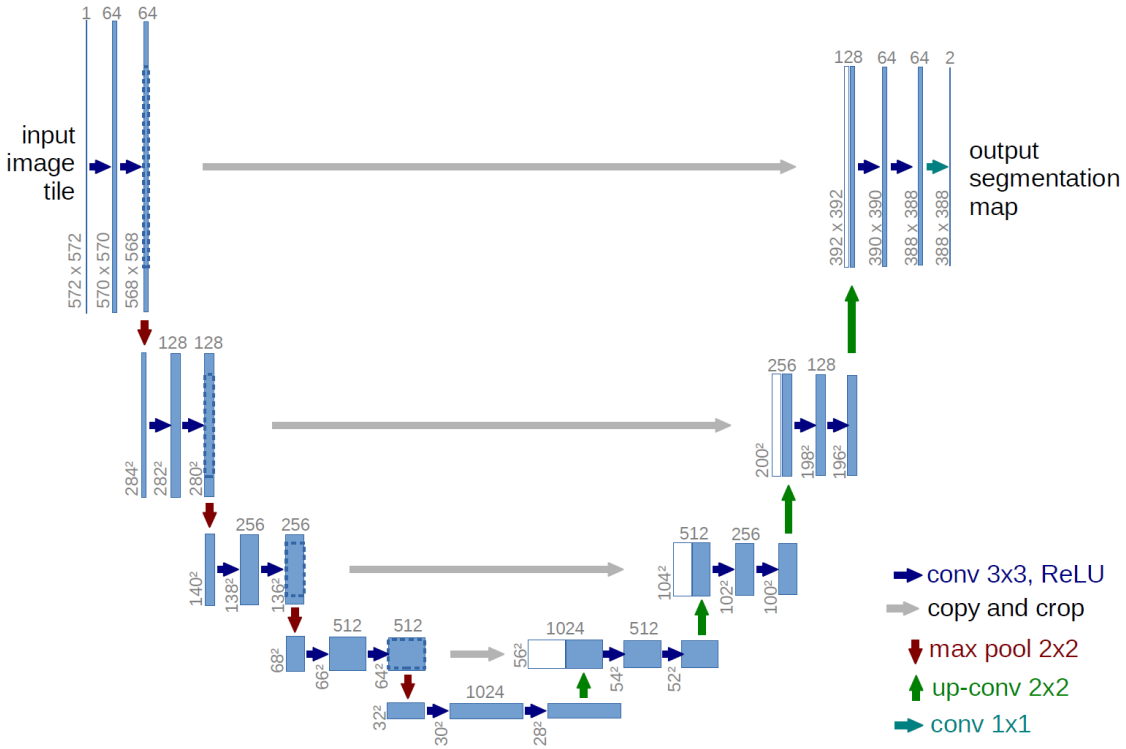


Figure 22. U-Net architecture - Example for 32x32 pixels in the lowest resolution

(Ronneberger, Fischer and Brox, 2015)

4.2.1 U-Net Network Architecture

U-net is an encoder-decoder network architecture which is U-shaped. Four encoder blocks and four decoder blocks are connected via a bridge. The encoder network half the spatial dimensions and double the encoder block filters. And the decoder network doubles the spatial dimensions and half the number of feature channels. (Tomar, 2021b)

4.2.2 Encoder Network

The encoder network extracts the features and learns the abstract representation of the input image through the encoder blocks. Each encoder block consists of 3*3 convolutions which are followed by ReLU (Rectified Linear Unit) activation function. The output of the activation function ReLU acts as a skip connection for the decoder block. (Tomar, 2021b)

Next, follows a 2x2 max pooling, where the spatial dimensions (height and width) of the feature maps are reduced by half. This reduces the computational cost by decreasing the number of trainable parameters. (Tomar, 2021b)

4.2.3 Skip Connections

The skip connections help the decoder to generate better semantic features. Skip connections helps to learn better representation by helping in better flow of gradient while back propagation. (Tomar, 2021b)

4.2.4 Bridge

The bridge completes the flow of information by connecting encoder and decoder networks. It consists of 3*3 convolutions, which is followed by ReLU activation function. (Tomar, 2021b)

4.2.5 Decoder Network

The decoder network takes the abstract representation and generates the segmentation masks. The decoder block starts with 2*2 transpose convolution. Then it concatenates with the skip connection feature map from the encoder block. the skip connections provide features from earlier layers which get lost sometimes due to the depth of the network. Then two 3*3 convolutions are used, which are followed by activation function ReLU. The last decoder output passes through 1*1 convolution with sigmoid activation function which gives pixel-wise classification of segmentation masks. (Tomar, 2021b)

4.3 Backbone of Network

The backbone defines how these layers are arranged in the encoder network and how decoder network should be built. The backbones used in this thesis are VGG, ResNet, Inception, DenseNet, EfficientNet and MobileNet. These CNNs perform encoding and

down sampling itself and their counter parts are built to perform decoding and up sampling to form a final U-net model. (Alsabhan and Alotaiby, 2022)

4.4 Evaluation Metrics

The task of semantic segmentation is simply to predict the class of each pixel in an image. The prediction output shape matches the input's spatial resolution (width and height) with a channel depth equivalent to the number of possible classes to be predicted. Each channel consists of a binary mask which labels areas where a specific class is present.

4.4.1 Intersection over Union

Intersection over Union (IoU) metric is also known as Jaccard Index and closely related to Dice coefficient which is used as loss function during training.

The IoU metrics is measured by the number of common pixels in target and prediction masks divided by the total number of pixels present in both target and prediction masks. (Jordan, 2018)

$$IoU = \frac{A \cap B}{A \cup B}$$

4.4.2 F1 Score

F1 score, also known as Dice coefficient, is doubled the area of overlap between the prediction and ground truth divided by the total number of pixels in both images. This metric is very similar to IoU, and the results of these metrics are positively correlated. (Jordan, 2018)

$$F1 = \frac{2||A \cap B||}{||A|| + ||B||}$$

4.4.3 Precision

To evaluate the predicted mask, we have to compare each predicted mask with the target mask for the given input image.

