Norwegian University
of Life Sciences

**Master's Thesis 2023 30 ECTS**
Faculty of Science and Technology

# Exploring the potential of deep learning models for fish classification

Sanam Maharjan
Master of Science in Data Science

This page is intentionally left blank.

# Acknowledgements

I would like to thank and express my deepest gratitude to everyone who has contributed to the completion of my master's thesis.

First and foremost, I would like to express my sincere appreciation to my supervisor Habib Ullah for continuously supervising my Master's thesis and helping me learn, guiding me through the confusions and supporting my work.

I would also take this opportunity to thank Fadi al Machot for his co-supervision and guidance. His presence, support and suggestions during the meetings and discussions helped me throughout the completion of this thesis.

I am also deeply grateful to my friends, Rinju Manandhar, Shailaja Thapa and Nissan Karki for helping and inspiring me at the time of confusion and challenges. I am sincerely thankful to them for their support and suggestions.

Finally, I would also like to thank my family who always believed in my ability and interest towards attempting new achievements.

Sanam Maharjan

Oslo, May 15th 2023

# Abstract

Traditional techniques in aquaculture are time-consuming, laborious, has limited accuracy and are expensive. They are being replaced by modern day technologies and methods like IoT and artificial intelligence. Scientific techniques are more efficient, cost-effective, autonomous and accurate in the field of aquaculture. Fish classification is one of the popular domains where much research is being done. It is being made autonomous and more accurate by use of deep learning methods. Fish classification has a vital role in aquaculture to make it more sustainable in case of production and maintain a proper monitoring system.

In this thesis, we have studied and proposed a Vision transformer (ViT) deep learning model to classify fish images. We have explored the performances of proposed ViT model along with other two state-of-the-art performing deep learning architectures like VGG16 and Inception V3. The mentioned three deep learning models were trained on three different publicly available datasets. Additionally, the three architectures were used when pre-trained on large-scale image database (ImageNet) and without pre-training too. The pre-trained and non-pretrained frameworks were additionally trained on the three image datasets by applying seven different augmentation techniques and also without any augmentation. Their performances were studied and compared to evaluate which framework gives better results.

In all the alternate conditions applied, the Vision transformer showed stable results with high performance values. VGG16 and Inception V3 also demonstrated promising results but under the varying conditions applied ViT is steadier and reliable. We used accuracy as performance metrics for all three frameworks.

***Keywords***: Fish Classification, Smart aquaculture, VGG16, Inception V3, Vision Transformer, deep learning

# Table of Contents

# List of figures

# List of tables

# Abbreviations

| Abbreviation | Meaning |
|---|---|
| FAO | Food and Agricultural Organization |
| LIFDC | Low-income food-deficit countries |
| IoT | Internet of Things |
| SVM | Support Vector Machines |
| RF | Random Forest |
| VGG | Visual Geometry Group |
| CNN | Convolutional Neural Networks |
| ViT | Vision Transformer |
| RNN | Recurrent Neural Network |
| SIFT | Scale Invariant Feature Transform |
| YOLO | You only look once |
| IoU | Intersection of Union |
| RPN | Regional Proposal Network |
| RCNN | Region-based Convolutional Neural Network |
| RESNET | Residual Network |
| DL | Deep Learning |
| AI | Artificial Intelligence |
| ML | Machine Learning |
| RBM | Restricted Bolzmann Machine |
| RGB | Red, Green and Blue |

# 1 Introduction

## 1.1 Fish facts

Fish and seafood products have been one of the important parts of the human diet. Nutritionists recommend consuming fish products about two times a week, since they are valuable in terms of nutrition to humans [4]. Recent and past studies show that fish are rich in fatty acids, proteins, peptides, Vitamins, selenium, calcium and amino acids that are well known for positive health effects when consumed [5]. Fish products being rich in nutrition and beneficial to human health, the demand for such products has been growing intensely. We have described the nutritional facts from fish and its products in Figure 1.

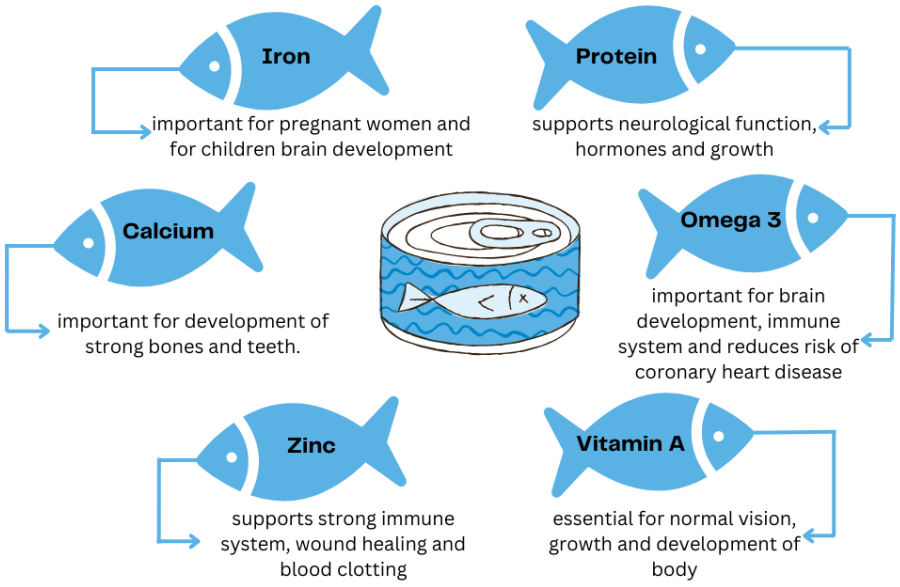


*Figure 1: Nutrients available in fish (products).*

Aquaculture has been one of the major sources of fish and raw materials for various sea-foods. As per a recent study done by FAO, about 178 million tons of aquatic animals are produced globally from which 157 million tons are consumed by humans, that is about 89% of the total productions [6]. This shows that the human population mostly feeds on aquatic products for their diet. In a world fisheries review, FAO has stated that fish covers about 20 percent or likely higher amount of animal protein consumption in low-income food-deficit countries (LIFDCs) [7]. Fish being affordable and beneficial, its consumption will likely continue to increase with time.

## 1.2 Aquaculture in general

Aquaculture has been efficient all around the world to produce and supply such huge needs. In order to keep it efficient, the aquaculture industry will also be required to adopt modern methodology and technologies. Aquaculture is mostly practiced by countries with large water bodies such as Norway, Denmark, China, USA, India, Bangladesh and so on [8]. Traditional methods in aquaculture will not be able to meet the required production level as they are mostly manual and require a longer time period to get results for a task. We can take the examples of tasks like detection, segmentation and classification of fish in a farm which are still not fully automated. Performing these tasks manually can take long, but introducing present day technologies like deep learning methods can make those tasks automated and be completed much faster with great accuracy and results [9].

Aquaculture can be generalized as a combined industry which is dependent upon water resources and the aquatic animals. It represents the organized nurturing, feeding, controlling, protecting and analyzing aquatic resources and animals for commercial or research purposes [10].

## 1.3 Smart Aquaculture

Smart aquaculture can be referred to as the implementation of present day technologies such as machine learning and cloud computing in the field of a controlled fish farm [8]. Like any other evolving industries, aquaculture is also leading towards a new standard of production with application of modern development methods and ideas. Smart Aquaculture includes integration of various smart devices in the aquatic environment which allows it to monitor the environment and also collect data [11]. Integration of smart devices and automatic data collection allows the aquaculture to be controlled and managed by application of IoT, artificial intelligence, cloud computing and robotics [12]. This all leads smart aquaculture towards automation and smart production with greater accuracy and precision. Application of artificial intelligence in aquaculture has been

increasing as it is proven to be capable of solving problems of traditional aquaculture systems [13].

Available data can be considered as one of the major elements to make smart aquaculture applicable. Data can be available in many forms like image, text and audio which might be high dimensional and complicated to handle [8]. These data can be used by deep learning methods that can give the results for specific tasks. There are already numerous artificial intelligence models that are dependent on data and can give out valuable information for smart aquaculture [14].

Traditionally, the tasks like monitoring and controlling of an aquaculture farm was dependent upon expert knowledge, outlines and heuristics. The traditional methods can be time consuming and are continuously exposed to human error. Beside traditional methods including human interactions, many traditional machine learning algorithms like K-means, Support Vector Machines (SVMs), Random Forest (RF) are already applied in the field of aquaculture. However, these methods have not been efficient enough to provide expected outputs as they are not capable of extracting deep features from available data [9]. Deep learning can overcome the weaknesses of traditional machine learning methods and provide significant results. Deep learning methods have significant contributions on different applications like object recognition, classification, speech recognition, object detection and other fields also [15]. The performance and output of these models also depends upon how they are being applied for the given input.

## 1.4 Thesis objective

Detection and classification of fish species in an aquaculture is one of the major tasks. Deep neural networks can be used to perform these tasks with various sizes of datasets to get desired outputs. Fish classification problem in deep learning has been a topic of interest in the field of computer vision and machine learning [16]. Various deep learning networks have been studied and applied for the multi-class problem of fish classification. However, there has not been much exploration of the popular methods being used for such tasks. Many well-known models like ResNet, DenseNet, VGG, RNNs, Inception and others have been applied for fish classification for small to large datasets [9]. These models are well known for classification and detection tasks with high accuracy.

The core purpose of our study is to investigate results of the few popular deep learning models with different conditions and contrasting datasets. We have also studied and applied Vision Transformer (ViT) [3], a recently developed deep learning model in 2021 for fish image classification. Vision Transformer has gained its popularity for outperforming Convolutional Neural Networks (CNNs) [40]. We have studied the results of three different deep learning models including ViT for three different publicly available datasets under various conditions. From our conclusion, we can sum up that ViT performs better in case of fish image classification.

# 2 Related works

Smart aquaculture has been a field of interest for study and application. Many deep learning methods have been applied in this field. Some of the most popular applications of deep learning in smart aquaculture are: Fish identification, fish species classification, fish size estimation, automatic segmentation, and intelligent diagnosis of fish behavior [8]. These applications are primarily focused on fish and its production than other aquatic animals. Besides fish, shrimps and lobsters are other aquatic animals that has been in highlights considering their demands and products. We went through some of the publications that have shown the impact of deep learning in aquaculture relative to fish.

## 2.1 Fish identification

Fish identification is widely used to identify fish under deep water or in their natural habitat. Lee et al. have studied the application of fish identification using contour extraction under a controlled environment [17]. In their study they used a high-resolution underwater color camera to collect the colored images of the fish. The high-resolution camera images allowed them to separate fish from the complex background. They have mostly used image processing techniques like background subtraction, edge detection and contour matching to identify fish. However, their major purpose was to monitor the fish migration patterns.

Moreover, an automated identification of fish was developed with consideration of visual features and using a 32 deep layered convolutional neural network [18]. The proposed method first preprocesses fish images to extract visual features using Scale Invariant Feature Transform (SIFT) algorithm. Based on the extracted visual features, the deep CNN learns to identify fish images. The performance of such deep CNN was compared to other deep learning models like VGG16, ResNet-50, LeNet-5 and GoogleNet [18].

Another real-time fish identification was developed using You only look once (YOLO) deep learning framework [19]. The authors have used a large dataset of fish images to

identify reed fishes in real-time. Unlike other systems, their system could successfully identify reef fishes in real-time with high accuracy. Their achievement was the identification of reef fish in live video feed. Their system can be improvised by including features and data augmentation techniques. Beside these, there are other traditional as well as recent studies being done to identify fish more accurately and feasibly. This can aid smart aquaculture to be more autonomous and advance under various conditions.

## 2.2 Fish Species Classification

Fish species classification is another topic that has been of interest for researchers. Classification of fish species can contribute to understanding aquaculture more and also learn fish behavior in Ichthyology [16]. There have been approaches with and without machine learning to detect and classify fish. In case of approaches without machine learning, L. M. Wolff et al. [20] proposed a fish detection method using sonar imaging. The system was tested and shown to be effective in shallow water environments during a field experiment. Such models can be improved by incorporating it with deep learning algorithms.

CNNs are one of the powerful deep learning frameworks for image classification [21]. Dhruv et al. [16] classified fish species on a fish image dataset using CNN with an accuracy of 96.29%. An additional step of noise removal was performed on the images to get the best results. Similarly, Parnav et al. [21] proposed and compared two CNN based approaches to classify 23 different species of fish. Two versions of VGG16 models were studied and compared: scaled-down VGG16 (VGG8) and traditional VGG16. The VGG8 gave an accuracy of 98.25% and VGG16 gave 96.07%. The author mentions that larger networks can lead to overfitting with smaller datasets. Simpler models like VGG8 can be applied to classify fish under restricted conditions like limited hardware and small datasets [21].

