T.R.

GEBZE TECHNICAL UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

CLASSIFICATION OF SINGLE CELL MARINE CREATURES: PHYTOPLANKTON

MEHMET AVNI ÇELİK

SUPERVISOR
DR. BURCU YILMAZ

GEBZE
2023

T.R.
GEBZE TECHNICAL UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

CLASSIFICATION OF SINGLE CELL MARINE CREATURES: PHYTOPLANKTON

MEHMET AVNI ÇELİK

SUPERVISOR
DR. BURCU YILMAZ

2023
GEBZE

GEBZE TECHNICAL UNIVERSITY	GRADUATION PROJECT JURY APPROVAL FORM

This study has been accepted as an Undergraduate Graduation Project in the Department of Computer Engineering on 19/01/2023 by the following jury.

JURY

Member
(Supervisor) : Dr. Burcu YILMAZ

Member : Dr. Burcu YILMAZ

Member : Doç Dr. Habil KALKAN

ABSTRACT

Phytoplanktons are single cell microscopic marine creatures which are one of the most important resources of the seas, important in various fields such as food, agriculture, medicine and pharmacy.

There are more than 20.000 species of phytoplanktons and that makes challenging to identify them without recognition and classification algorithms.

In this study, the problem mentioned above is solved by using image processing and deep learning techniques such as ResNet50 and VGG16.

Keywords: ResNet50, VGG16, Phytoplankton.

ÖZET

Tek hücreli mikroskobik deniz canlıları olan fitoplanktonlar; tarım,hayvancılık, tıp ve eczacılık gibi çeşitli alanlarda oldukça önemli bir kaynak olarak kabul edilmektedir.

20.000'den fazla çeşidi olan fitoplanktonlar, benzer şekilleri ve özelliklerinden dolayı görüntü işleme ve derin öğrenme teknikleri olmadan ayırt edilmesi mümkün olmayan yapılara sahiptirler.

Bu çalışmada, yukarıda bahsedilen sorunlar için görüntü işleme ve VGG16, ResNet50 gibi derin öğrenme yöntemleri kullanılarak çözüm geliştirilmiştir.

Anahtar Kelimeler: ResNet50, VGG16, Fitoplankton.

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to my supervisors, Dr. Burcu Yılmaz, Dr. Raif Yuecel and Dr. Atilla Bebes, for their help, guidance, and mentorship throughout this project. A debt of gratitude is also owed to my colleague Elif Keles who did not leave any of my questions unanswered.

I am very grateful to all my teachers, especially Yusuf Sinan Akgul, Yakup Genc, Alp Arslan Bayrakci, Fatih Erdogan Sevilgen, Habil Kalkan and Erkan Zergeroglu for teaching me how to code and think as an engineer. Without their constant guidance and feedback, this project would not have materialized.

Finally, I would also like to thank my family, who unconditionally supported my academic career. I owe all that I am, or ever hope to be, to my entire family.

Mehmet Avni CELIK

LIST OF SYMBOLS AND ABBREVIATIONS

Symbol or Abbreviation	:	**Explanation**
CNN	:	Convolutional Neural Network
ResNet50	:	Pretrained Residual Neural Network
VGG16	:	Pretrained CNN Model
RGB	:	Red-Green-Blue
GTU	:	Gebze Technical University
Ch	:	Channel

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Phytoplanktons are single cell microscopic marine creatures which are one of the most important resources of the seas, important in various fields such as food, agriculture, medicine and pharmacy. There are more than 20.000 species of phytoplanktons and that makes challenging to identify them without recognition and classification algorithms..

Phytoplankton images are obtained by a method called flow cytometry which is a technique used to detect and measure physical and chemical characteristics of a population of cells or particles. The Cells are observed with different techniques. By this way a cell has 3 graysceled images with divergent channels such as brightfield image (Ch04), fluorescence (Ch05) and scatter signal (Ch06). 1.1

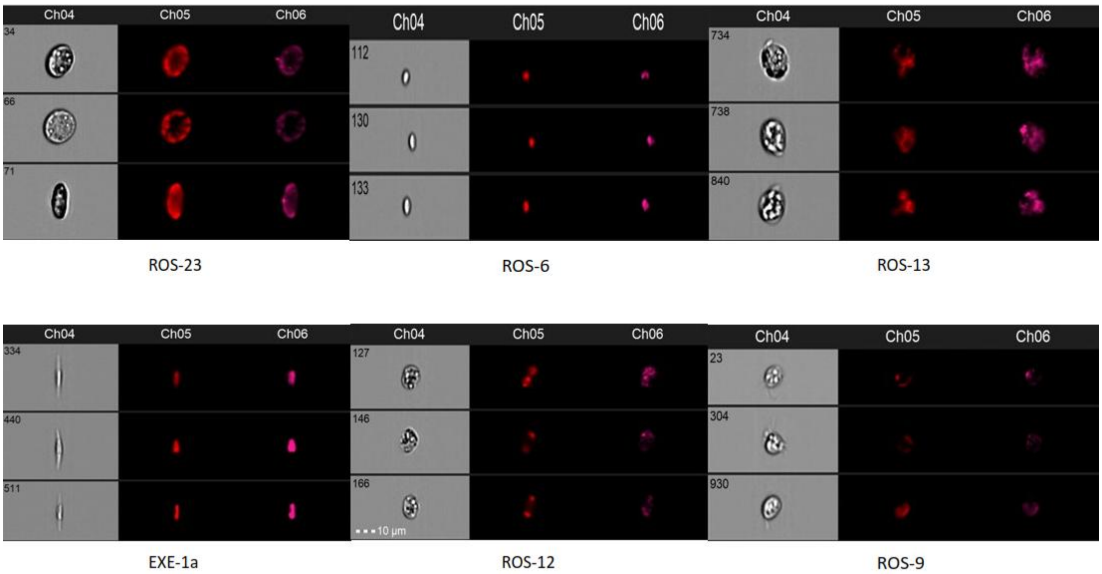

Figure 1.1: Different cell images with different channel

Grayscaled images can be converted to RGB images to use them for the deep learning models, yet a cell has 3 divergent channles. Therefore the channels can be combined and used in RGB structure.1.2

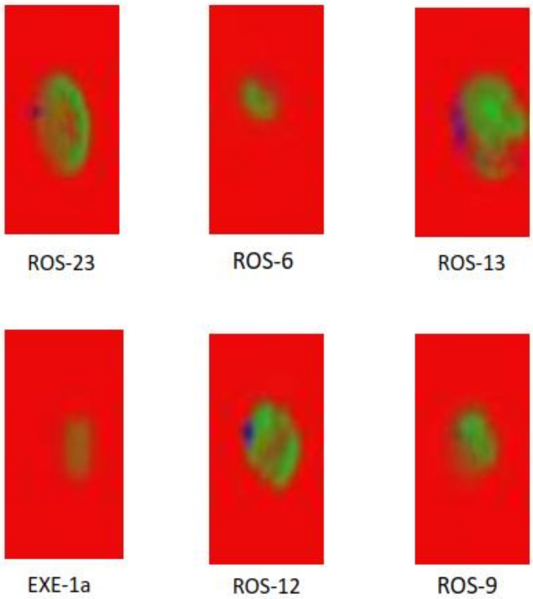

Figure 1.2: Cell images with combined channels

In this project, pre-trained deep learning models,ResNet50 and VGG16, are studied with different parameters and different usage of the dataset. Applied methods are listed below:

1) Individual channels are converted to RGB images.
2) Deep learning models are trained individually.
3) Images without 3 channels are removed from the dataset.
4) Channels are combined as RGB values.
5) Deep Learning models are trained with combined images.