- A **true positive** is observed when IoU score of the prediction-target pair exceeds the predefined value.
- A **false positive** indicates a predicted object mask has no corresponding ground truth object mask.
- A **false negative** indicates that there is no predicted object mask for the ground truth object mask. (Jordan, 2018)

Figure 23. Precision prediction possibilities

(Jordan, 2018)

$$Precision = \frac{TP}{TP + FP}$$

4.4.4 Recall

Recall describes the completeness of our positive predictions relative to the ground truth. (Jordan, 2018)

$$Recall = \frac{TP}{TP + FN}$$

5 RESULTS AND CONCLUSIONS

Experiments were performed using U-Net with different backbones using Keras to develop and train the models. (Iakubovskii, 2020) The backbones used in this thesis are ResNet50, DenseNet-121, InceptionV3, VGG-16, MobileNetV2 and EfficientB2. The dataset has 193K images and every image has a resolution of 768x768. All the models were trained using 768*768 resolution images.

Experiments are performed on Keras models. Keras models are trained with early stopping callback being monitored on validation loss. All the encoders used in the experiments are pre-trained models on Imagenet. The Keras models were initially trained using BCE_jaccard loss plus dice loss for 15 epochs using images with ships. Models were trained by freezing the encoder and only training the decoder.

5.1 ResNet-50 as Backbone

ResNet-50 is a convolutional Neural Network that is 50 layers deep. U-Net with ResNet-50 as backbone was trained for 15 epochs for images with ships. Below graph shows the training vs. validation metrics for 15 epochs.

Figure 24. ResNet-50 metrics, 1-15 epoch

5.2 DenseNet-121 as Backbone

U-Net with DenseNet-121 as backbone was trained for 15 epochs and the training vs. validation metrics were plotted as below

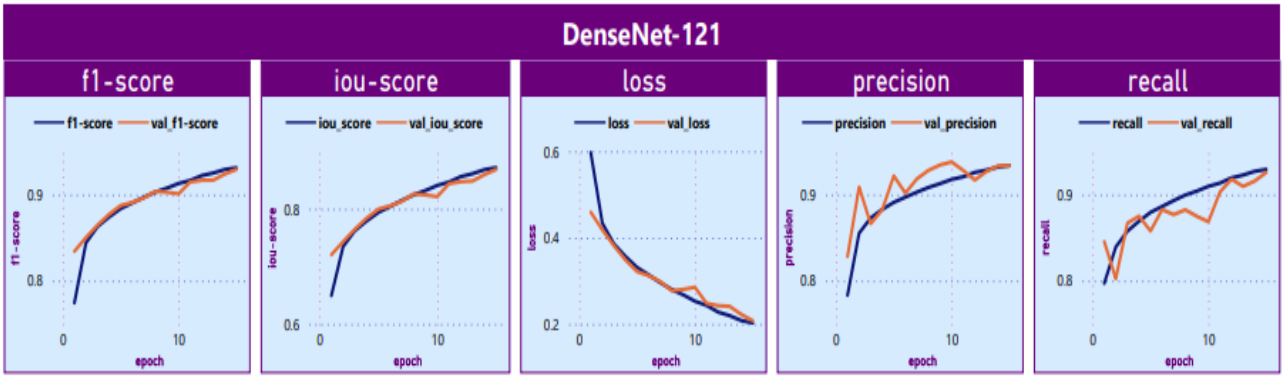


Figure 25. DenseNet-121 metrics, 1-15 epoch

5.3 InceptionV3 as Backbone

U-Net with InceptionV3 as backbone was trained for 15 epochs and the training vs. validation metrics were plotted as below

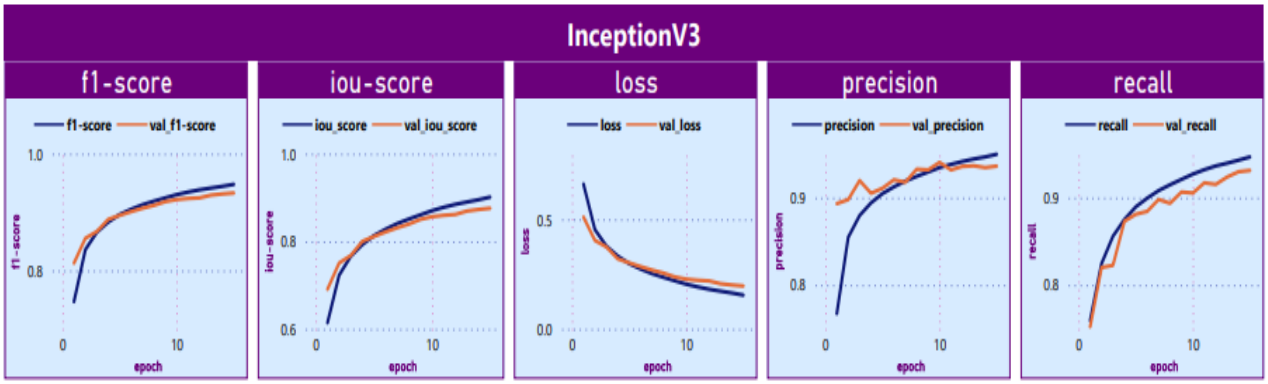


Figure 26. InceptionV3 metrics, 1-15 epoch

5.4 VGG-16 as Backbone

U-Net with VGG-16 as backbone was trained for 15 epochs and the training vs. validation metrics were plotted as below