Additionally, there have been other works that focus towards automatic fish classification using other deep learning approaches and integrating other methods with deep learning. AlexNet and VGGNet have been implemented and studied with inclusion of dropout

layers for automatic fish classification [22]. AlexNet has also been combined with a naïve Bayesian layer to enhance classification capability [23].

## 2.3 Intelligent diagnosis of fish behavior

Another important application of deep learning in aquaculture is fish behavior analysis. Regular monitoring, reporting on fish behavior in their habitat can be used to monitor water quality and also to get warnings on fish diseases [9]. Fish behavior diagnosis can also be used to optimize real-time feed control in an aquaculture [24]. A practical method was developed by Zhao et al. [25] by using a modified version of influence map and recurrent neural network (RNN) to monitor local unusual fish behavior in a fish farm environment. Firstly, the fish were detected and tracked by using a motion influence map. Such maps were generated by preprocessing fish images from the farm by application of particle advection scheme. They also made use of the minimum distance matrix framework for localization and detection. At last, through their customized RNN, they successfully localized unusual behavior. They have claimed their method to perform better than other state-of-art frameworks.

An efficient end-to-end CNN was proposed by Iqbal et al. [24] to classify the fish behavior into two categories. They defined two classes for fish behavior: Normal behavior and starvation behavior. The experiment was carried out by use of a differing number of fully connected layers and by including, excluding the max pooling layer. A laboratory-based dataset was used in the study where it contained 100 black scrapers. It was concluded that CNN architecture with max pooling and extra numbers of fully connected layers gave more accurate results. This method can be also used for detection and classification of fish species. However, without well study it cannot be concluded as the best performing architecture.

## 2.4 Automatic Fish Segmentation

Fish segmentation deals with structure or form of fish which includes fish body length, width, eye diameter and other external features. It is one of the critical pieces of information required in smart aquaculture and marine-culture [26]. As the fish demand is increasing, the rate of fishing in open waters have also increased rapidly. In this race of fishing, large number of caught fishes are discarded as they were not the right species or size that fishers were looking for. This all leads towards overfishing and endangering certain fish species [27]. R Gracia et al. [27] have put forward an image-based method for automatic fish segmentation in-order to lower the number of undersized being fished. They have used Mask R-CNN to segment the fish images that were preprocessed. In-order to gain the correct estimation of outline for each fish the segmented images are refined by applying local gradients. Mask R-CNN is also seen to be useful to determine the boundaries of fish in overlapping fish cases. The segmentation performance in their work is measured by using Intersection of Union (IoU) and pixel accuracy metrics. Similarly, Yu, Chuang, et al. [26] have also implemented Mask R-CNN for fish segmentation and extraction of morphological features of fish in fish images. They have conducted the study with images having varying backgrounds.

Alshdaifat et al. [28] have put forward an improved deep learning framework for fish segmentation, stating the issue with other ideas that used static images instead for segmenting fish in their natural habitat or under water. They have overcome the issues with underwater videos like presence of noise, other underwater creatures and bad lighting environments. Their work is divided into four major steps which includes pre-processing of videos to enhance detection, use of RESNET deep learning model to increase the detection, application of Region Proposal Network (RPN) framework to detect multiple fish in the video and lastly, use of dynamic instance segmentation [28]. While comparing their model performance with other highly used models like Mask R-CNN, LACT, and CASCADE R-CNN, their model showed better performance with higher accuracy rate.

## 2.5 Fish size estimation and counting

In-order to maintain a required number of stock and profit in the fish farm sector, the farm needs to have a proper way of counting and estimating the number and size of the fish present. The traditional methods of doing so are manual, time consuming and expensive [29]. Recently, Petrellis, N. [30] provided a method for estimating fish's length, height and area by combining edge detection and pattern stretching methods. He took four fish species in account for the study and application of the proposed method where the morphological features were extracted using image processing techniques. After feature extraction the segmentation was carried out using Mask R-CNN. Since neither Mask R-CNN nor Mask R-CNN with GrabCut was able to detect the counter, he used an implementation of OpenCV with GrabCut [31]. He then used annotated landmarks to calculate the fish length and height. Similarly, Álvarez-Ellacuría, Amaya, et al. [29] have also implemented Mask R-CNN for the estimation of European hake length without any image preprocessing techniques.

Deep learning methods have also been implemented in counting fish which has made the job autonomous and less costly. CNN has been applied by French et al. [32] to count and monitor the fish from available video data. They have used N4-fields image transformation method for foreground segmentation to distinct fish from its background and foreground pixels. A CNN based regressor was used to map input patches and output patches we added back to output images. The output images were scaled base on the mean of corresponding pixels. This method is proposed for real time counting and even for high resolution video streams.

All above-mentioned categories have been well studied and various methods have been put forward. Although there are frameworks that are claimed to have highest accuracy, a comparative study of such methods with newly introduced algorithms are required. Table 1 gives a general informative summary of the works that were published in the past which is related to our study. They have provided their own applications of deep learning methods in the field of aquaculture to solve various issues. These ideas and works are taken as references for our study to grab ideas and information.

*Table 1: Summary on relative studies done for deep learning with aquaculture.*

| Title | Application | Method used | Accuracy | Dataset details | Published year |
|---|---|---|---|---|---|
| Contour Matching for Fish Species Recognition and Migration Monitoring [17] | Fish identification, fish migration monitoring | TADA (Trace-Augmented Discriminative Analysis) (not deep learning but an introduction to possibilities in advancement), | 73.3 % | Video recording of fish in natural habitat | 2008 |
| Visual features based automated identification of fish species using deep convolutional neural networks [18] | Fish identification | 31 layered deep CNN | 96.63 % | images of 11 species of fish found in the local market. | 2019 |
| Real-time reef fishes identification using deep learning [19] | Fish identification | You only look once (YOLO) | 90.70% | images of 24 species of reef fish | 2020 |
| Imaging sonar-based fish detection in shallow waters [20] | species classification | Sonar-based fish detection | Not mentioned but | data are collected by BlueView imaging sonar system | 2014 |

| | | | | | |
|---|---|---|---|---|---|
| Underwater Fish Species Classification using Convolutional Neural Network and Deep Learning [16] | fish species classification | CNN | 96.29% | Fish4Knowledge dataset was created by researchers at the University of Girona | 2017 |
| Towards Designing the Best Model for Classification of Fish Species using Deep Neural Networks [21] | classification of fish species | VGG16 and VGG8 | 98.25 % for VGG16 and 96.07 % for VGG8 | Fish4Knowledge Project at the University of Edinburgh | 2020 |
| Automatic Fish Species Classification Using Deep Convolutional Neural Networks [22] | Fish species classification | AlexNet | 86.65% | QUT fish dataset | 2021 |
| Naive Bayesian fusion based deep learning networks for multisegmented classification of fishes in aquaculture industries [23] | Fish species classification | AlexNet with Bayesian fusion | 98.64 % | 'Fish-Pak' image dataset | 2021 |
| Modified motion influence map and recurrent neural network-based monitoring of the local | fish detection, localization and recognition | Customized RNN | 98.91%, 91.67% and 89.89% | dataset from Zhejiang University | 2018 |

| | | | | | |
|---|---|---|---|---|---|
| unusual behaviors for fish school in intensive aquaculture [25] | | | | | |
| Intelligent Diagnosis of Fish Behavior Using Deep Learning Method [24] | Fish behavior diagnosis | end-to-end CNN | 98% | dataset from China Agricultural University, Beijing. F | 2022 |
| Automatic segmentation of fish using deep learning with application to fish size measurement [27] | fish detection, localization and recognition | Mask R-CNN | 99.40% | data obtained from cruises in North Atlantic | 2020 |
| Segmentation and measurement scheme for fish morphological features based on Mask R-CNN [26] | Fish segmentation | Mask R-CNN | based on segmentation | samples from Hainan University Marine College Aquaculture Professional Production and Research Base | 2020 |
| Improved deep learning framework for fish segmentation in underwater videos [28] | Fish segmentation | RESNET 3 | 95.16 | Fish4Knowledge dataset | 2020 |
| Measurement of Fish Morphological Features through Image | Fish size estimation | Mask R-CNN, VGG16 | various accuracies as per the fish height, | dataset of four Mediterranean fish species | 2021 |

| | | | | | |
|---|---|---|---|---|---|
| Processing and Deep Learning Techniques [30] | | | length and area | | |

# 3 Theoretical background

This thesis involves terminologies mostly related to deep learning frameworks. This section consists of general explanation of terms like deep learning, image augmentation and transfer learning with their working mechanisms.

## 3.1 Deep learning

Deep learning is an extension of a machine learning algorithm that is based on an artificial neural network [8]. It provides learning ability to machines through a series of algorithms which enables them to perform various tasks. Deep learning and machine learning were built to make tasks autonomous and more machine dependent. The most distinct characteristic of deep learning is, it uses multiple levels of representation for learning. It also enables deep learning to extract features from input and represent them in higher form of representation [9]. Deep learning (DL) can be viewed as a derivation from the concept of artificial intelligence (AI) and machine learning (ML). They are similar in a way since they all are used to provide intellectual characteristics to a machine through various ways.

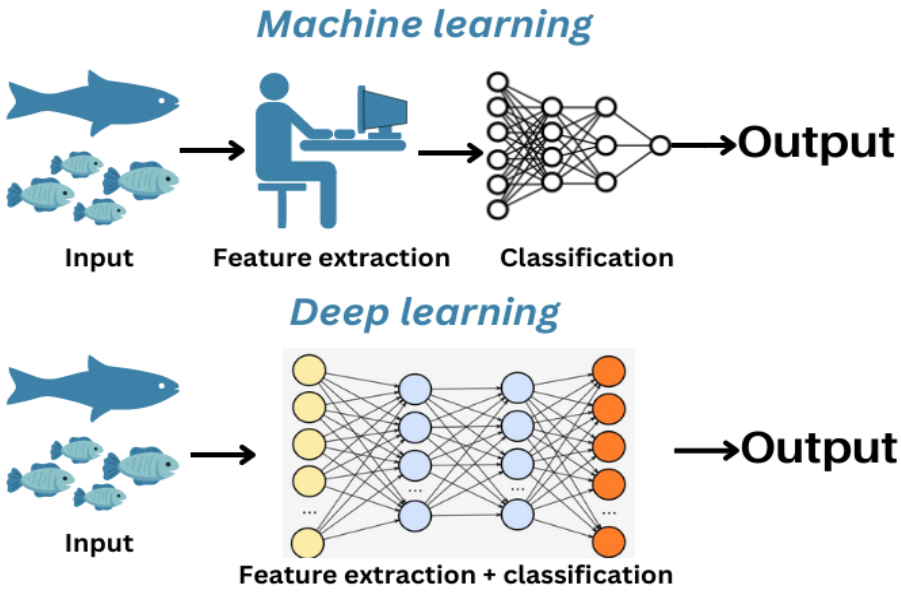


*Figure 2: Contrast between machine learning and deep learning.*

Both DL and ML are subspace of AI but they are different in case of techniques and scope. As shown in Figure 2, ML has more of a manual task like solving each task by dividing it into smaller pieces and adding them at the end to get output or a solution. This trait of ML makes it slow and costly for applying it into real world large problems [33]. However, DL on the other hand overcomes this issue and is much more flexible. It has complicated neural network layers that increases its computational power and enables it to handle complex problems of the real world. Traditional ML lacks the feature to extract essential features of data which is fulfilled by DL by automatically extracting those important features [9].

DL is further categorized into supervised and unsupervised learning. Supervised learning in DL uses labeled data for learning and to map the input to output. Whereas in unsupervised learning, it deals with data that have no labels and finds patterns without any supervision [8]. CNNs and recurrent neural networks (RNN) are popular supervised deep learning frameworks. Generative adversarial network (GAN), Restricted Boltzmann machine (RBM) and Autoencoders (AE) are few unsupervised DL models. RNN and CNN are widely used in the field of machine vision which enables machines to detect, segment and classify images or videos [8]. They are also popular in the field of aquaculture. We have applied three different frameworks to give a conclusive, best performing model by comparing them with different conditions.

## 3.2 Transfer learning

Deep learning is setting a base for most of the artificial intelligent applications. Image classification is among such applications where deep learning is being applied to classify images to a set of possible categories [34]. Classification of images through deep learning can be eased by use of transfer learning. Transfer learning is a method used in a deep learning model where the model uses knowledge which is learned from its previous task to a new task. A model including transfer learning can be referred to as a pre-trained model. Use of such a pre-trained model saves resources, time and improves the efficiency while a new model is being trained [34].

Image classification for a large dataset can be a challenging task, it can be simplified by using models that are already trained with a set of another large image dataset. For this specific study we have used pre-trained models that are already trained with the ImageNet [35] database. We have also compared the performance of such models without transfer learning. The following four general steps are used during transfer learning: [36]

i. A pre-trained model is initialized. The model learns from a larger dataset, like ImageNet in our case.
ii. The final layer of the desired model is set with the number of classes present in the new dataset.
iii. The layers are tuned or changed as per desire.
iv. Model is trained on a new dataset to get the results.