1.1. Rudiments

1.1.1. Dataset

The dataset is provided by Dr. Ian Probert (Roscoff Institute, France) and cultures were kept under standardised conditions until measurement.
Each of the cell channels has 3000 to 4000, hence a cell has more than 9000 individual images. The data is splitted by 30%-70% for testing and training sets. Data augmentation methods are not applied due to the images are unique and will be in the same form when a new cell is observed with flow cytometry machine.

1.1.2. Libraries

Within the scope of the project, the libraries mentioned below were used:
1)Rasterio: processing 'tif' extension.
2)OpenCV: Image processing techniques.
3)Keras,Tensorflow: Deep Learning Methods.
Other requirements can be found in the requirements.txt file in the project folder.

1.1.3. Deep Learning Models

ResNet50 and VGG16 models are studied on account of the literature research performed. One of the most important property of ResNet50 is overcoming the 'vanishing gradient problem' which causes the value of gradient decreases significantly during backpropagation. In addition, it is observed that similar project such as "blood cell classification"[1] or "mitosis detection in breast canser"[2] are developed by ResNet50 model. Therefore, it is observed that the accuracy of ResNet in cell datasets are better than other models.

2. TRAINING MODELS

The models are trained with different ways after the dataset is manipulated and organized. There are 2500 images for training of each channel of cells and 750 images for testing the model. The usage of dataset is explained below:

1)Training with channel 4
2)Training with channel 5
3)Training with channel 6
4)Training with combined channels as RGB

2.0.1. Training with Channel 4

Channel 4 images are bright-field images of cells with width:32 and height:64 pixels. Some example cells are showed below.2.1 —

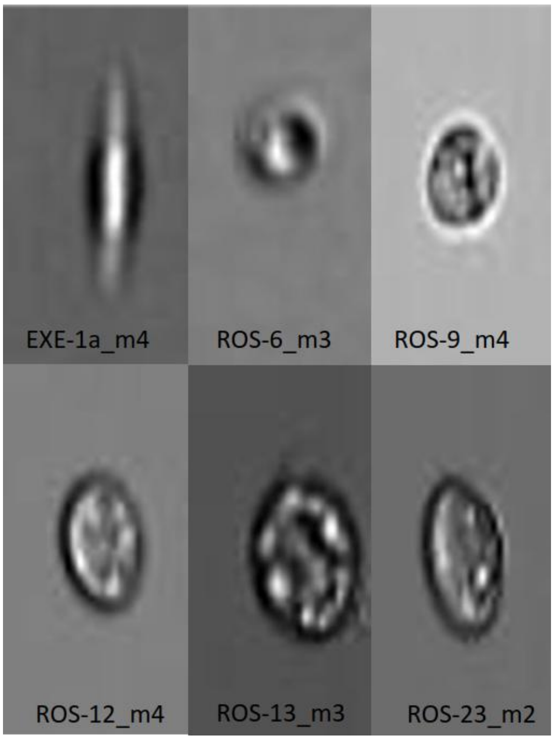

Figure 2.1: Different cell images with channel 4

2.0.2. Training with Channel 5

Channel 5 images are fluorescence images of cells with width:32 and height:64 pixels. Some example cells are showed below.2.2

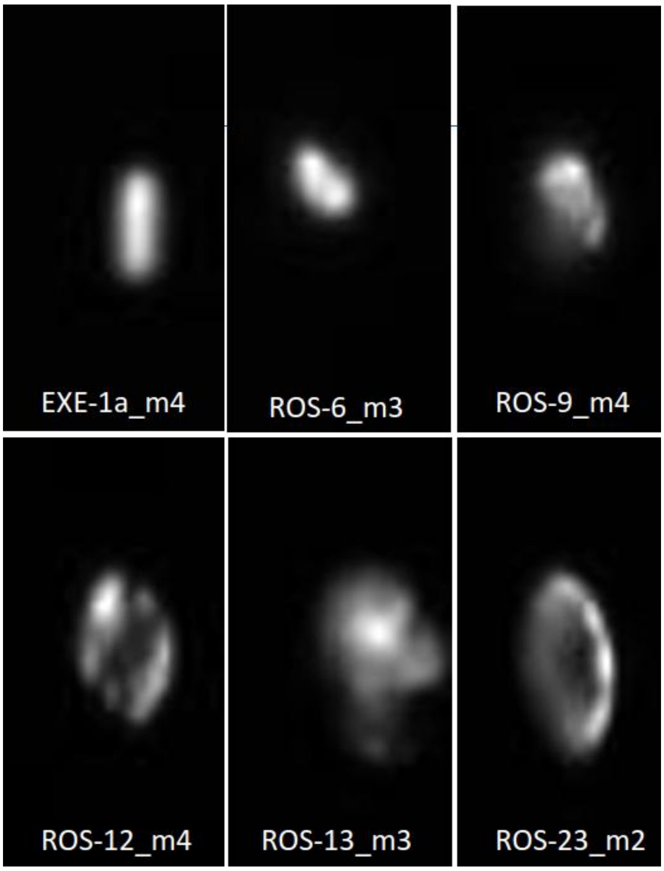

Figure 2.2: Different cell images with channel 5

2.0.3. Training with Channel 6

Channel 6 images are scatter signal images of cells with width:32 and height:64 pixels. Some example cells are showed below.2.3

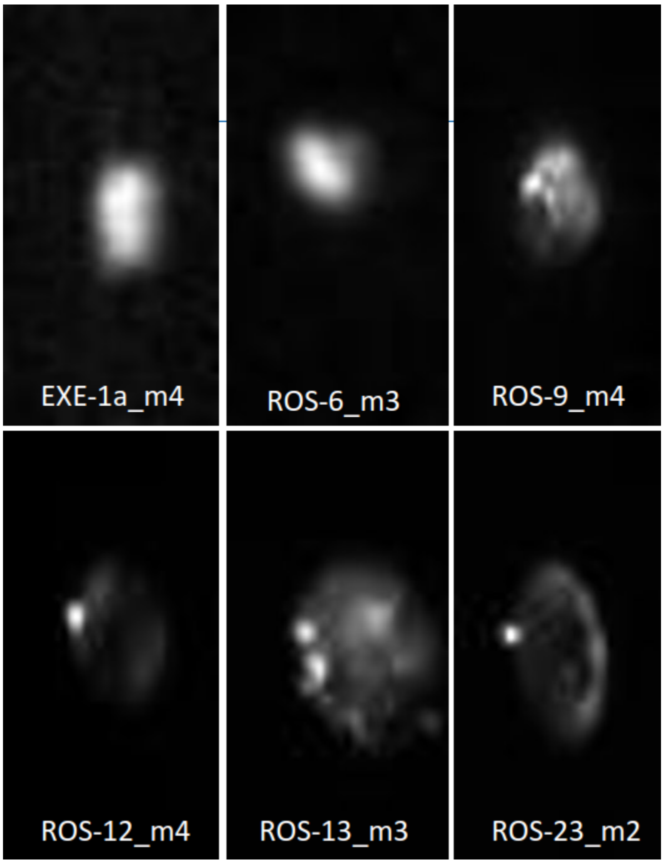

Figure 2.3: Different cell images with channel 6

2.0.4. Training with Combined Images as RGB

A cell has 3 divergent images and all can be used for training the model. However the cell, the structure and the imaging angle is same. Therefore a relationship between the channels can be tuneable.In the following visualisations, cell images are created by positioning the channel 4 as R value, channel 5 as G value and channel 6 as B value. 2.4