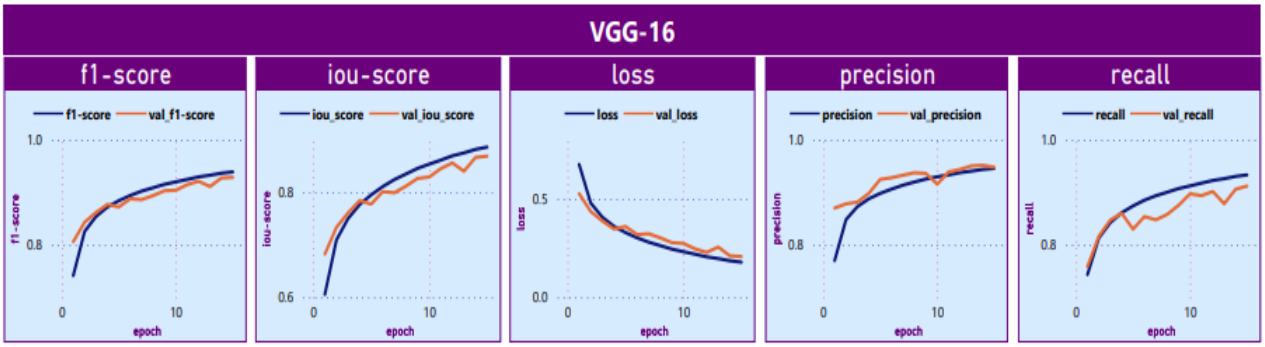


Figure 27. VGG-16 metrics, 1-15 epoch

5.5 MobileNetV2 as Backbone

U-Net with MobileNetV2 as backbone was trained for 15 epochs and the training vs. validation metrics were plotted as below

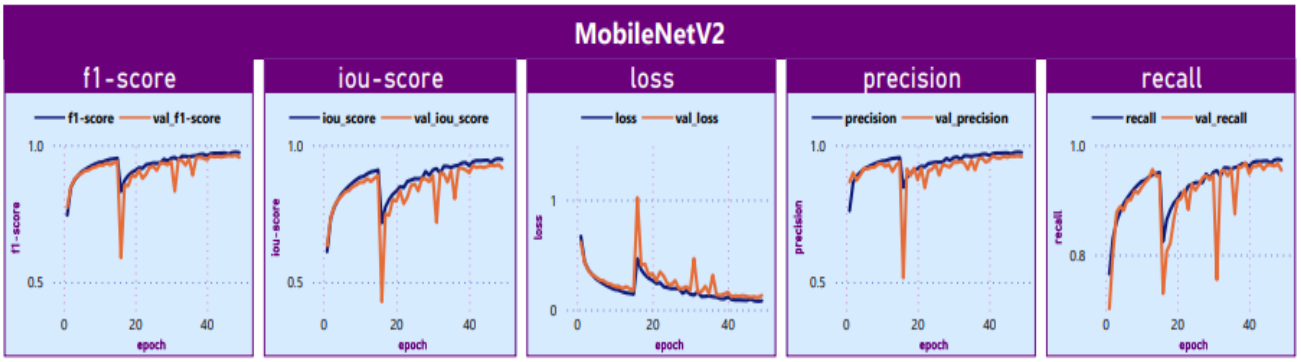


Figure 28. MobileNetV2 metrics, 1-15 epoch

5.6 EfficientNetB2 as Backbone

U-Net with EfficientNetB2 as backbone was trained for 15 epochs and the training vs. validation metrics were plotted as below

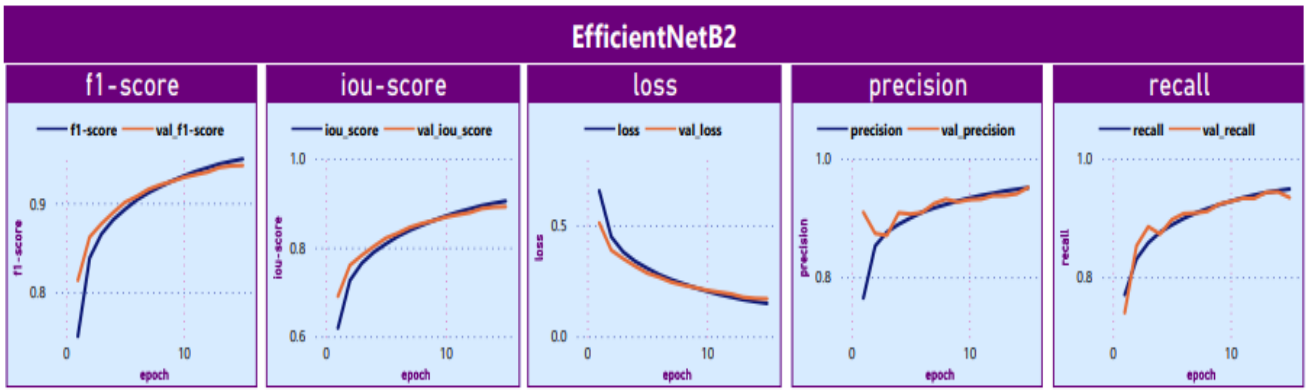


Figure 29. EfficientNetB2 metrics, 1-15 epoch

5.7 Model Test Result

U-Net with different backbones were trained for 15 epochs and the models were tested for the test data. As shown in the table below, F1 score, IoU score, loss, precision and recall were compared. The IoU scores were 0.8676 and 0.8681 and F1 scores were 0.9282 and 0.9280 for DenseNet-121 and VGG-16 respectively, which were the least among other backbones.

Test scores for U-Net+Backbone - for test data

U-Net+Backbone	f1-score	iou_score	loss	precision	recall
Densenet-121	0.9282	0.8676	0.2104	0.9337	0.9242
EfficientNetB2	0.9431	0.8936	0.1681	0.9520	0.9354
InceptionV3	0.9317	0.8739	0.2005	0.9356	0.9294
MobileNetV2	0.9415	0.8902	0.1740	0.9393	0.9443
ResNet-50	0.9393	0.8869	0.1793	0.9469	0.9330
VGG-16	0.9280	0.8681	0.2106	0.9472	0.9117

Table 1. Comparing model metrics.

As shown in the above table MobileNetV2 and EfficientNetB2 outperformed other backbones with f1 scores 0.9415 and 0.9431 and IoU scores 0.8902 and 0.8936 respectively.