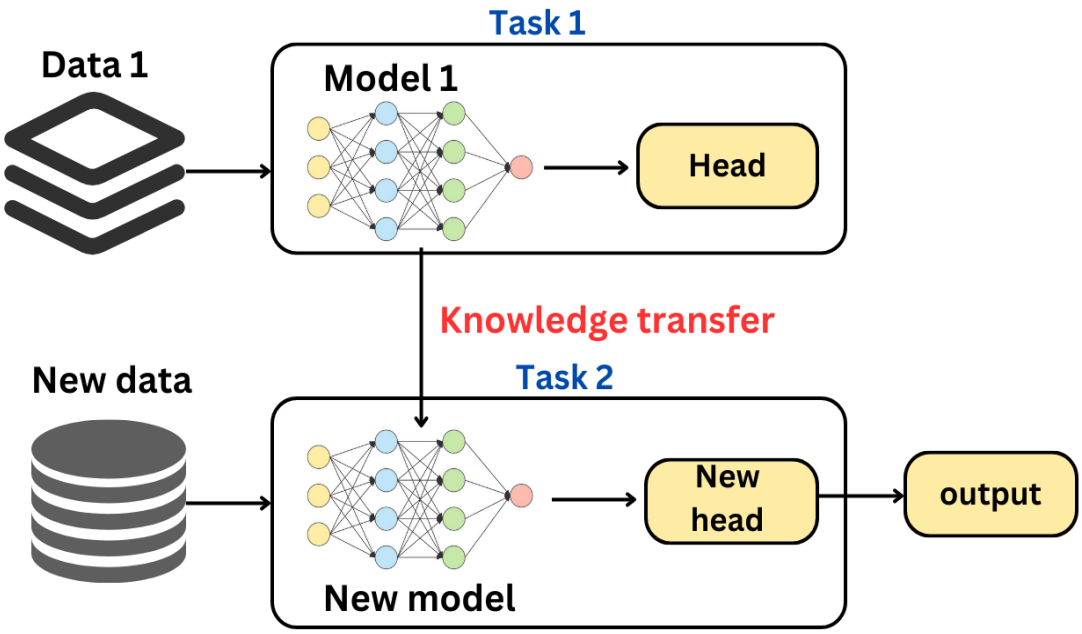


*Figure 3: Transfer learning in a picture.*

Transfer learning comes along with its drawbacks too. One big limitation of transfer learning is, it also transfers the non-relative knowledge which can also be called negative transfer. Negative transfer causes decrease in performance of the new model if the target problems of both models are not quite similar [37]. Overfitting is another highlighted drawback of transfer learning. Overfitting in deep learning is when the model gets a good fit for the seen training data but fails to fit on new, unseen data. In transfer learning,

overfitting occurs when the pre-trained model learns noises and details from a large dataset and overly fits on the new training data [37].

## 3.3 Image Augmentation

Performance of a deep learning model also depends upon the amount and variety of data that is provided while training it. So, in-order to enhance its performing capacity we can bring alterations in data making it large in number and variations. Augmentation is one of the solutions to bring such alterations in data. Image augmentation deals with one or many alternating processing steps applied to image data. Rotation, shifts, shear and flips are some of the popular image augmentation techniques being used in DL.

APIs like ImageDataGenerator [38] from Keras can be used to generate the augmented images in real-time. We have used the same to generate images using various image augmentation techniques. Figure 4 shows results of the mentioned augmentation methods applied to an input fish image. From this it can be visualized that through this method a single data can be used in various forms. These newly augmented images can be used by the proposed model for pre-training which as a result can enhance the performance. Moreover, there are other data augmentation techniques but the ones we have chosen are most commonly used and bring variations in input images also.

```
train_gen = ImageDataGenrator(
      rotation_range=20,
      width_shift_range=0.1,
      height_shift_range=0.1,
      shear_range=0.1,
      zoom_range=0.2,
      horizontal_flip=True,
      vertical_flip=True)
```

The block of code above shows the use of image augmentation techniques like rotation, width shift, height shift, shear range, horizontal flip and vertical flip through the use of ImageDataGenerator. The given values show the amount of such techniques being applied to the input images.


*Figure 4: Various image augmentation methods. A: Original Image, other images are results of augmentation methods applied to original image. B: with rotation range 20, C: with height shift range 0.1, D: with width shit range 0.1, E: with shear range 0.1, F: with zoom range 0.1, G: with horizontal flip set as and H: with vertical flip set as true.*

# 4 Methodology

As mentioned in thesis objectives, we have used three different deep learning and tried to understand and compare the performances of those models under different conditions using three different publicly available datasets. The three different models are: Vision Transformer (ViT), Visual Geometry Group 16 (VGG16) and Inception V3. This section gives more information about those models and insights to their operations. The results of these models will be explained later in the result section of the thesis.

## 4.1 Vision Transformer

Vision Transformer (ViT) is the implementation of transformer architecture in the domain of computer vision. Transformer architecture is well known for its impressive results in the field of natural language processing through the implementation of attention [3]. Dosovitskiy et al. [3] with aim to classify images with better performance results than the state-of-art model like CNN, applied the transformer architecture directly to images. In short, the model works by splitting the image into patches and a transformer gets the linear embeddings of such patches as input. The patches are handled as tokens, like words are handled in Natural language processing applications [3].

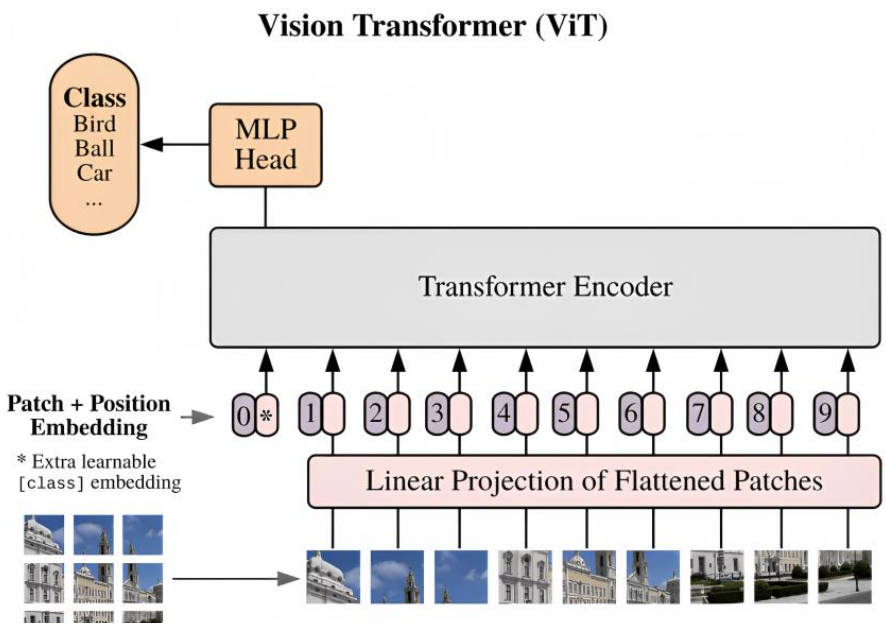


*Figure 5: Architecture of the Vision Transformer (ViT) [3].*

Figure 5 illustrates the steps involved in a ViT proposed by Dosovitskiy et al. Moreover, they have also explained how CNN can be used before the transformer encoder in their model. As per them, CNN feature maps can be used to generate image patches by feeding the original image as input before it passes through the transformer encoder. In order to handle various data sizes, they proposed three different variants of ViT: ViT-Base, ViT-Large and ViT-Huge with 12, 24 and 32 layers respectively. ViT models can also be pre-trained with larger training datasets like ImageNet and also fine-tuned.

In this thesis, we have used the keras implementation of the vision transformer model present in GitHub repository [39]. It reflects the models explained by Dosovitskiy et al. in their paper, *an image is worth 16x16 words: Transformers for Image recognition at scale*. We have used the 32 layered ViT-Huge model from the mentioned github repository. In-order to use the package it was first installed by using the `pip install vit-keras` command. A clear picture of this model being used is shown in Figure 6. It has transformer architecture, self-attention layer and feed-forward layer.

First of all, the input image of height H, width W, and C number of channels is divided into patches. This is done to match the structure of input of the transformer model. This

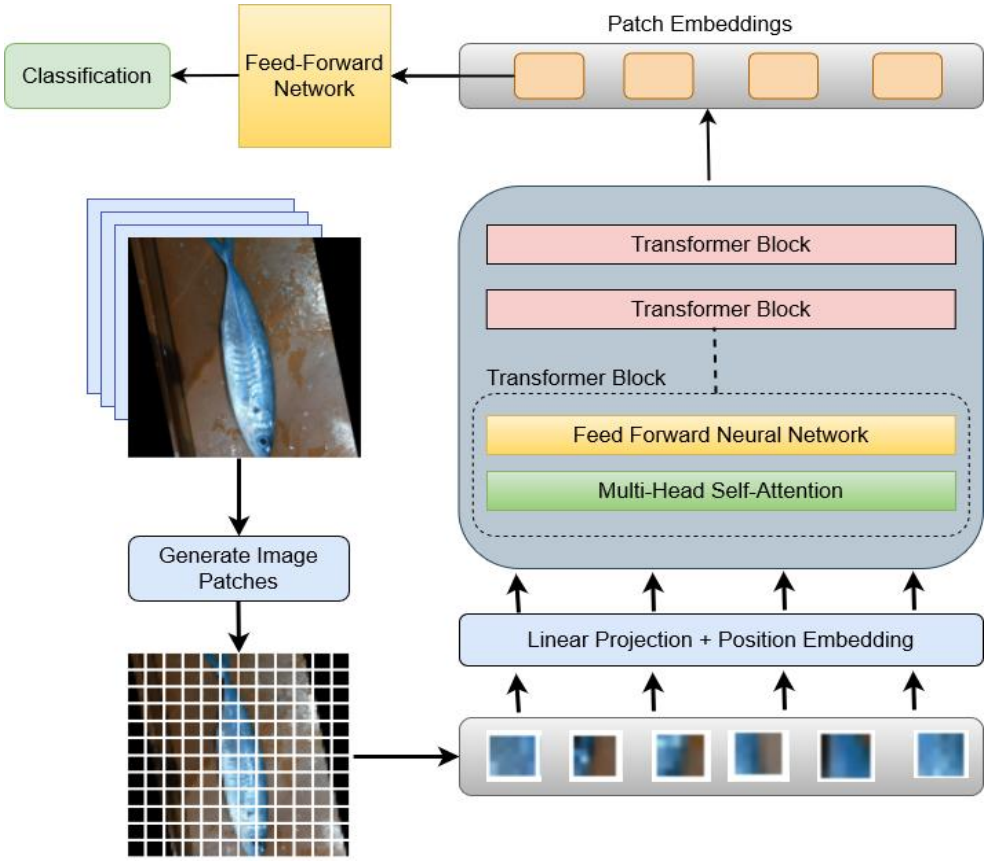


*Figure 6: Architecture of proposed ViT model.*

gives the number of patches N as an output where $N = \frac{HW}{P^2}$ and $(P, P)$ pixels are the resolution of each patch.

A series of operations are applied to the input data before it is fed into the transformer. The operations are [40]:

1. All image patches are flattened into a vector, $X_p^n$ whose length is $P^2 \times C$, where $n = 1, \dots, \dots, N$.
2. The flattened patched with a trainable linear projection, E are mapped to D dimensions and a sequence of embedded patches are generated.
3. $X_{class}$, a learnable class embedding is arranged to sequence image patches.
4. Finally, the patch embeddings are augmented with $E_{pos}$ , one-dimensional positional embeddings.

The output from these operations is the sequence of embedding vectors [40] [3]:

$$z_0 = \left[x_{class} : x_p^1 E; \dots ; x_p^N E\right] + E\_pos$$

The classification is carried out by passing $z_0$ as input to the transform encoder consisting of L layers. Afterwards, the classification head gets the value of $x_{class}$ present at $L^{th}$ layer of encoder output [40].

The main function used to define the model is:

```
model = vit.vit_l32(
    image_size = (224, 224),
    activation = 'softmax',
    pretrained = True,
    include_top = False,
    pretrained_top = False,
    classes = 20)
```

'vit_132' specifies the architecture of ViT being used, here we are using the 32 layered architecture. 'image_size' defines the size of input images to the model, 'activation' is used to specify the activation function as per the case model is being used. 'pretrained' specifies whether the model uses the pre-trained model, 'include_top' defines if the classification layer of the pre-trained model is included or not. 'pretrained_top' describes

whether to include pre-trained weights for the layer and finally ‘classes’ shows the number of classes present in the target dataset.