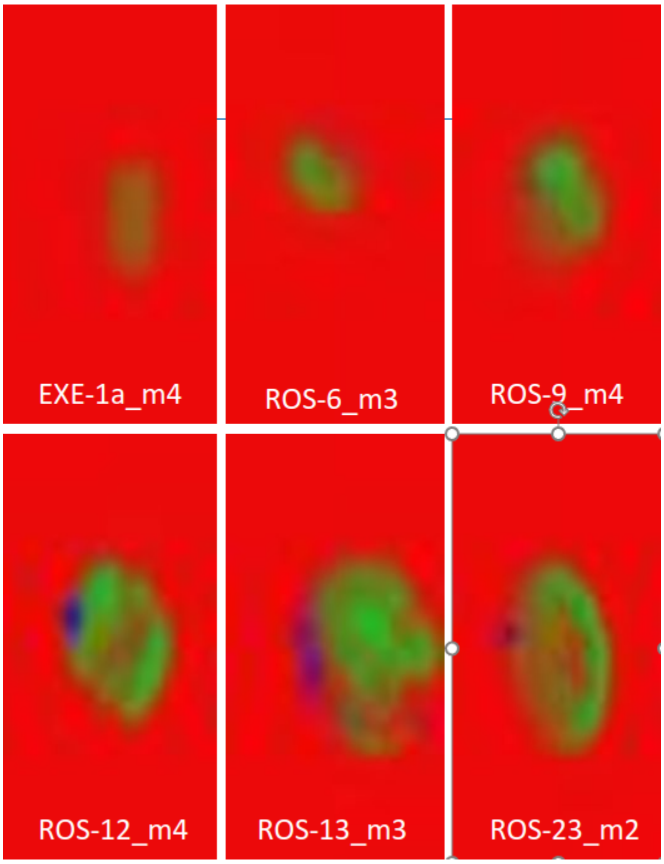

Figure 2.4: Different cell images with RGB colors

2.1. ResNet50

ResNet stands for Residual Network and is a specific type of convolutional neural network (CNN) introduced in the 2015 paper "Deep Residual Learning for Image Recognition" by He Kaiming, Zhang Xiangyu, Ren Shaoqing, and Sun Jian. CNNs are commonly used to power computer vision applications. ResNet-50 is a 50-layer convolutional neural network (48 convolutional layers, one MaxPool layer, and one average pool layer).
The ResNet model used in this project is pre-trained with imagenet dataset. 2.5

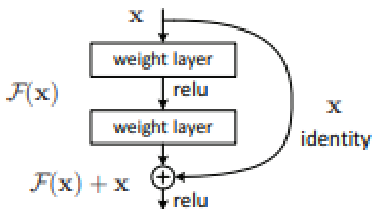

Figure 2.5: Residual learning: a building block[3]

Models should be trained with different ways to find most accurate weights and parameters. To achieve that, different epoch values,callbacks such as early-stopping and different layers can be implemented to the model. After that the model with best weights can be saved and used.

2.1.1. Fine-Tuning

The images belonging to the same channels are extracted individually. Therefore channels should be studied individually in the same way as RGB images as well. The results of each of them will be shown in the next section. Initially, to prove fine tuning effects the result, channel 4 will be studied . Afterwards the best tuned model's results' will be shown.
Studied model structures are explained below.

Algorithm 1 Model I: With Less Dense Layer

$resnetModel \leftarrow sequential()$
$pretrainedModel \leftarrow resnet50$
$layerTrainable \leftarrow False$
$resnetModel.add(pretrainedModel)$
$resnetModel.add(Flatten())$
$resnetModel.add(Dense(512, activation = relu)$
$resnetModel.add(Dense(6, activation = softmax)$

Algorithm 2 Model II: With More Dense Layer

$resnetModel \leftarrow sequential()$
$pretrainedModel \leftarrow resnet50$
$layerTrainable \leftarrow False$
$resnetModel.add(pretrainedModel)$
$resnetModel.add(Flatten())$
$resnetModel.add(Dense(512, activation = relu)$
$resnetModel.add(Dense(256, activation = relu)$
$resnetModel.add(Dense(128, activation = relu)$
$resnetModel.add(Dense(6, activation = softmax)$

Algorithm 3 Model III: More Dense Layer with Dropout Layer

$resnetModel \leftarrow sequential()$
$pretrainedModel \leftarrow resnet50$
$layerTrainable \leftarrow False$
$resnetModel.add(pretrainedModel)$
$resnetModel.add(Flatten())$
$resnetModel.add(Dense(512, activation = relu)$
$resnetModel.add(Dropout(0.2)$
$resnetModel.add(Dense(256, activation = relu)$
$resnetModel.add(Dropout(0.2)$
$resnetModel.add(Dense(128, activation = relu)$
$resnetModel.add(Dropout(0.2)$
$resnetModel.add(Dense(6, activation = softmax)$

2.1.2. Results

The accuracy rates, visualisation and related results of mentioned models are displayed. Models are trained with 10 epoch, yet Early-Stopping callback halts in the proper position by monitoring validation loss.