Below plot compares the F1 scores, IoU scores, loss, precision and recall for the U-Net with the backbone - DenseNet-121, EfficientNetB2, InceptionV3, MobileNetV2, ResNet-50 and VGG-16.

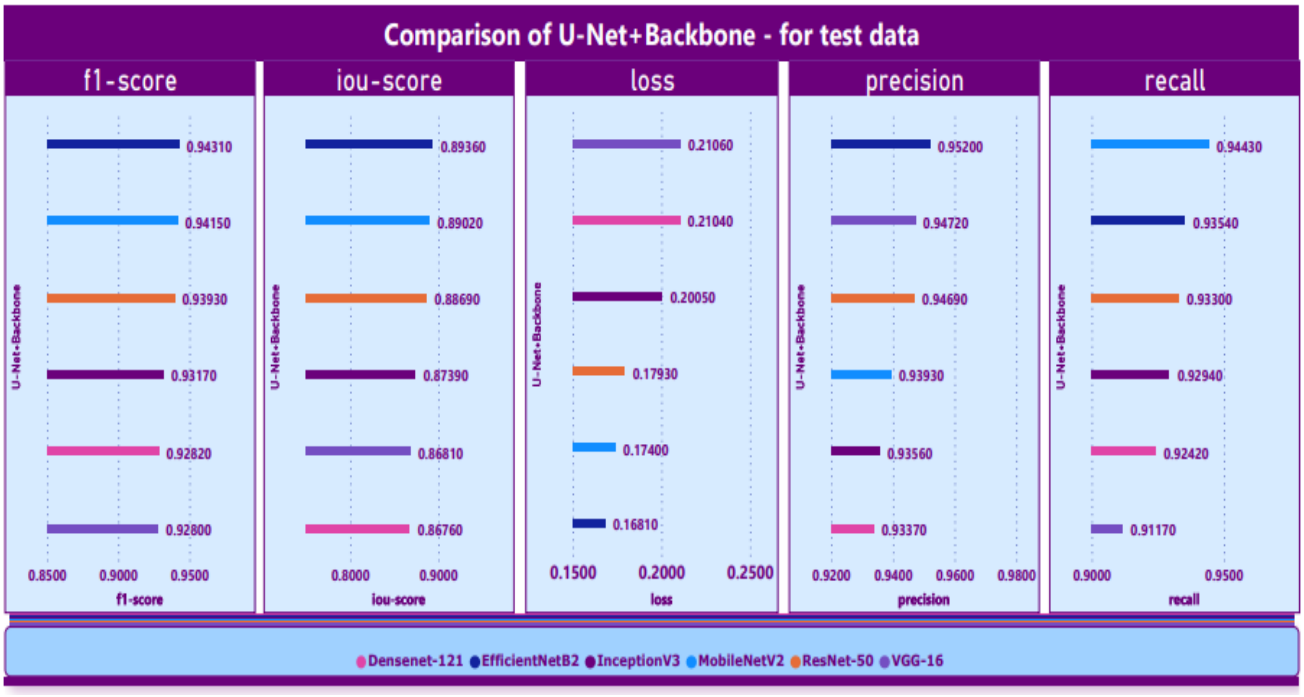


Figure 30. Comparison Metrics, at epoch 15

As seen in *Table 1* U-Net with backbone MobileNetV2 and EfficientNetB2 performed well among other backbones when trained for 15 epochs by freezing the encoder and training only the decoder for images with ships. Further MobileNetV2 and EfficientNetB2 were trained for another 34 epochs (total 49 epochs) without freezing the encoder.

5.7.1 U-Net+MobileNetV2

After training for 49 epochs the metrics is plotted as shown in the below graph. Since we trained for the full model with encoder and decoder we can see in the below graphs how f1 score and IoU score goes down from 16th epoch and gradually increases.

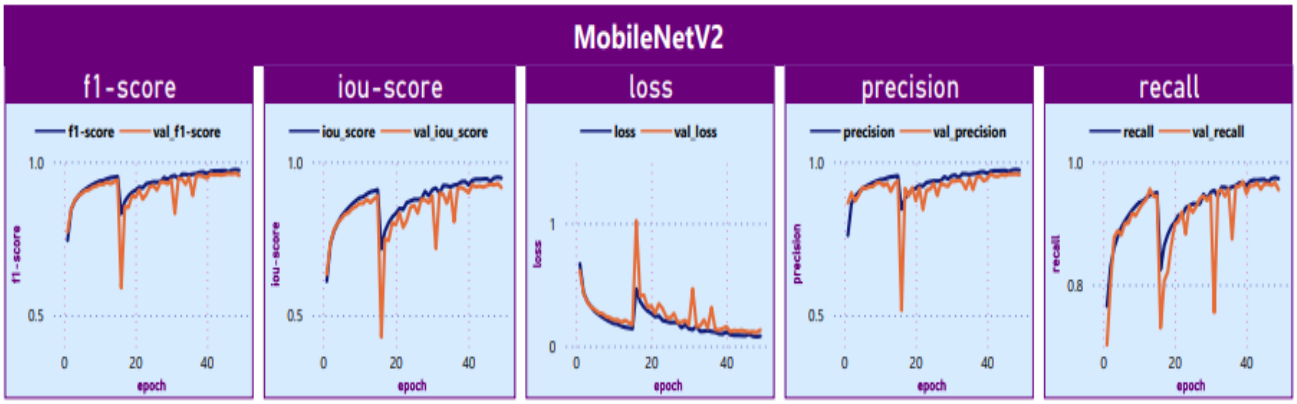


Figure 31. MobileNetV2 metrics. 1-49 epochs

5.7.2 U-Net+EfficientNetB2

U-Net with EfficientNetB2 is also further trained for 34 epochs with encoder and decoder and the metrics for training vs. validation is plotted below for the 49 epochs. Here also we can see F1 and IoU scores drop from 16th epoch and gradually increases.