## 4.2 VGG16

VGG16 is one of the best performing convolutional neural net architectures that performs better than architectures like AlexNet [21]. It is known for its performance on ImageNet competition during 2014. It was developed by K. Simonyan and A. Zisserman with an accuracy score of 92.7% on ImageNet dataset and had more capability with improvement [1]. VGG16 is a deep convolutional network with 16 layers in which the input layer is 224x224 RGB having convolutional layers of 3x3 filter and stride1. They have the same max-pooling layer of 2x2 filter and stride 2 where the input images are reduced. VGG16 has contributed for image classification tasks with positive improvement in CNN [1]. Figure 7 shows an overview of VGG16 architecture.

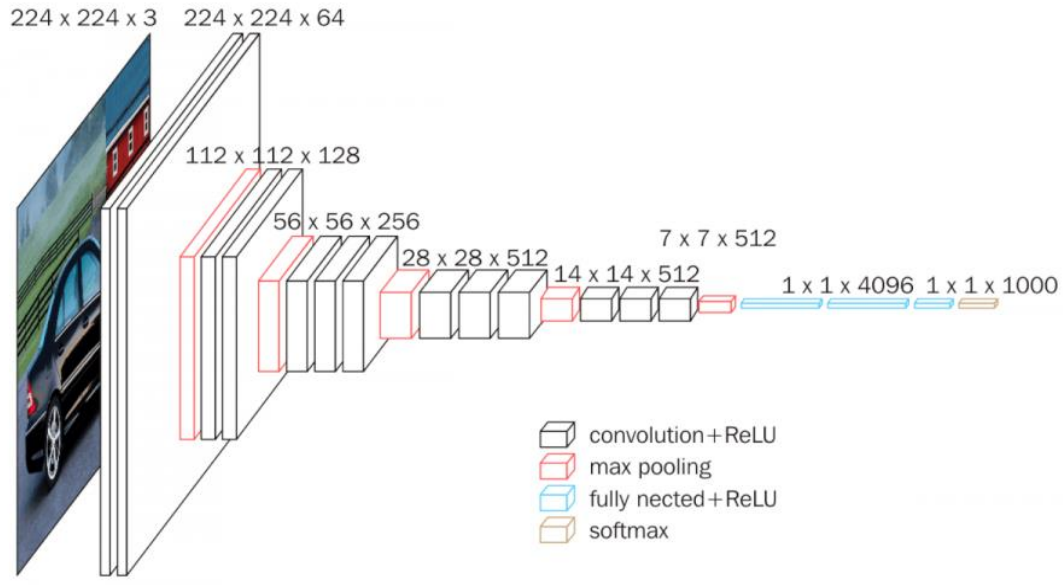


*Figure 7: VGG16 model architecture [1].*

In this study, we have used the keras implementation of VGG16. We have used the imagenet weights for transfer learning and for without transfer learning, we have built the VGG16 model from scratch.

## 4.3 Inception V3

After development of deep CNNs like VGG16, Szegedy, Christian, et al. [2] proposed a modified inception architecture that is deep CNN and focused on less use of computational power. It is primarily used for image analyses and object detection. Figure 8 shows a diagram of inception v3 architecture with the layers involved.

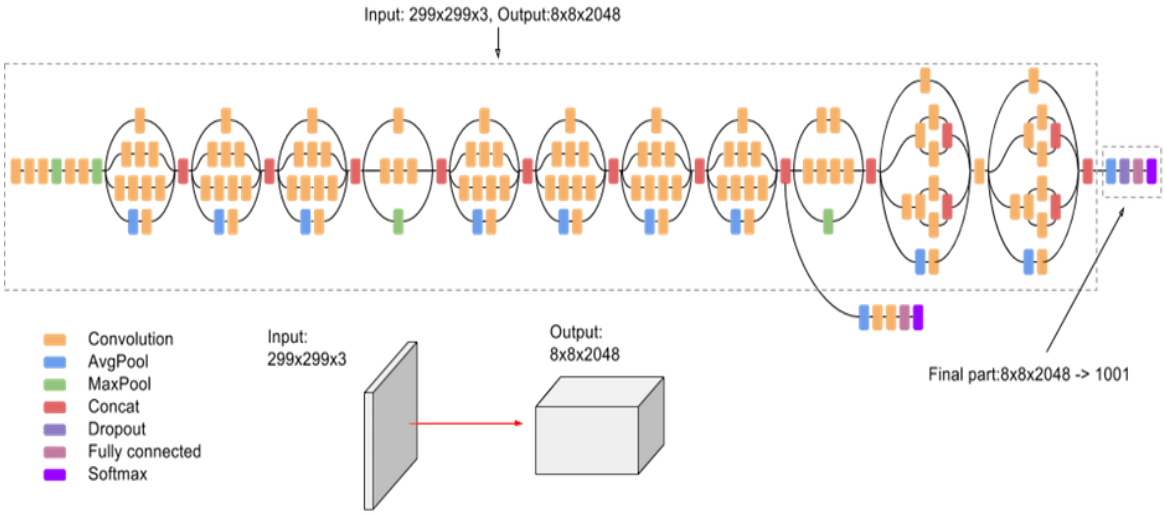


*Figure 8: Inception V3 architecture with its layers [2].*

The input for the model is of size 229x229 RGB, which is preprocessed by subtracting the mean RGB value from the training set from each pixel. The first convolutional and pooling layers are responsible for extraction of low-level features from the input image. The inception architecture used in inception v3 are concatenations of various blocks of convolutional layers that differ in filter sizes. Such combinations of convolutional layers contribute the model to learn features at inconsistent scales and resolutions. It is also the main reason for the lower number of parameters in this architecture. Auxiliary classifiers are also used by the models to aid the training process. These classifiers empower model to learn important features and outline the output beforehand. Output layer of the model consists of a softmax layer that gives the probability distribution of different classes.

We have used the keras implementation of Inception V3 proposed by Szegedy, Christian, et al. in their paper *Rethinking the Inception architecture for computer vision* [2]. Like ViT and VGG16 we have used weights of imagenet for transfer learning.

# 5 Experimental results

Most of the model and analysis were done on a personal computer with an Intel(R) core (TM) i5-8265U CPU @ 1.60GHz 1.80GHz and 8.00GB of RAM. The PC has windows 10 as base operating system and google colab was used. However, for deep convolutional networks like Inception v3 without pre-trained weights, Orion server provided by the university was used. Inception V3 without transfer learning took a longer time period to train, hence Orion was used. In this chapter of the thesis, we will discuss datasets and results of various models with conditions that we tried.

## 5.1 Dataset description

The datasets used in this study were publicly available in Kaggle, a public online platform for data scientists. We have selected three fish image datasets that have different species of fish images.

### 5.1.1 Fish species dataset

The first image dataset was published by Giannis Georgiou [41] in 2020 titled fish species. The fish species dataset consists of images of 20 different Mediterranean fish species. It contains train and test set each having 34,000 and 6,000 images respectively. There are 1,700 images of each species in the train test whereas, in the test set there are 300 images per species. This was a big number of images for us so we decided to further select 100 images from each species in the train set and 20 images from each species in the test set. So, for training our models we had 2,000 images for the train and 400 images for the test. The sub division of images into desired numbers was done by simply creating a function that randomly selects the defined number of images into the sub folder that was defined in the function itself. The species of fish present in dataset are: *Solea solea, Pseudocaranx dentex, Polyprion americanus, Chlorophthalmus agassizi, Rhinobatos cemiculus, Coris julis, Gobius niger, Squalus acanthias, Mugil cephalus, Tetrapturus*

*belone, Trachinus draco, Anthias anthias, Atherinomorus lacunosus, Boops boops, Trigloporus lastoviza, Belone belone, Phycis phycis, Dasyatis centroura, Epinephelus caninus* and *Scomber japonicus*. The images had varying backgrounds and conditions while taken. Figure 9 shows the sample images of each species present in the dataset.

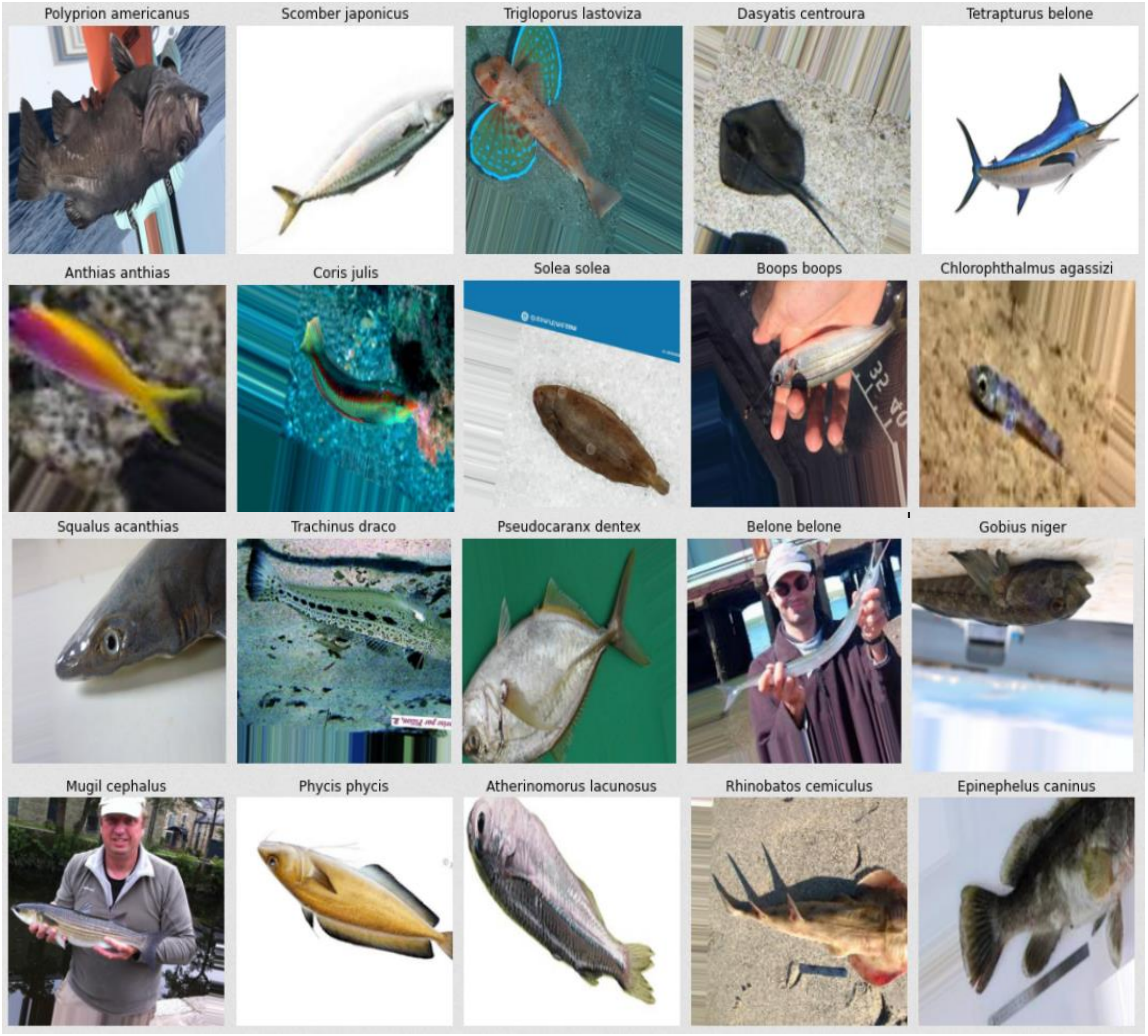


*Figure 9: Random images from each class of fish species image dataset.*

## 5.1.2 Fish Dataset

The second image dataset is also taken from Kaggle. It was published by Mark Daniet Lampa et al. [42] that had images of 31 fish species present in Marinig fishing port in Cabuyao city. The dataset has a train, test and validation set which contains a different number of images of each species. There are a total 13,304 images in which train, test and validation has 8791, 2751 and 1760 numbers respectively. For this dataset we didn't

do further selection as the number was fairly enough and our aim was to study the performance of models in various dataset sizes too. The species of fishes included in this dataset are: *Bangus, Big Head Carp, Black Spotted Barb, Catfish, Climbing Perch, Fourfinger Threadfin, Freshwater Eel, Glass Perchlet, Goby, Gold Fish, Gourami, Grass Carp, Green Spotted Puffer, Indian Carp, Indo-Pacific Tarpon, Jaguar Gapote, Janitor Fish, Knifefis', Long-Snouted Pipefish, Mosquito Fish, Mudfish, Mullet, Pangasius, Perch, Scat Fish, Silver Barb, Silver Carp, Silver Perch, Snakehead, Tenpounder and Tilapia*. Figure 11 shows the images of each species present in the train set of the fish dataset. From there we can see that the fish dataset images also have varying backgrounds with noises too.