2.1.2.1. Channel 4 Dataset

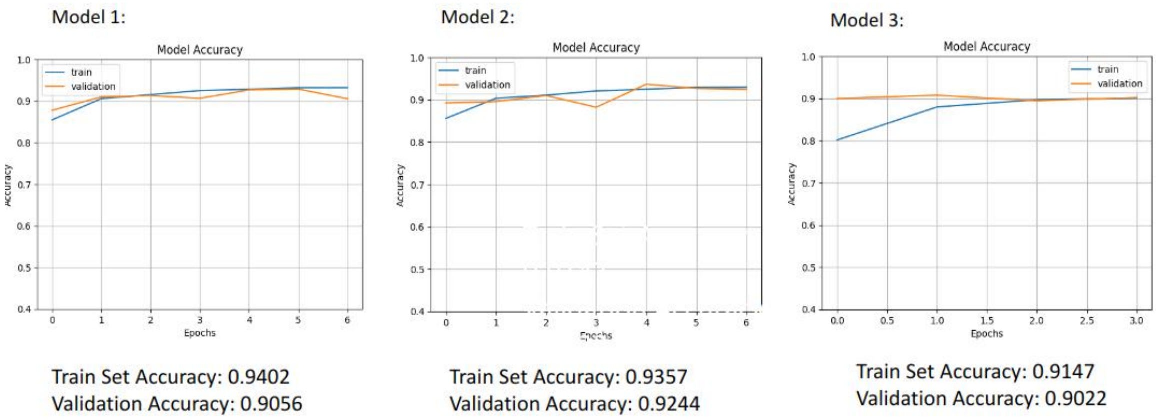

Figure 2.6: Model Accuracies of Channel 4 Dataset

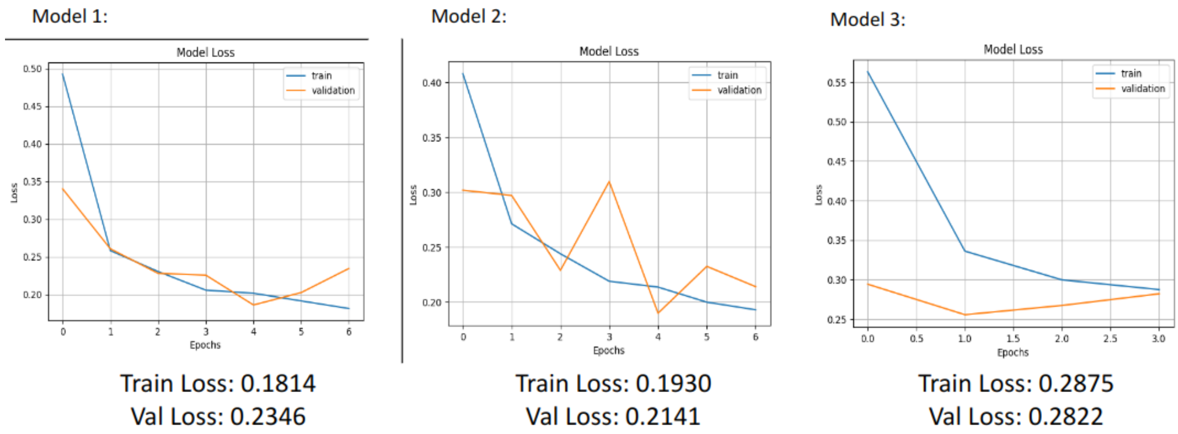

Figure 2.7: Model Loss of Channel 4 Dataset

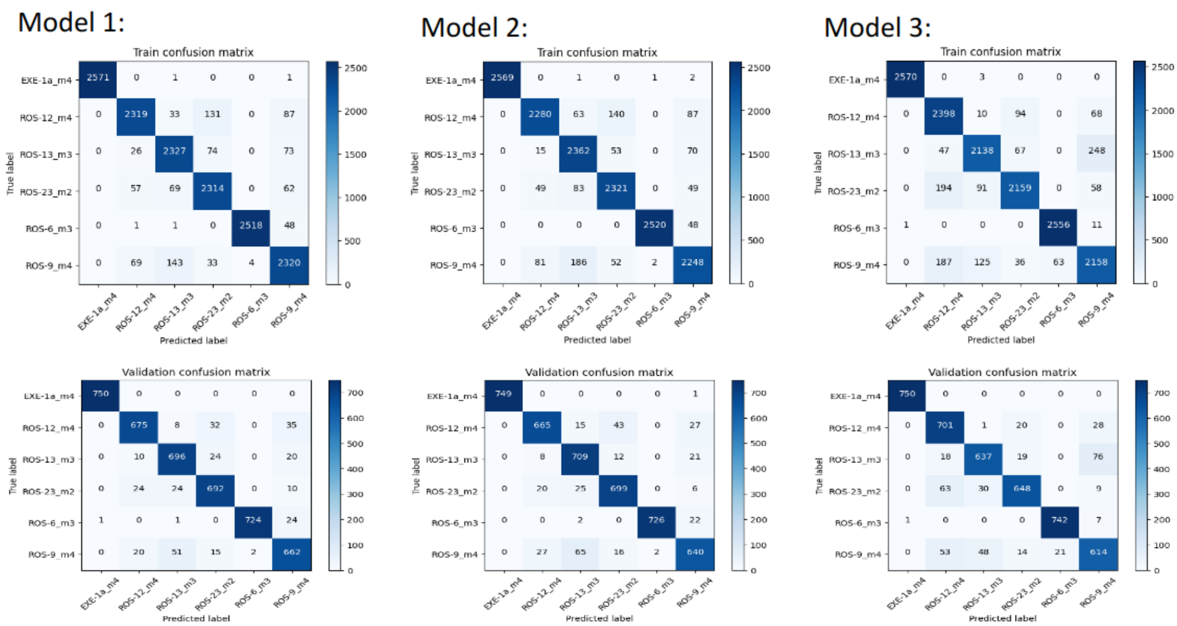

Figure 2.8: Confusion Matrices of Channel 4 Dataset

Tested models prove that fine-tuning techniques causes non-negligible differences. Even though the accuracy of it is lower than the others, for the sake of preventing overfitting Model 3 structure is selected and explained in the next sections.