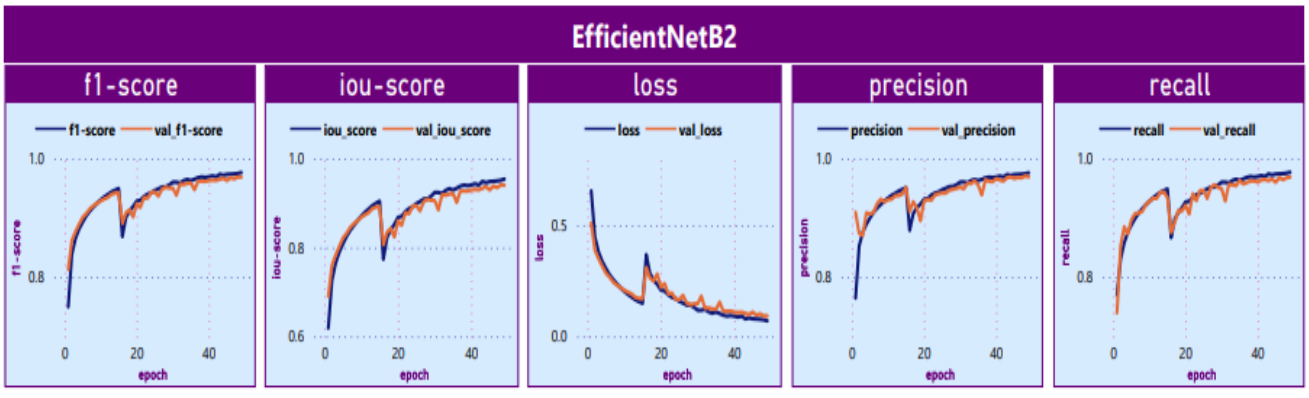


Figure 32. EfficientNetB2 metrics, 1-49 epochs

5.7.3 Comparison of U-Net+EfficientNetB2 and U-Net+MobileNetV2

Below table shows the model test results for U-Net+mobileNetV2 and U-Net+EfficientB2 after training for 49 epochs. The F1 score for U-Net+EfficientNetB2 is 0.9682 and IoU score is 0.9392 which performed better than MobileNetV2. The F1 score for MobileNetV2 is 0.9562 and IoU score is 0.9167.

Test scores for U-Net+Backbone - MobileNetV2 & EfficientNetB2

U-Net+Backbone	f1-score	iou_score	loss	precision	recall
EfficientNetB2	0.9682	0.9392	0.0958	0.9692	0.9679
MobileNetV2	0.9562	0.9167	0.1318	0.9578	0.9550

Table 2. Comparing EfficientNetB2 and MobileNetV2 after 49 epochs

Below graph show the comparison of U-Net+MobileNetV2 and U-Net+EfficientB2 after the training the models for 49 epochs and testing on test data.

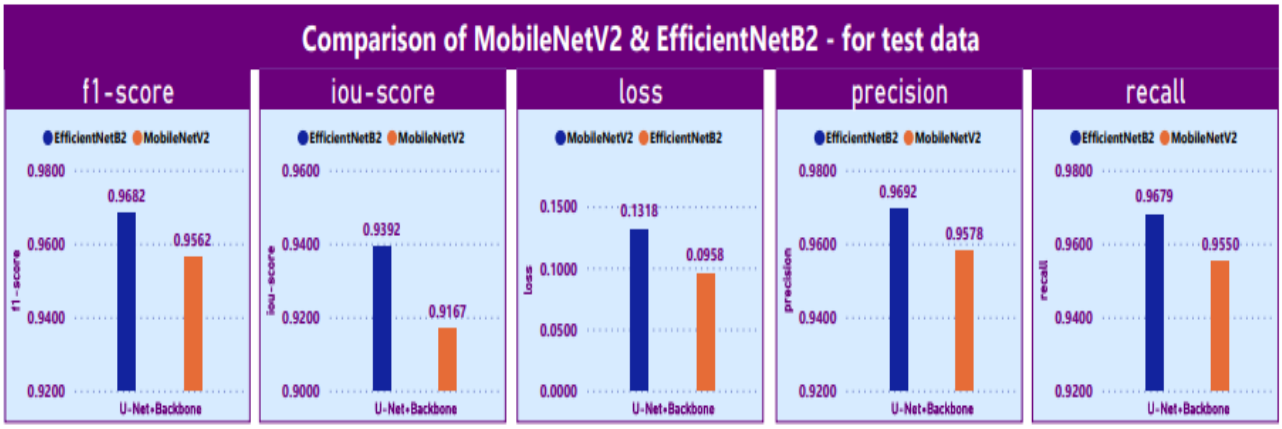


Figure 33.Comparing Metrics - MobileNetV2 & EfficientNetB2 - for test data

5.7.4 Predictions

Below are the images from test data and predictions by U-Net with EfficientNetB2 and MobileNetV2.