### 5.1.3 A large-scale fish dataset

Our final image dataset is also taken from the Kaggle platform which was published by Ulucan O, Karakaya D, Turkan M. [43] in 2020. This dataset consists of images of 9 different species of fish including shrimp. The previous datasets had a larger number of classes for fish whereas this has comparatively less. The images are taken from a fish market in Turkey using Kodak Easyshare Z650 and Samsung ST60 cameras [43]. The

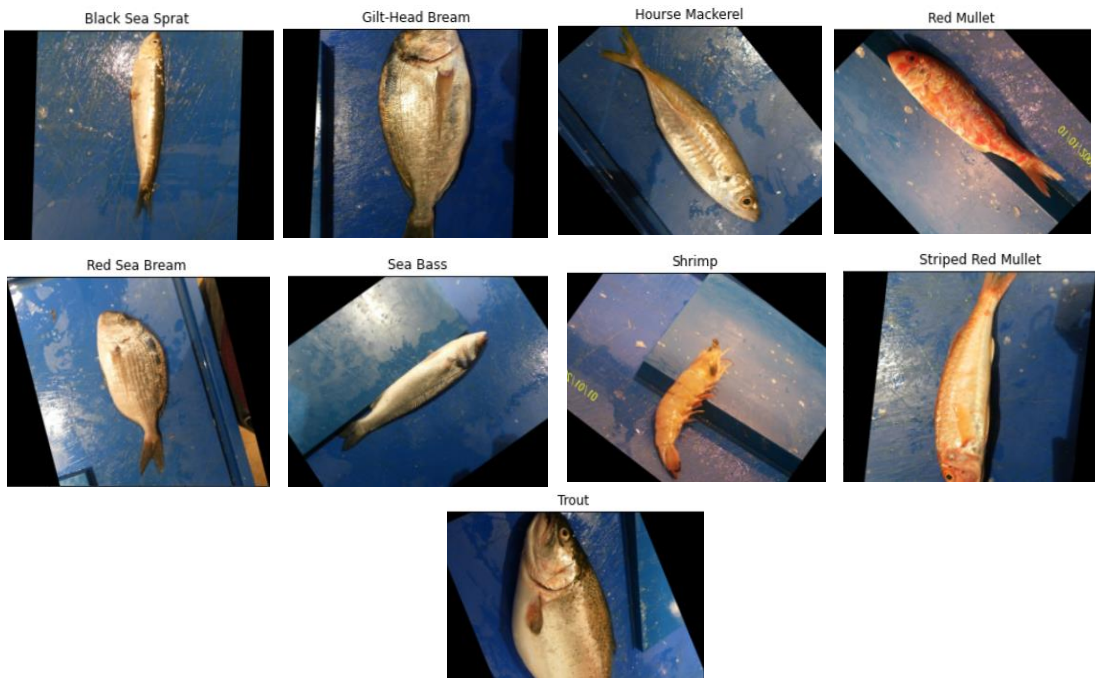


*Figure 10: Images of each species of fish present in the large-scale fish image dataset.*

large-scale fish dataset has two sets of images for each class, one with RGB images and another for their pairwise ground truth labels. We have used only RGB images in which each class has 1000 images.

So, we have a total of 9,000 images which are divided into train, test and validation sets by using train_test_split from sklearn. After the train test split, we have 6,349 images for train, 1,350 images for test and 1,301 images for validation. Figure 10 shows the images of each class of fish present in the dataset where we have images of *trout, red mullet, hourse mackerel, sea bass, glit head bream, shrimp, red sea bream, black sea sprat and stripped red mullet.*

*Table 2: Summary of datasets used in the thesis.*

| Dataset | Number of classes | Total number of images | Train size | Test size | Validation size | Source |
|---|---|---|---|---|---|---|
| **Fish species** | 20 | 40,000 | 2,000 | 400 | - | Giannis Georgiou [41] |
| **Fish Dataset** | 31 | 13,304 | 8,791 | 2,751 | 1,760 | Mark Daniet Lampa et al. [42] |
| **Large-scale fish image** | 9 | 9,000 | 6,349 | 1,350 | 1,301 | Ulucan O et al. [43] |

Table 2 shows the summary of all three datasets that were used in this study. We can see that the datasets show variations in size, number of classes and sets of images present. Use of such a diverse dataset helps to understand more about the performance of deep learning models. We will further discuss the results of three different deep learning models on these datasets.

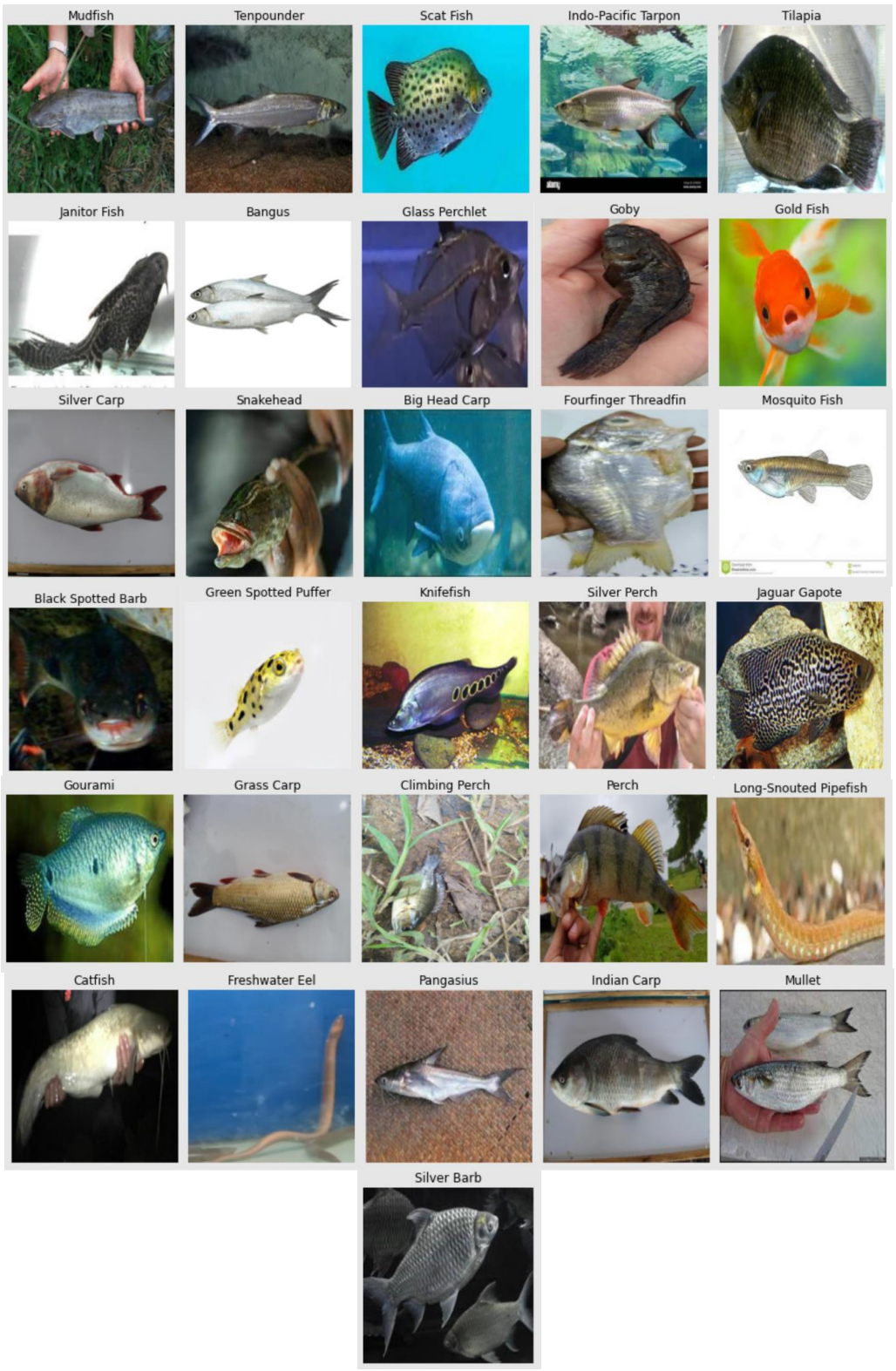


*Figure 11: Random images from each species present in fish dataset.*

## 5.2 ViT with varying conditions

After collection and overview of all the datasets, we used the ViT model with all the varying conditions like: with and without transfer learning as explained on section 3.2, training the models further with and without data augmentation like explained on section 3.3. In this section, we will go through the results obtained with such conditions. We have used accuracy and loss as our performance metrics. We have also included plots and tables to compare and study the influence of transfer learning and augmentation.

### 5.2.1 ViT with transfer learning but with and without augmentation

Vision transformer (ViT) like described in section 4.1 is known for its excellent performance in image classification tasks. In this section, we will discuss performance of the ViT model with and without transfer learning using both augmented and non-augmented image datasets. We took three different image datasets mentioned in section 5.1 as input for the pre-trained ViT model. The input images went through all the augmentation processes as declared in section 3.3 and for without augmented dataset images were used as inputs with no alternations.

#### 5.2.1.1 On fish species dataset

ViT was successful to classify the images with high accuracy on the fish species dataset. For this dataset with augmentation it gave test accuracy of 96.75% and for without augmentation, it gave 94.113%. Figure 12 shows comparative plots of loss and accuracy of pre-trained ViT models when using augmented images and non-augmented images. For augmented images we can see that the model learns quickly in the first few epochs, approximately upto $10^{th}$ epoch but later the model starts to memorize the data. The loss drops throughout the epochs and the accuracy rises fast during the beginning of a few epochs.

Whereas, for pre-trained ViT models without image augmentation, although the plot seems to be similar, we can see that the model overfits much more quickly than the other

model. The model performs not as well as compared with when the images were augmented. We can say that a pre-trained ViT model is overfitted more quickly on not augmented datasets than on augmented datasets given the property of the dataset is less in size and large in the number of classes present.

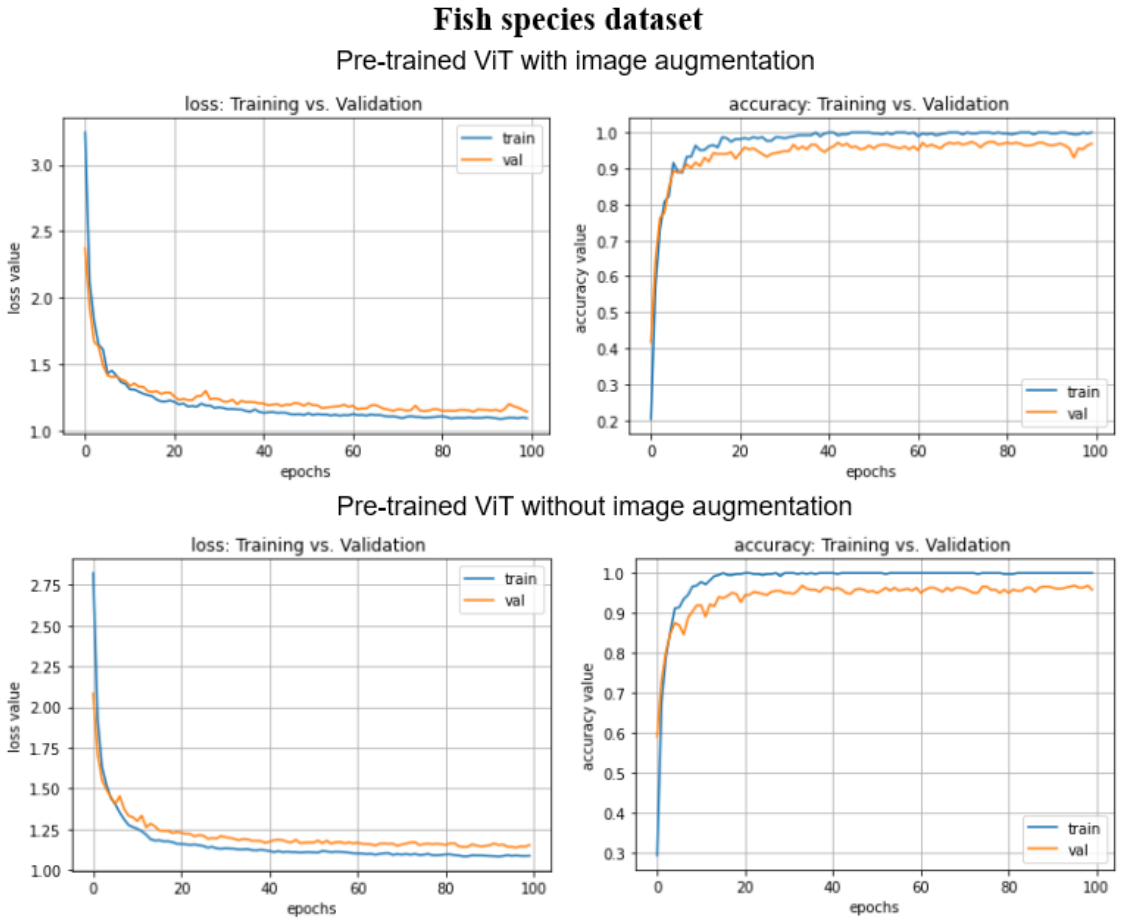


*Figure 12: Comparative plots for training vs validation loss (left side) and accuracy (right side) of pre-trained ViT model with augmentation (top twos) and without augmentation (bottom twos).*

### 5.2.1.2 On the fish dataset

ViT is known for performing well in large size datasets. Fish dataset can be considered as the largest dataset (in case of size and number of classes present) we have for this study and ViT gives a good performance result on this specific dataset. Pre-trained ViT model with augmented images as input gives test accuracy of 97.443% and for not augmented images as input it gives 98.75%.