2.1.2.2. Channel 5 Dataset

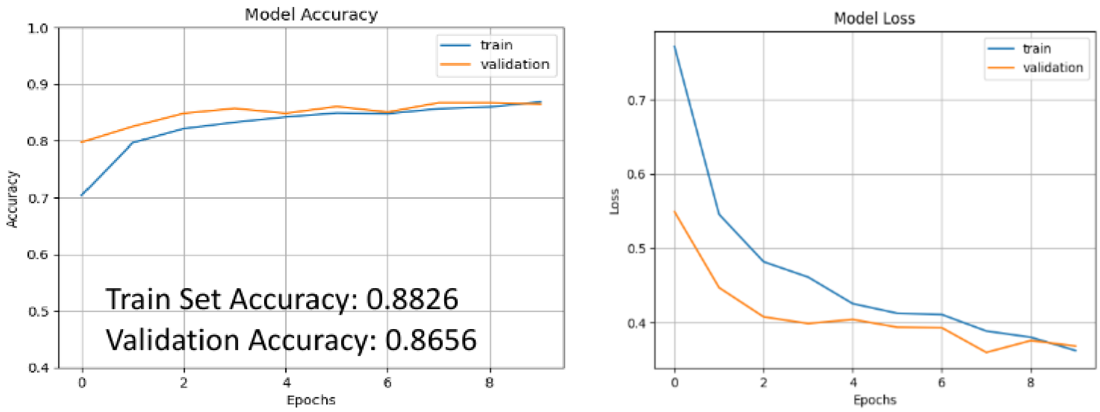

Figure 2.9: Model Accuracy and Loss of Channel 5 Dataset

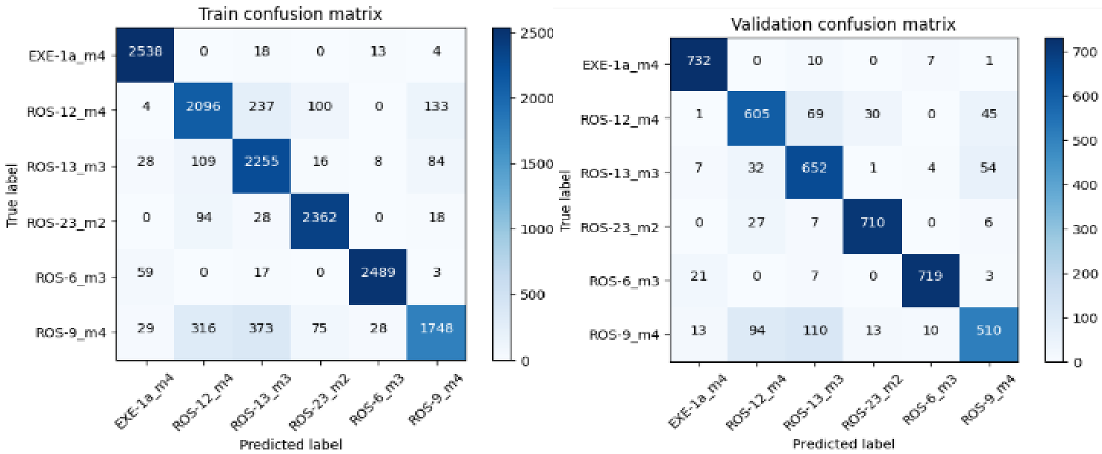

Figure 2.10: Train and Validation Confusion Matrices of Channel 5 Dataset

2.1.2.3. Channel 6 Dataset

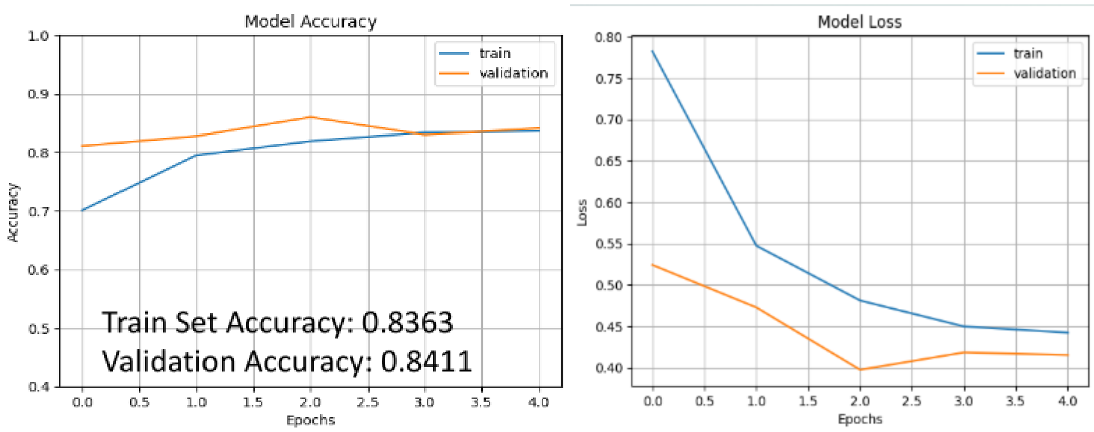

Figure 2.11: Model Accuracy and Loss of Channel 6 Dataset

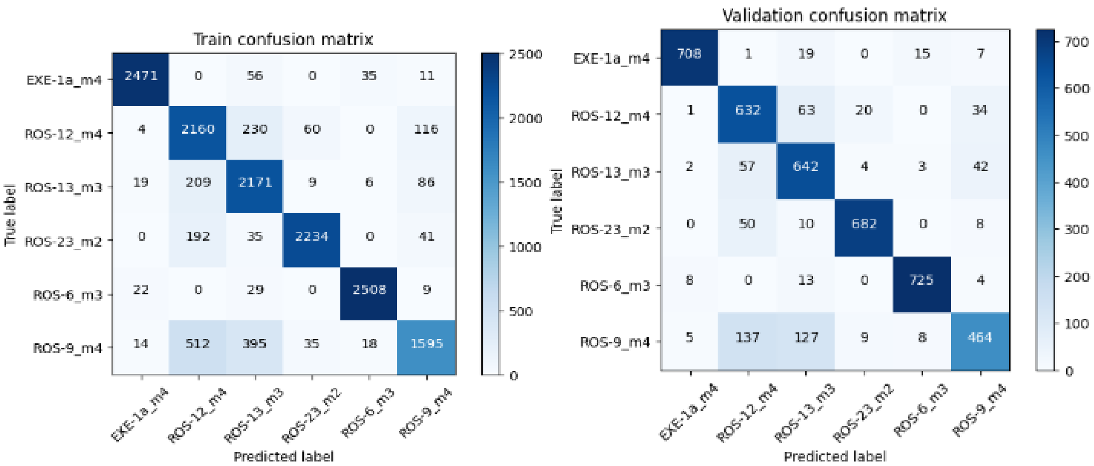

Figure 2.12: Train and Validation Confusion Matrices of Channel 6 Dataset

2.1.2.4. RGB Dataset

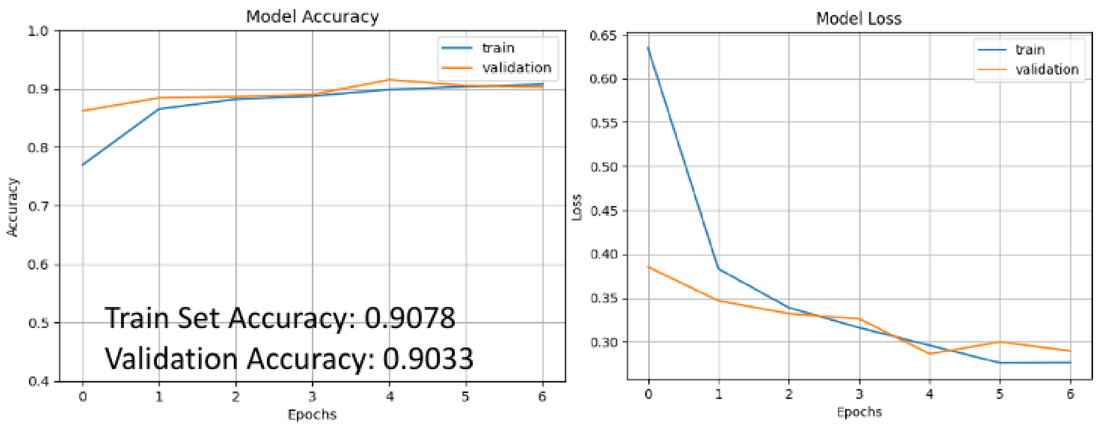

Figure 2.13: Model Accuracy and Loss of RGB Dataset

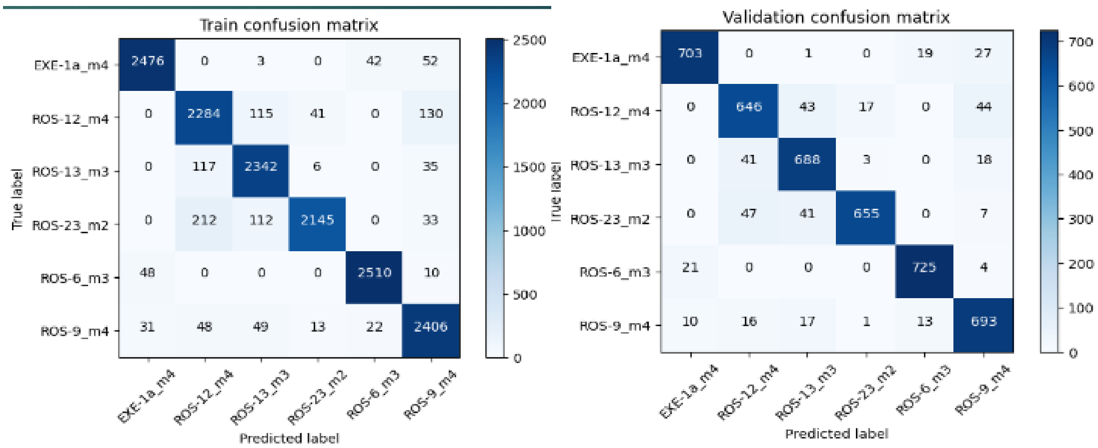

Figure 2.14: Train and Validation Confusion Matrices of RGB Dataset

2.1.3. Comparison of the Results

The obtained results in the earlier sections show that, processed data increase the accuracy and give better outcomes. Even though the model accuracy is sufficiently high, more fine tuning methods can be implemented to achieve more consistent results. Validation accuracies of different datasets is ranked below.

$$RGB > Ch4 > Ch5 > Ch6$$
$$90.33\% > 90.22\% > 86.56\% > 84.11\%$$

2.2. VGG16

VGG16 refers to the VGG model, also called VGGNet. It is a convolution neural network model supporting 16 layers. K. Simonyan and A. Zisserman from Oxford University proposed this model and published it in a paper called Very Deep Convolutional Networks for Large-Scale Image Recognition.[4]
The VGG16 model used in this project is pre-trained with imagenet dataset.

2.2.1. Fine-Tuning

Channels are studied individually in the same way as RGB images as it is in the ResNet-50 model. The results of each of them will be shown in the next section. Some fine tuning methods are attempted as explained below. Nevertheles, for the sake of preventing overfitting, Model 3 structure is selected and explained in the next sections.