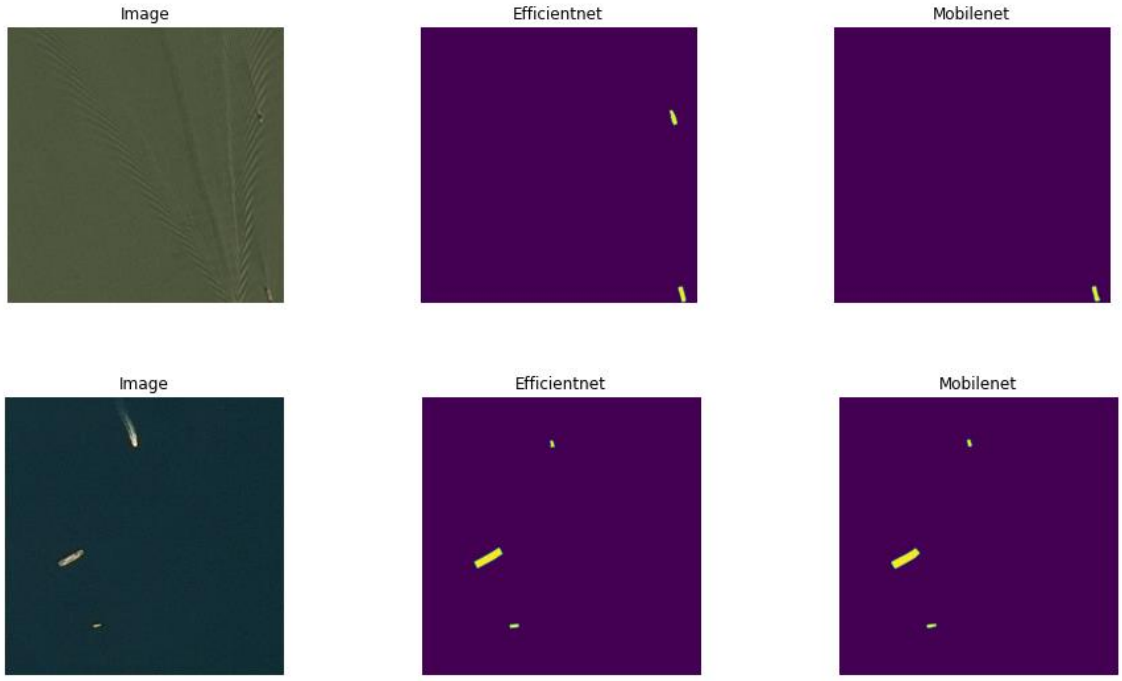


Figure 34. Predictions

5.8. Conclusions

The goal of this study was to evaluate the ship detection using U-Net with well-known transfer learning backbone networks. The U-Net with backbone DenseNet-121, ResNet-50, VGG-16, MobileNetV2, EfficientNetB2, and InceptionV3 were trained initially for 15 epochs with freezing the encoder. U-Net with EfficientNetB2 and MobileNetV2 outperformed the other models.

U-Net with EfficientNetB2 and MobileNetV2 were further trained for 34 epochs (total 49 epochs) for the same data and EfficientNetB2 with F1 score 0.9682 and IoU score 0.9392 performed better than MobileNetV2.

6 FURTHER IMPROVEMENTS

- Further improvements can be done using U-Net with different backbones and versions.
- Also we can further train the model to extract more features and predict finer boundary.
- Other image segmentation model like PSPNet, DeepLab etc. can be used with different backbones.
- Data augmentation can be done to make the model robust by passing augmented images. The different data augmentation techniques like RGB shift, distort image, zoom out etc. can be used.
- Ablation study can be done by adding/removing few layers from the backbone.

.

7 REFERENCES

Airbus ship detection challenge (2018) *Kaggle.com*. Available at: https://www.kaggle.com/c/airbus-ship-detection/data (Accessed: June 7, 2022).

Alghazo, J. *et al.* (2021) "Maritime ship detection using convolutional neural networks from satellite images," in *10th IEEE International Conference on Communication Systems and Network Technologies (CSNT). IEEE; 2021*, pp. 432–437.

Alsabhan, W. and Alotaiby, T. (2022) "Automatic building extraction on satellite images using unet and ResNet50," *Computational intelligence and neuroscience*, 2022, p. 5008854. doi: 10.1155/2022/5008854.

Balodi, T. (2021) *Convolutional neural network with python code explanation, Analyticssteps.com*. Available at: https://www.analyticssteps.com/blogs/convolutional-neural-network-cnn-graphical-visualization-code-explanation (Accessed: May 9, 2022).

Chen, L.-C. *et al.* (2018) "Encoder-decoder with atrous separable convolution for semantic image segmentation," *arXiv [cs.CV]*. Available at: https://openaccess.thecvf.com/content_ECCV_2018/papers/Liang-Chieh_Chen_Encoder-Decoder_with_Atrous_ECCV_2018_paper.pdf (Accessed: June 8, 2022).

Chen, Y., Zheng, J. and Zhou, Z. (2018) *Airbus ship detection -traditional v.s. Convolutional neural network approach, Stanford.edu*. Available at: https://cs229.stanford.edu/proj2018/report/58.pdf (Accessed: May 9, 2022).

Chollet, F. (2016) "Xception: Deep learning with depthwise separable convolutions," *arXiv [cs.CV]*. Available at: http://arxiv.org/abs/1610.02357.

Fei-Fei Li & Justin Johnson & Serena Yeung (2017) *"Detection and Segmentation*," 10 May. Available at: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf.

Gaudenz Boesch (2021) *VGG Very Deep Convolutional Networks (VGGNet) - What you need to know - viso.ai*. Available at: https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/ (Accessed: May 9, 2022).

Gupta, T. (2017) *Deep learning: Feedforward neural network, Towards Data Science*. Available at: https://towardsdatascience.com/deep-learning-feedforward-neural-network-26a6705dbdc7 (Accessed: May 9, 2022).

Han, L. *et al.* (2020) "Multi-size convolution and learning deep network for SAR ship detection from scratch," *IEEE access: practical innovations, open solutions*, 8, pp. 158996–159016. doi: 10.1109/access.2020.3020363.

He, K. *et al.* (2016) "Identity mappings in deep residual networks," *arXiv [cs.CV]*. Available at: http://arxiv.org/abs/1603.05027.

How does satellite imaging work? (2021) *Globalforestlink.com*. Available at: https://globalforestlink.com/how-does-satellite-imaging-work (Accessed: June 8, 2022).

Huang, G. *et al.* (2016) "Densely connected convolutional networks." doi: 10.48550/ARXIV.1608.06993.

Iakubovskii, P. (2020) *Segmentation_models*.

IBM Cloud Education (2020) *What is deep learning? Ibm.com*. Available at: https://www.ibm.com/cloud/learn/deep-learning (Accessed: May 9, 2022).

Image Segmentation Guide (2021) *Fritz.ai*. Available at: https://www.fritz.ai/image-segmentation/ (Accessed: June 8, 2022).

Jordan, J. (2018) *Evaluating image segmentation models, Jeremy Jordan*. Available at: https://www.jeremyjordan.me/evaluating-image-segmentation-models (Accessed: June 8, 2022).