We can compare the performance of pre-trained ViT model from Figure 13, where we can see that it is somehow similar to each other. The model does not seem to over fit at all but we can notice that it executes smoothly on augmented image dataset than on without augmentation. However, pre-trained model performs similarly for large datasets in both cases.

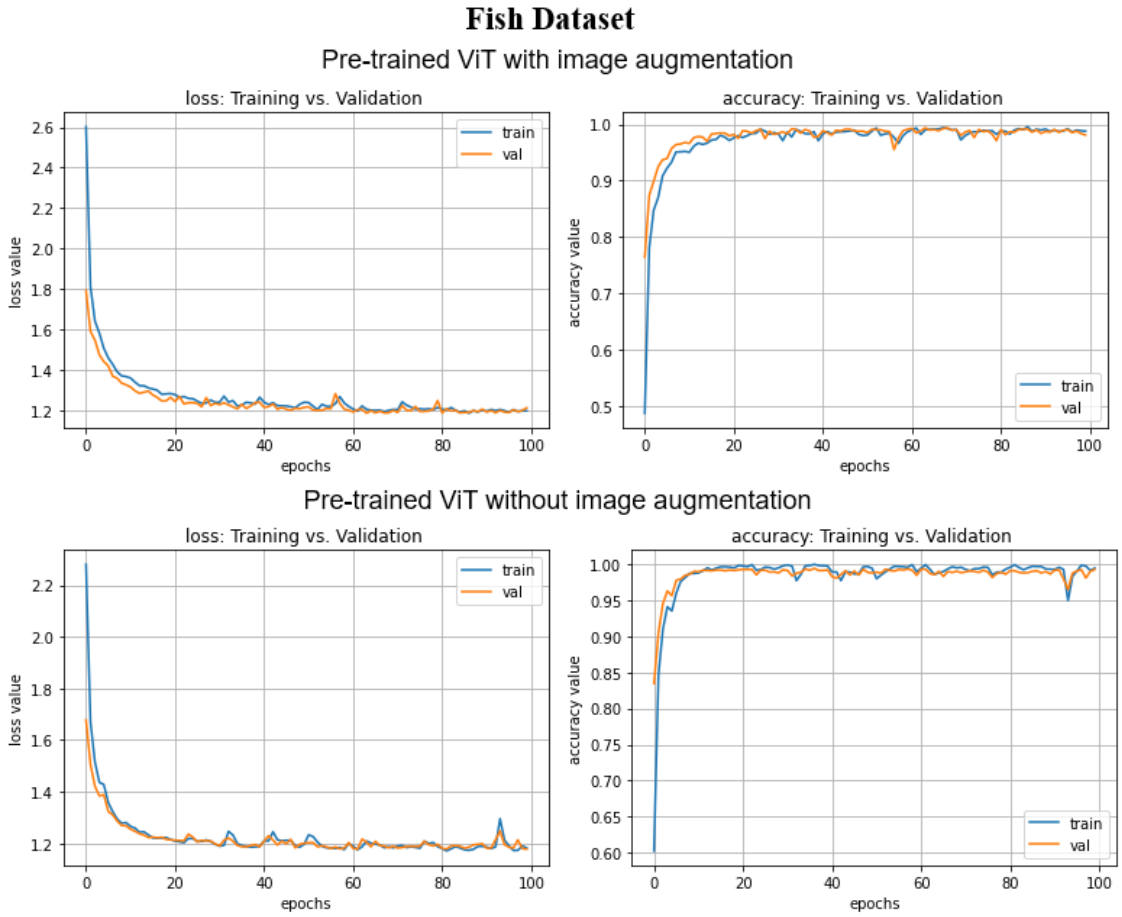


*Figure 13: Comparative plots of training vs validation loss (on the left) and accuracy (on the right) from pre-trained ViT model with (top twos) and without (bottom twos) image augmentation.*

### 5.2.1.3 On A large-scale fish dataset

Large-scale dataset had a simple non-varying background for fish images compared to the other two datasets. It also has a smaller number of classes of fish but higher number of images. Pre-trained ViT models perform well on this dataset as well. It gives test accuracy of 94.133% and 100% on dataset with augmentation and without augmentation respectively. The validation accuracy is a strong 100% for both datasets.

As shown in Figure 14, the performance is similar but we can tell that it performs best with augmented dataset. We can also observe that the model reaches 100% accuracy at around 5$^{th}$ epoch, which is fastest among other two datasets. One of the reasons might be the fewer number of classes to classify with a higher number of data given.

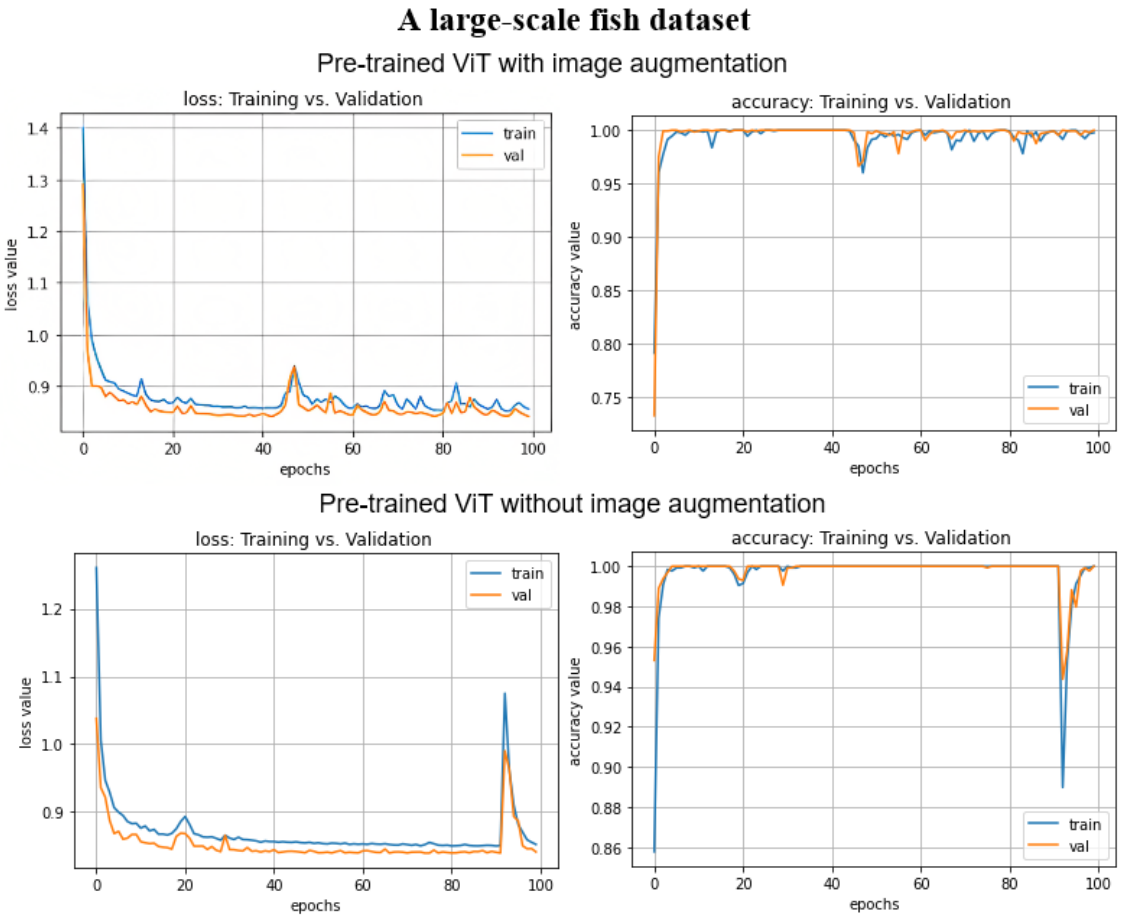


*Figure 14: Comparison of training vs validation loss (on the left) and accuracies (on the right) of pre-trained ViT model on large scale fish image dataset with (top twos) and without (bottom twos) image augmentation.*

The overall performance comparisons of pre-trained ViT models with augmented images and without augmented images is shown in Table 3. We can compare the train, validation and test accuracies and losses of the models on all three datasets. We can see that the pre-trained ViT model has only little difference values for test accuracy on all three datasets whereas, it has high train and validation accuracies on the one with image augmentation than without augmentation. Moreover, all the values are the same for the third dataset without image augmentation.

*Table 3: Summary of performance of pre-trained ViT model with and without augmentation on all three datasets.*

| **Pre-trained Vision Transformer** | | **Fish species** | | **Fish Dataset** | | **Large-scale dataset** | |
|---|---|---|---|---|---|---|---|
| | | Accuracy (%) | Loss | Accuracy (%) | Loss | Accuracy (%) | Loss (%) |
| With Augmentation | Train | 99.9 | 1.073 | 98.339 | 1.199 | 99.905 | 0.824 |
| | Val | 96.75 | 1.147 | 98.11 | 1.211 | 100 | 0.841 |
| | Test | 96.29 | 1.575 | 97.443 | 1.226 | 99.852 | 0.842 |
| Without Augmentation | Train | 100 | 1.068 | 99.886 | 1.162 | 100 | 0.841 |
| | Val | 96 | 1.148 | 99.1 | 1.178 | 100 | 0.841 |
| | Test | 94.133 | 1.987 | 98.75 | 1.193 | 100 | 0.841 |

### 5.2.2 Vit without transfer learning but with and without image augmentation

In this section we will discuss the performance of a non-pretrained ViT model on three different datasets. In a non-pretrained ViT model, like any other deep learning models without transfer learning the weights are initialized randomly. The model is trained from the scratch with the available dataset unlike pre-trained models where the pre-trained weights are used.

#### 5.2.2.1 On fish species dataset

Fish species dataset has a comparatively lower number of images and 20 fish classes. A non-pretrained ViT framework gives test accuracy of 34.02% on augmented dataset and 25.02% on non-augmented dataset. Since the model has no pre-trained weights it can be expected to give lower performance.

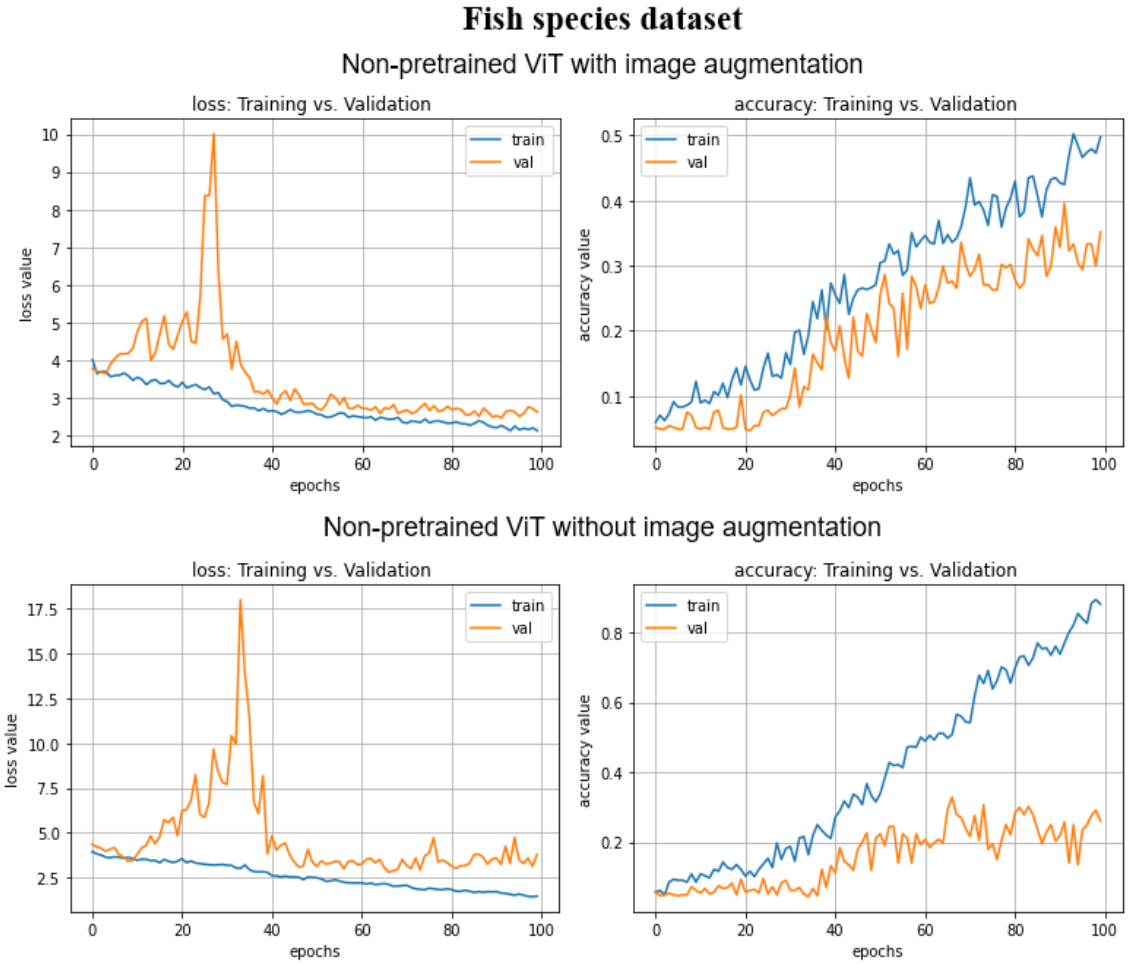


*Figure 15: comparison of training vs. validation losses (on the left) and accuracies (on the right) for a non-pretrained ViT model with augmented data (top twos) and with non-augmented data (bottom twos) on the fish species dataset.*