Algorithm 4 Model I: With Less Dense Layer

$vggModel \leftarrow sequential()$
$pretrainedModel \leftarrow vgg16$
$layerTrainable \leftarrow False$
$vggModel.add(pretrainedModel)$
$vggModel.add(Flatten())$
$vggModel.add(Dense(512, activation = relu)$
$vggModel.add(Dense(6, activation = softmax)$

Algorithm 5 Model II: With More Dense Layer

$vggModel \leftarrow sequential()$
$pretrainedModel \leftarrow vgg16$
$layerTrainable \leftarrow False$
$vggModel.add(pretrainedModel)$
$vggModel.add(Flatten())$
$vggModel.add(Dense(512, activation = relu)$
$vggModel.add(Dense(256, activation = relu)$
$vggModel.add(Dense(128, activation = relu)$
$vggModel.add(Dense(6, activation = softmax)$

Algorithm 6 Model III: More Dense Layer with Dropout Layer

$vggModel \leftarrow sequential()$
$pretrainedModel \leftarrow vgg16$
$layerTrainable \leftarrow False$
$vggModel.add(pretrainedModel)$
$vggModel.add(Flatten())$
$vggModel.add(Dense(512, activation = relu)$
$vggModel.add(Dropout(0.2)$
$vggModel.add(Dense(256, activation = relu)$
$vggModel.add(Dropout(0.2)$
$vggModel.add(Dense(128, activation = relu)$
$vggModel.add(Dropout(0.2)$
$vggModel.add(Dense(6, activation = softmax)$

2.2.2. Results

The accuracy rates, visualisation and related results of mentioned models are displayed. Models are trained with 10 epoch, yet Early-Stopping callback halts in the proper position by monitoring validation loss.