Kanjir, U., Greidanus, H. and Oštir, K. (2018) "Vessel detection and classification from spaceborne optical images: A literature survey," *Remote sensing of environment*, 207, pp. 1–26. doi: 10.1016/j.rse.2017.12.033.

Kurama, V. (2020) *A guide to ResNet, Inception v3, and SqueezeNet*, *Paperspace Blog*. Available at: https://blog.paperspace.com/popular-deep-learning-architectures-resnet-inceptionv3-squeezenet/ (Accessed: May 9, 2022).

Lendave, V. (2021). A Comparison of 4 Popular Transfer Learning Models [Online]. Analytics India Magazine. Available at: https://analyticsindiamag.com/a-comparison-of-4-popular-transfer-learning-models (Accessed: 27 May 2022)

Leonardblier, P. (2016) *A brief report of the Heuritech Deep Learning Meetup #5*, *Tech Blog*. Available at: https://heuritech.wordpress.com/2016/02/29/a-brief-report-of-the-heuritech-deep-learning-meetup-5 (Accessed: June 8, 2022).

Li, L. *et al.* (2020) "A novel CNN-based method for accurate ship detection in HR optical remote sensing images via rotated bounding box," *arXiv [cs.CV]*. Available at: https://hi.booksc.eu/dl/82170139/ab86b1 (Accessed: June 8, 2022).

Li, Y. *et al.* (2019) "SAR ship detection based on resnet and transfer learning," in *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE pp 1188-1191 Available at: https://doi.org/10.1109/IGARSS.2019.8900290

Oligschläger, C. (2020) *Detect ships in critical areas, EARSC and OGEO Portal: Bringing EO user communities together*. Available at: https://earsc-portal.eu/display/EOwiki/Detect+ships+in+critical+areas (Accessed: June 8, 2022).

Pujara, A. (2020) *MobileNet Convolutional neural network Machine Learning Algorithms, Analytics Vidhya*. Available at: https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470 (Accessed: May 9, 2022).

Pawan, S. (2020) *Residual networks (ResNet) - deep learning, GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning/ (Accessed: May 9, 2022).

Ronneberger, O., Fischer, P. and Brox, T. (2015) "U-Net: Convolutional Networks for Biomedical Image Segmentation," *arXiv [cs.CV]*. Available at: https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/ (Accessed: May 9, 2022).

Schulz, H. and Behnke, S. (2012) "Deep learning: Layer-wise learning of feature hierarchies," *KI - Künstliche Intelligenz*, 26(4), pp. 357–363. doi: 10.1007/s13218-012-0198-z.

Seth, N. (2021) *How does Backward Propagation Work in Neural Networks, Analytics Vidhya*. Available at: https://www.analyticsvidhya.com/blog/2021/06/how-does-backward-propagation-work-in-neural-networks/ (Accessed: May 9, 2022).

Shanmugamani, R. (2018) *Deep Learning for Computer Vision: Expert techniques to train advanced neural networks using TensorFlow and Keras*. Birmingham, England: Packt Publishing.

Simonyan, K. and Zisserman, A. (2014) "Very deep convolutional networks for large-scale image recognition," *arXiv [cs.CV]*. Available at: http://arxiv.org/abs/1409.1556.

Singhal, G. (2020) *Introduction to DenseNet with TensorFlow, Pluralsight.com*. Available at: https://www.pluralsight.com/guides/introduction-to-densenet-with-tensorflow (Accessed: June 8, 2022).

Szegedy, C. *et al.* (2014) "Going deeper with convolutions," *arXiv [cs.CV]*. Available at: http://arxiv.org/abs/1409.4842.

Tan, M. and Le, Q. V. (2019) "EfficientNet: Rethinking model scaling for convolutional Neural Networks," *arXiv [cs.LG]*. Available at: http://arxiv.org/abs/1905.11946.

Tomar, N. (2021a) *What is residual network or ResNet? — idiot developer, Analytics Vidhya*. Available at: https://medium.com/analytics-vidhya/what-is-residual-network-or-resnet-idiot-developer-6a1daa7c3b09 (Accessed: June 8, 2022).

Tomar, N. (2021b) *What is UNET? - Analytics Vidhya - Medium, Analytics Vidhya*. Available at: https://medium.com/analytics-vidhya/what-is-unet-157314c87634 (Accessed: May 9, 2022).

Understanding of MobileNet - EN (2022) Available at: https://wikidocs.net/165429 (Accessed: June 8, 2022).

Yao, Y., Jiang, Z. and Zhang, H. (2016) *"High-resolution optical satellite image simulation of ship target in large sea scenes,"* in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE.

Zou J., Han Y., So SS. (2008) *Overview of Artificial Neural Networks. In: Livingstone D.J. (eds) Artificial Neural Networks. Methods in Molecular Biology*, vol 458. pp.14-22. Humana Press.