The performance of non-pretrained ViT can be seen through the training vs. validation loss and accuracy plot as shown in Figure 15. We have trained the model up to 100 epochs from which we can tell that the non-pretrained ViT model overfits more on the one without image augmentation. Since the image data is already small in number, the non-pretrained model without image augmentation starts to memorize the training data as it cannot find other alterations and patterns in the data.

Moreover, the model trained with augmented images also looks like beginning to overfit after the $90^{th}$ epoch. Although from image augmentation we can obtain variations in same the data, with small dataset size the overfitting cannot be avoided.

#### 5.2.2.2 On the fish dataset

Unlike fish species dataset, fish dataset is bigger in number of images and classes. A test accuracy of 56.66% was obtained when training a non-pretrained ViT framework with image augmentation. Similarly, test accuracy of 87.045% was obtained when trained without image augmentation. Comparatively, on a large dataset like the fish species dataset, the non-pretrained model seems to perform better without image augmentation. As per our observation, the reason behind this can be: since the dataset was already diverse with a bigger number of images and classes, addition of augmentation on such dataset added noise to input data. Also, the non-pretrained model learns all the features from raw input images and the augmentation should not have been necessary.

A comparative plot of losses and accuracies for non-pretrained ViT on datasets with and without augmentation is shown in Figure 16. It is observable that the loss on both with and without augmentation increases at first then falls down rapidly at the beginning of the first 10 epochs. Also, on the side of accuracy it can be seen that the model without augmentation is a little bit overfitted than the one with augmentation even though the accuracy score is high. The accuracy of model with augmentation can be seen to be on an increasing trend. We can increase the number of epochs and study more on this but by comparison we can conclude that the actual performance of a non-trained ViT model is better on data with augmentation although the learning process might be somehow slower. Furthermore, a non-trained ViT model tends to overfit without augmented images even though we have a large number of input images belonging to numbers of classes.

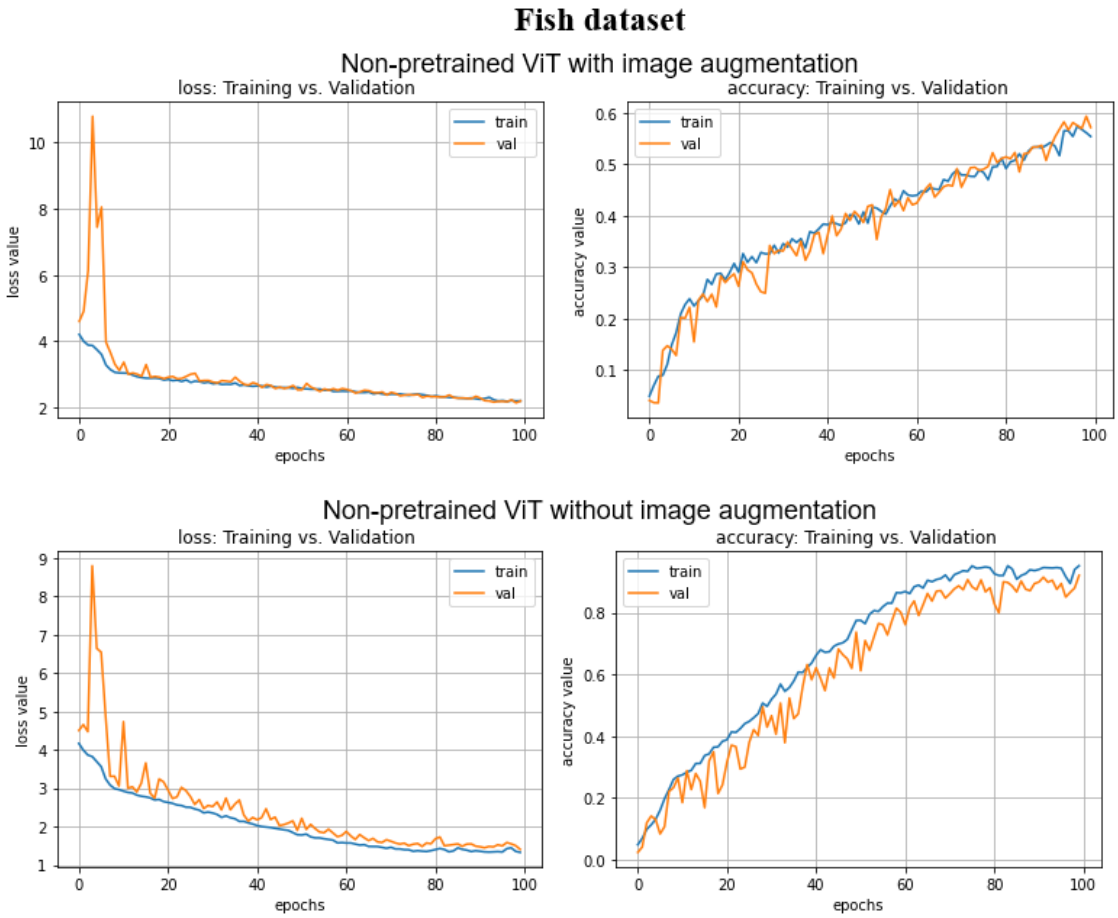


*Figure 16: Losses (on the left) and accuracies (on the right) comparison of non-pretrained ViT model on fish dataset, with and without image augmentation applied.*

### 5.2.2.3 On A large-scaled fish dataset

Performance of a non-pretrained ViT model on a large-scaled fish dataset was high in comparison to other two datasets. We obtained a test accuracy of 92.074% when input images were augmented and a test accuracy of 95.926% when they were not augmented. These results drive to the same conclusion as that of the fish species dataset. We have a small number of classes to classify with varying sets of images. For more detailed study we can look at the comparative plots of losses and accuracies of the model on datasets with and without augmentation as shown in Figure 17. The losses are high at the start of epochs like on other datasets as the model without any pre-trained weights have not learned any features. But the loss gradually climbs down and on the other side we can see accuracies build up gently too.

In the case of a large scaled fish dataset, the model is not overfitted on both conditions. It performs better on the dataset without augmentation as the images were already variant and when the images were augmented it made them noisier hence affecting the performance of the model. From this we can see that a non-pretrained ViT model performance can be affected by the type of input data and conditions. Augmentation mainly helps the model to learn more features gradually. As it is not pre-trained the execution time is surely affected, it takes longer to train the model than using pre-trained weights.

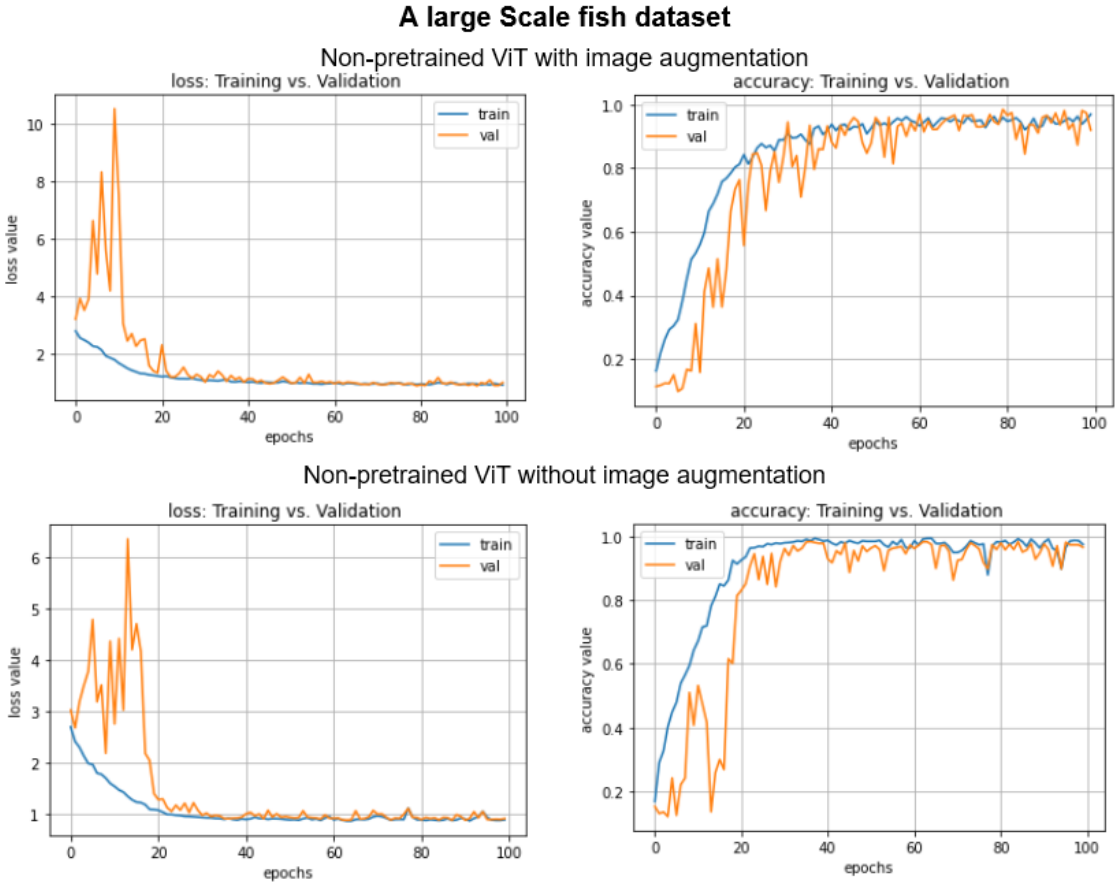


*Figure 17: comparative plots of losses (on the left) and accuracies (on the right) for non-pretrained ViT model with augmentation (top twos) and without augmentation (bottom twos) on a large-scaled fish dataset.*

Moreover, a wholesome comparison can be done on the performance of non-pretrained ViT models on all three datasets too. Table 4 shows all the train, valid and test losses and accuracies of the model on all three datasets, with and without augmentation. The results were gathered after training all the models on the desired input datasets.

*Table 4: Summary of performance of non-pretrained ViT model, with and without augmentation on all three datasets.*

| **Non-pretrained Vision Transformer** | | **Fish species** | | **Fish dataset** | | **Large-scale dataset** | |
|---|---|---|---|---|---|---|---|
| | | Accuracy (%) | Loss | Accuracy (%) | Loss | Accuracy (%) | Loss |
| With Augmentation | Train | 42.35 | 2.379 | 56.66 | 2.202 | 91.731 | 1.014 |
| | Val | 35.25 | 2.64 | 57.143 | 2.195 | 91.929 | 1.011 |
| | Test | 34.02 | 2.98 | 58.295 | 2.176 | 92.074 | 1.012 |
| Without Augmentation | Train | 62.65 | 2.117 | 96.769 | 1.278 | 98.598 | 0.889 |
| | Val | 25.75 | 3.797 | 92.039 | 1.407 | 96.618 | 0.022 |
| | Test | 25.02 | 3.93 | 87.045 | 1.561 | 95.926 | 0.933 |

Collectively, we can notice that the performance metrics of a non-pretrained ViT model is higher on a dataset without augmentation when it is trained up to 100 epochs. The performance is not so bad when data is augmented either, the plots on the other hand provided a clear picture that when images are not augmented the model is more likely to overfit than when augmented.