2.2.2.1. Channel 4 Dataset

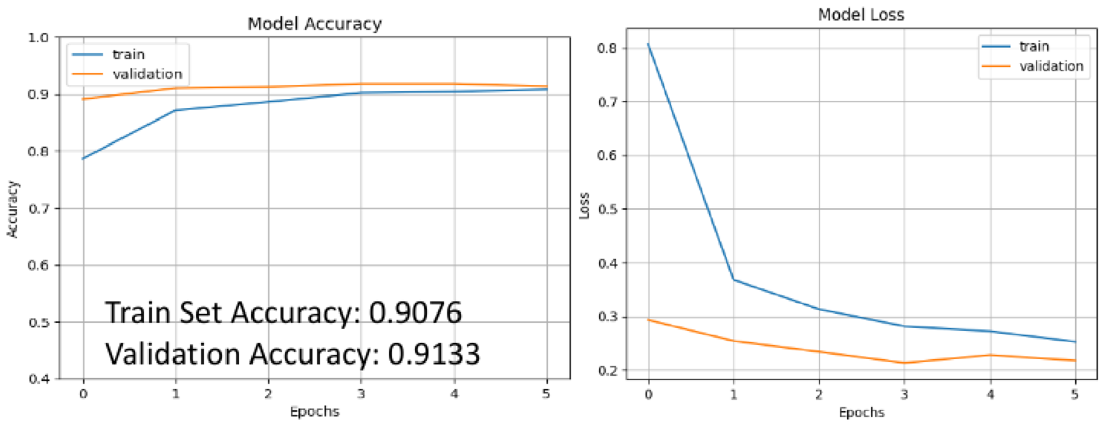

Figure 2.15: Model Accuracy and Loss of Channel 4 Dataset

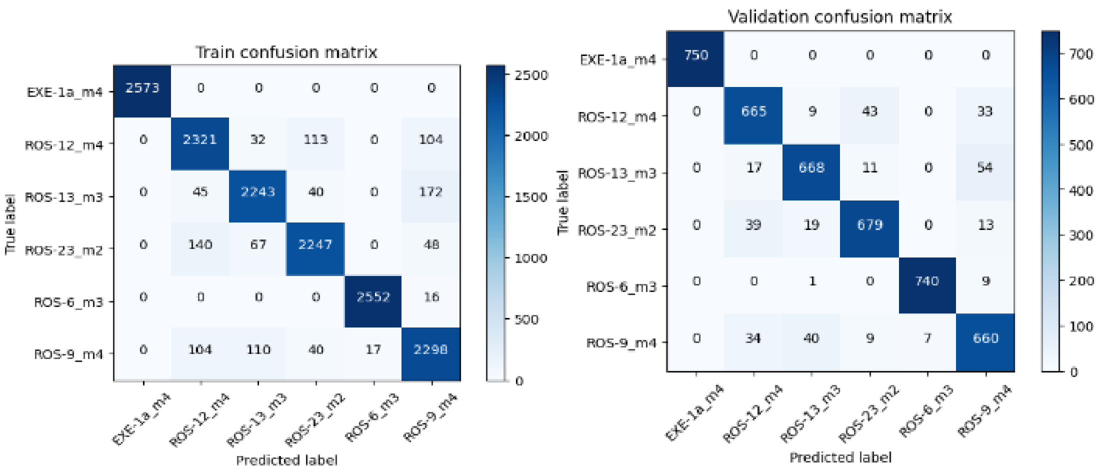

Figure 2.16: Train and Validation Confusion Matrices of Channel 4 Dataset

2.2.2.2. Channel 5 Dataset

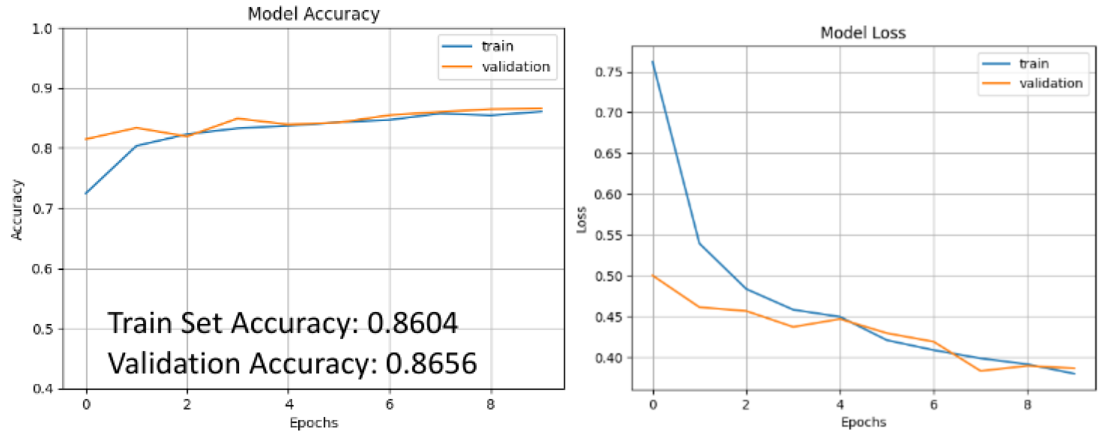

Figure 2.17: Model Accuracy and Loss of Channel 5 Dataset

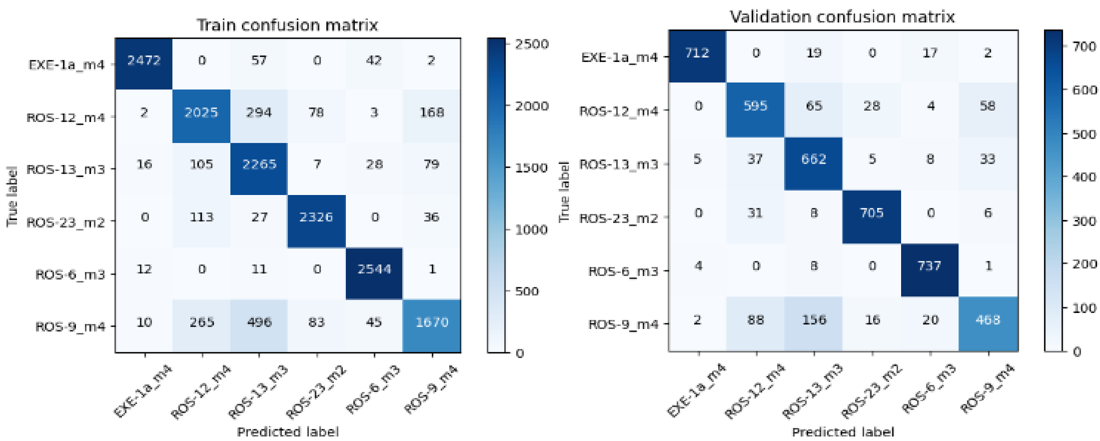

Figure 2.18: Train and Validation Confusion Matrices of Channel 5 Dataset

2.2.2.3. Channel 6 Dataset

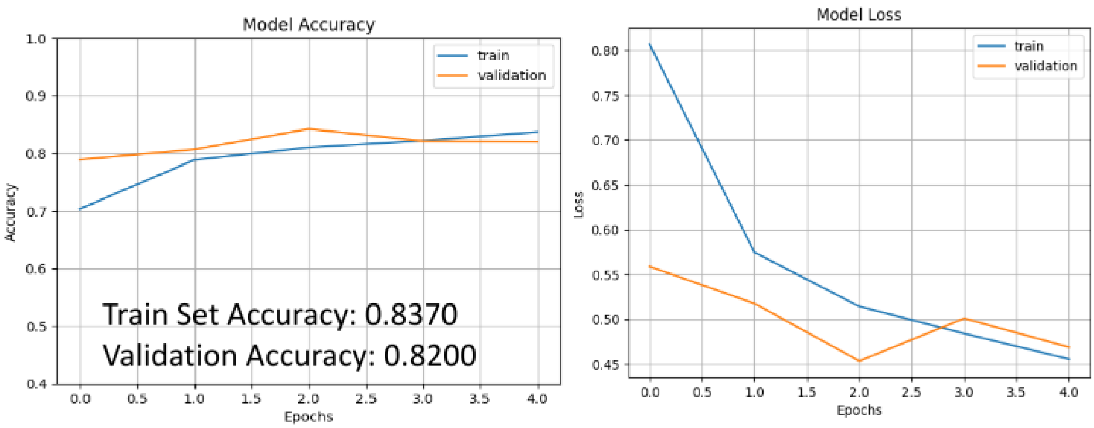

Figure 2.19: Model Accuracy and Loss of Channel 6 Dataset

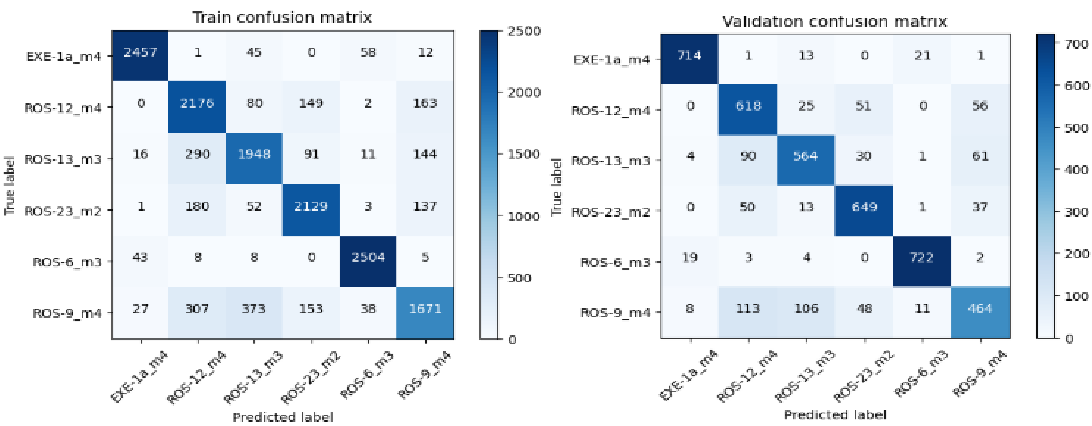

Figure 2.20: Train and Validation Confusion Matrices of Channel 6 Dataset

2.2.2.4. RGB Dataset

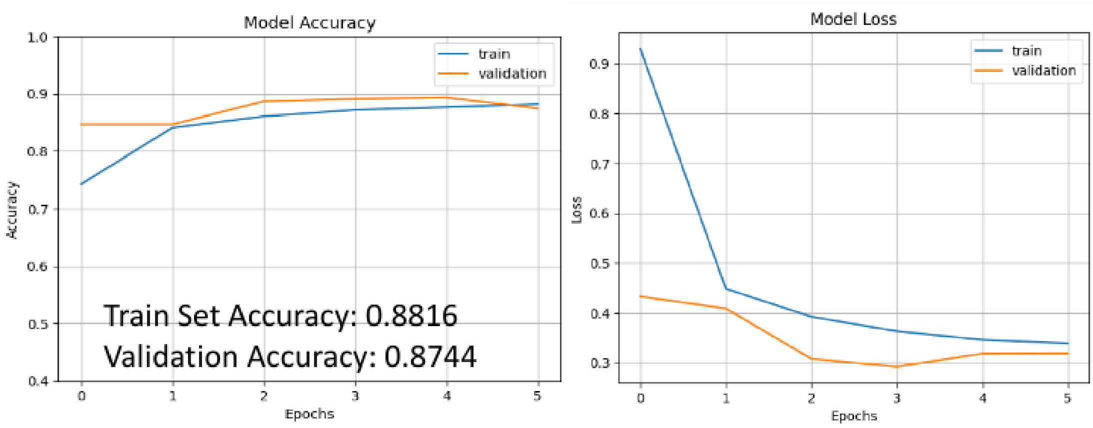

Figure 2.21: Model Accuracy and Loss of RGB Dataset

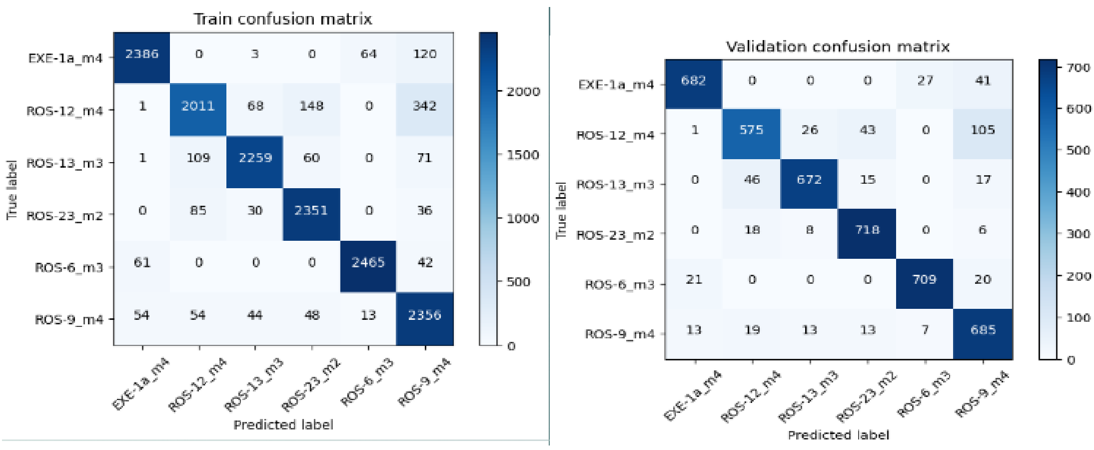

Figure 2.22: Train and Validation Confusion Matrices of RGB Dataset

2.2.3. Comparison of the Results

The obtained results in the earlier sections show that, processed data increase the accuracy and give better outcomes in VGG16 as it is with ResNet-50.
The accuracy rates in VGG16 did not achieved as it is supposed to be. The difference between train and validation accuracy shows that VGG16 model did not work well for the dataset since the validation accuracy is more than the train accuracy. Most possible reason for the mentioned problem is underfitting due to inadequate images in the dataset.
However, combining channels as RGB values slightly helps to overcome the underfitting problem. Validation accuracies of different datasets is ranked below.