8 APPENDICES

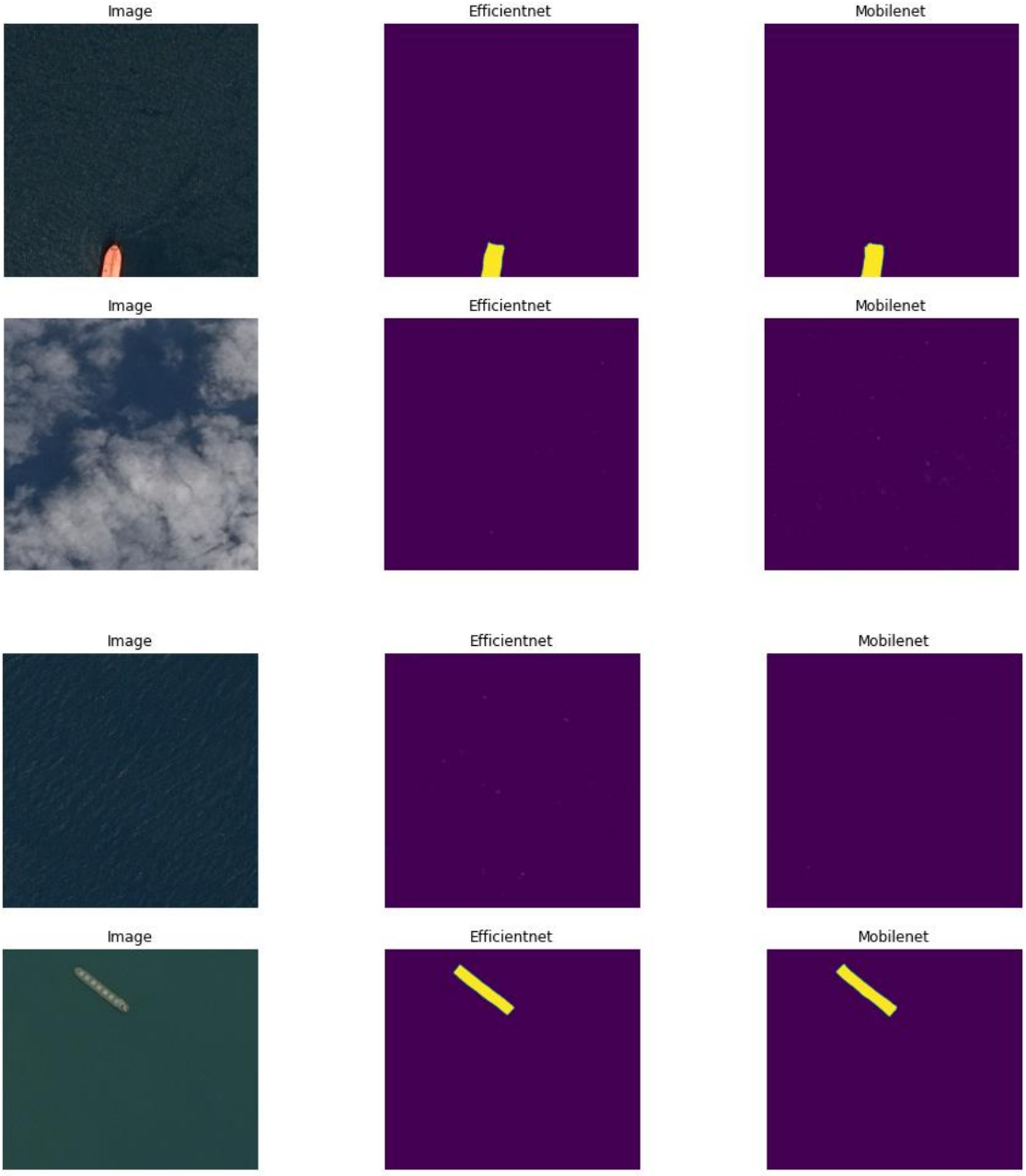
8.1 CSC-Puhti sample batch script

Example of batch script for training the image segmentation model with EfficientNetB2.

```
#SBATCH --job-name=EfficientNet-ship

#SBATCH --nodes=1

#SBATCH --ntasks-per-node=1

#SBATCH --cpus-per-task=30

#SBATCH --partition=gpu

#SBATCH --gres=gpu:v100:1,nvme:10

#SBATCH --time=35:00:00

#SBATCH --mem-per-cpu=12G

#SBATCH --account=project_2001220

Export PYTHONPATH=~/.local/lib/python3.8/site-packages/:$PYTHONPATH

#module purge

module load python-data/3.7.6-1

module load tensorflow/2.6

module list

module load plotly

module load segmentation-models==1.0.1

set -xv

srun python3 new_efficientnet.py
```

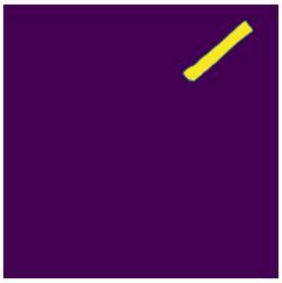
8.2 Image prediction results

Below images show the original image and the predictions using U-Net with EfficientNetB2 and MobileNetV2 for the test images.

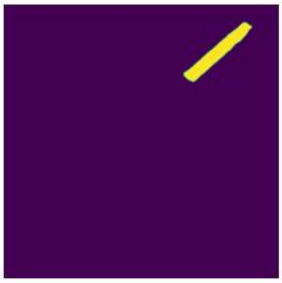


Image

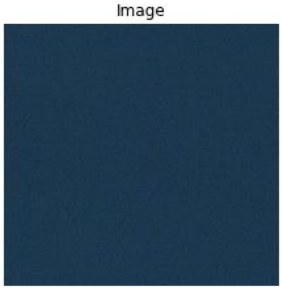
Efficientnet



Mobilenet



Image



Efficientnet

Mobilenet

Image

Efficientnet

Mobilenet

Image

Efficientnet

Mobilenet

Image

Efficientnet

Mobilenet

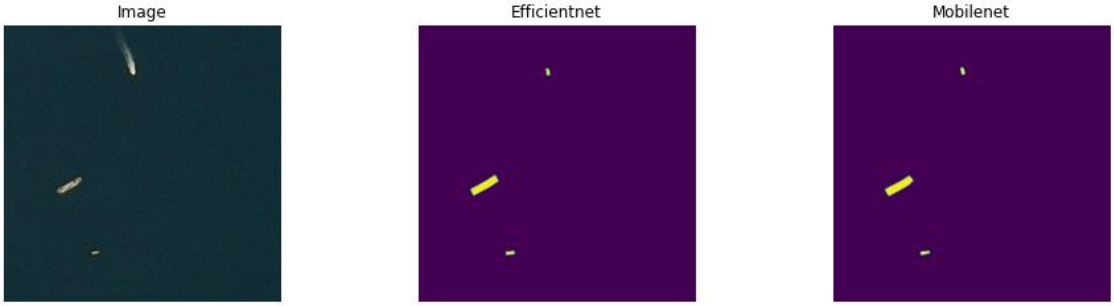


Image
Efficientnet
Mobilenet