$$Ch4 > RGB > Ch5 > Ch6$$
$$91.33\% > 88.16\% > 86.56\% > 82.00\%$$

3. IMPROVEMENTS

Even though it is not a generic problem for ResNet-50, model did not produce reliable results for some dataset-hyperparameter combinations. Therefore model structure is modified as shown in the code below, to overcome the mentioned underfitting problem.

Algorithm 7 Model IV: Modifid Model Structure

$resnetModel \leftarrow sequential()$
$pretrainedModel \leftarrow resnet50$
$layerTrainable \leftarrow False$
$resnetModel.add(pretrainedModel)$
$resnetModel.add(Flatten())$
$resnetModel.add(Dense(512, activation = relu)$
$resnetModel.add(Dense(256, activation = relu)$
$resnetModel.add(Dense(128, activation = relu)$
$resnetModel.add(Dropout(0.25)$
$resnetModel.add(Dense(6, activation = softmax)$

The improved model produces enhanced outcomes since the number of dropout layer, which is a randomized factor for training the model, is decreased.

3.1. Improved Results

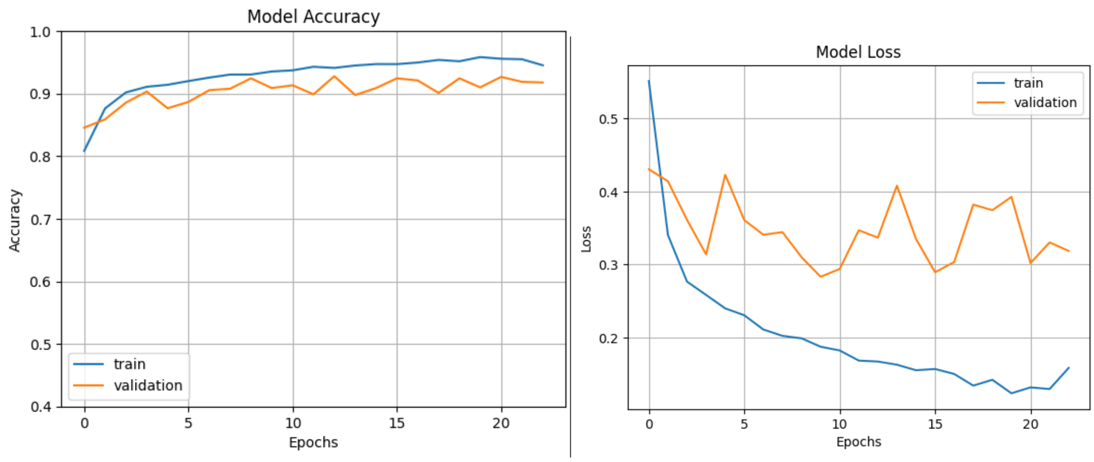

Figure 3.1: Model Accuracy and Loss of Improved RGB Dataset

Train Accuracy: 93.76% Validation Accuracy: 90.88%

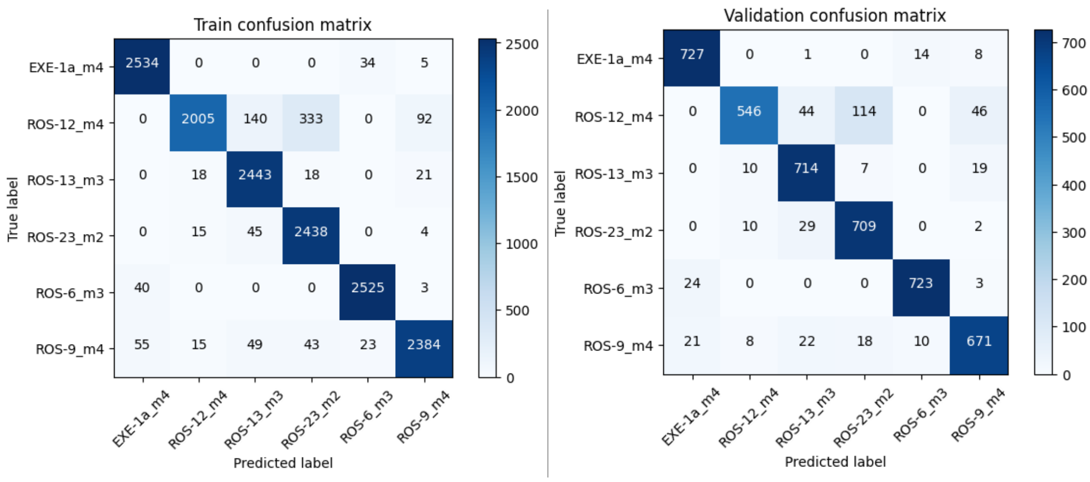

Figure 3.2: Train and Validation Confusion Matrices of Improved RGB Dataset

4. CONCLUSION AND DISCUSSION

4.1. Conclusion

Expected results for the end of the project were determined as:

1. Accuracy goal $\geq 80\%$ with ResNet with independent channels
2. Accuracy goal $\geq 80\%$ with ResNet by using 3 channels together
3. The Response Time Goal $\leq 2second$

The success criterias mentioned above are fulfilled with extra VGG16 studies and graphical user interface design.

The results obtained show that:

- The dataset is not adequate for VGG16, thus underfitting problem occurs.
- The processed dataset increased the accuracy by 2% more than Ch4 and 10% more than Ch6 and provided improved results for both VGG and ResNet models.
- ResNet-50 model produces better outcomes than VGG16 model.
- The ResNet-50 model explained in the Chapter 3 will be presented in the demonstration.

4.2. Discussion

The models are pre-trained with imagenet dataset. However the dataset in this project is consist of small,unique and rare cell images. Therefore, better results could have been obtained if this project had been executed with a model that had pre-trained with cell images.[1]–[6]

BIBLIOGRAPHY

[1] L. Ma, "Combining dc-gan with resnet for blood cell image classification," 2020. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/32221797/.

[2] T. Mahmood. "Artificial intelligence-based mitosis detection in breast cancer histopathology images using faster r-cnn and deep cnns." (2020), [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7141212/.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015. [Online]. Available: https://arxiv.org/abs/1512.03385.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014. [Online]. Available: https://arxiv.org/abs/1409.1556.

[5] C. Li Mahjoubfar, "Deep cytometry: Deep learning with real-time inference in cell sorting and flow cytometry," 2019. [Online]. Available: https://www.nature.com/articles/s41598-019-47193-6#citeas.

[6] M. Doan Barnes, "Deepometry, a framework for applying supervised and weakly supervised deep learning to imaging cytometry.," 2021. [Online]. Available: https://www.nature.com/articles/s41596-021-00549-7#citeas.