### 5.2.3 Overall comparison of ViT

Performance of a deep learning model not only upon the accurate results but also factors like computational time and resources required. We have trained and tested the vision transformer model using with and without transfer learning with conditions like with and without image augmentation. As per our observation, the model performs better with transfer learning or we can say that transfer learning improves the performance of a vision transformer model. In addition, the pre-trained ViT with augmented images as input has better performance. Image augmentation does not always enhance the performance of models. Incase of dataset with maximum variations, the addition of augmentation can lead to overfitting like we saw in case of fish species dataset in section 5.2.1.1 and 5.2.2.1.

*Table 5: Overall comparison on test accuracies and losses of a ViT model based on varying conditions.*

| | **Pre-trained Vision Transformer** | | | | **Non-pretrained Vision Transformer** | | | |
|---|---|---|---|---|---|---|---|---|
| | *With Augmentation* | | *Without Augmentation* | | *With Augmentation* | | *Without Augmentation* | |
| | Test | | | | Test | | | |
| | Accuracy | Loss | Accuracy | Loss | Accuracy | Loss | Accuracy | Loss |
| **Fish Species** | 96.29 | 1.575 | 94.133 | 1.987 | 34.02 | 2.98 | 25.02 | 3.93 |
| **Fish dataset** | 97.443 | 1.226 | 98.75 | 1.193 | 58.295 | 2.176 | 87.045 | 1.561 |
| **Large scale fish dataset** | 99.852 | 0.842 | 100 | 0.841 | 92.074 | 1.012 | 95.926 | 0.933 |

In contrast to the pre-trained ViT model, the ViT model without transfer learning is much slower in execution as it is needed to be trained from the scratch. However, the non-pretrained ViT model can be used for smaller datasets and fine tuning. Table 5 shows comparative results of test accuracies for Vision transformer in various conditions and datasets. It is apparent that the results of the pre-trained model are better in all three datasets.

## 5.3 VGG16 and Inception V3 with varying conditions

In-order to get a better comparison on the performance of ViT model with varying techniques and datasets, other deep learning models like VGG16 and Inception V3 were also used with the same techniques involved. Outputs of ViT models only cannot be considered as the best solution. VGG16 and Inception V3 are the Convolutional neural networks that are well known for their performances, comparing their performance with that of ViT can give us more insights and room for improvements.

### 5.3.1 With transfer learning but with and without image augmentation

ImageNet dataset was used to pre-train both VGG16 and Inception V3. Also, keras implementation of VGG16 and Inception V3 was used to initialize the models. Later, the input images were augmented using the same techniques for both models and raw images were given as input for case of without data augmentation.

#### 5.3.1.1 On fish species dataset

In-case of the fish species dataset, we got test accuracies of 65% and 94.75% for VGG16 and Inception V3 respectively when images were augmented. Inception being a much deeper network than VGG16 performs better as its multiple deep layers are capable of extracting important features. Despite being a very deep convolutional network, ViT outperformed both models when fish species image dataset was augmented.

Similarly, we got test accuracies of 60.598% and 82.75% for VGG16 and Inception V3 models respectively when images were not augmented. Here also, inception v3 beats VGG16 with its ability to learn extra features from deeper CNN architecture. Figure 18 and Figure 19 shows the plots of losses and accuracies of pre-trained VGG16 and Inception V3 models respectively when input images are augmented and not augmented. On the side of pre-trained VGG16 as shown in Figure 18 we can observe that the model overfits more when the input images are not augmented. A little overfitting is seen when images are augmented but while comparing with pre-trained ViT models as discussed on section 5.2.1.1, ViT performs better in both cases.

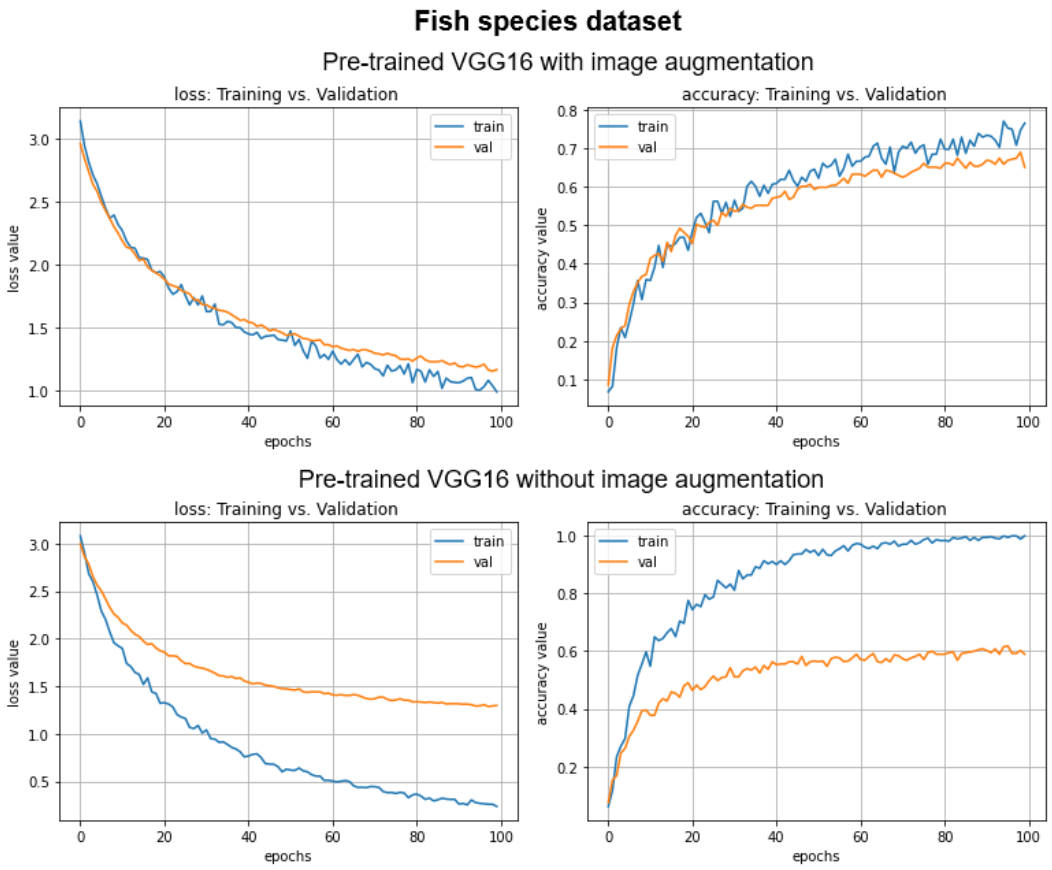


*Figure 18: Plots of losses (on left) and accuracies (on right) for pre-trained VGG16 model with augmentation (top twos) and without (bottom twos) augmentation.*

Inception V3 also displays similar results as of VGG16 for both conditions. The accuracies are higher than that of VGG16 but the model still overfits when the images are not augmented. A sudden drop of accuracy can be seen around the 80$^{th}$ epoch under non-augmented condition but the model training could have been halted after the 20$^{th}$ epochs as we can see that it started to overfit continuously. So, for a dataset having a large number of classes and few numbers of input images, deep CNN architectures like VGG16 and Inception V3 when trained with transfer learning tend to overfit. They are outperformed by the Vision transformer model which is less likely to overfit. Table 6 shows the overall summary for performance on train, test and valid data of pretrained ViT, VGG16 and Inception V3 models. It is noticeable that the ViT model is better at classifying images that are unseen and not involved while training a model.

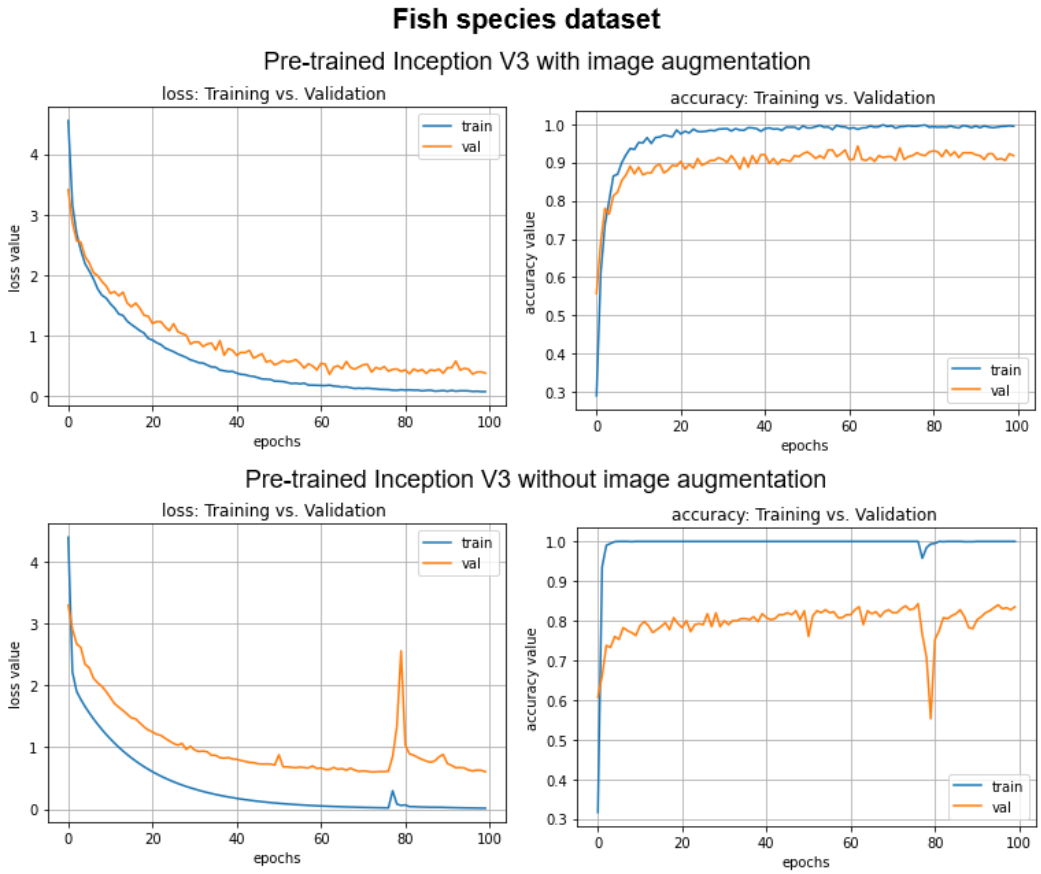


*Figure 19: Plots of losses (on left) and accuracies (on right) for pre-trained Inception V3 model with augmentation (top twos) and without (bottom twos) augmentation.*

*Table 6: Summary of train, test and valid accuracies and losses of all three models when trained with fish species dataset.*

| **Fish Species Dataset** | | | **Vision Transformer** | | **VGG16** | | **Inception V3** | |
|---|---|---|---|---|---|---|---|---|
| | | | Accuracy | Loss | Accuracy | Loss | Accuracy | Loss |
| Pre-Trained | With Augmentation | Train | 99.9 | 1.073 | 73.3 | 1.023 | 99.937 | 0.059 |
| | | Val | 96.75 | 1.147 | 61 | 1.314 | 93.75 | 0.379 |
| | | Test | 96.29 | 1.575 | 65 | 1.178 | 94.75 | 0.339 |
| | Without Augmentation | Train | 100 | 1.068 | 99.55 | 0.246 | 100 | 0.639 |
| | | Val | 96 | 1.148 | 58.75 | 1.298 | 83.3 | 0.604 |
| | | Test | 94.133 | 1.987 | 60.598 | 1.799 | 82.75 | 0.639 |

#### 5.3.1.2 On the fish dataset

While training pre-trained VGG16 and Inception V3 models with fish dataset using image augmentation, we obtained test accuracies of 87.841% and 99.954% respectively. Likewise, we got test accuracies of 95.625% and 97.898% for pre-trained VGG16 and Inception V3 models when using fish dataset without augmentation. Fish dataset being a comparatively larger dataset, the models seem to be performing better when images are not augmented. As seen in section 5.2.1.2 where pre-trained ViT models also performed better when the images were not augmented, the results are similar with pre-trained VGG16 and Inception V3 models. Since the images already contained variations, introducing augmentation was not necessary.

We can look at Figure 20 that gives a comparative plot of losses and accuracies of a pre-trained VGG16 model under conditions when images are augmented and not augmented. The model overfits a little bit in both cases but in case of use of images without augmentation, the model memorizes the features faster as there are not many variations. We can halt the training at about the $20^{th}$ epoch but on the side where image augmentation is being used the train and validation accuracies are in increasing trend but with sight overfitting.

Also, the pre-trained Inception V3 model has much less overfitting as shown in Figure 21. The model learns faster when images are not augmented than when images are augmented. The reason is the same as that for pretrained ViT and VGG16 models. The extensions of CNNs in Inception V3 is reason for exclusion of overfitting that was seen in VGG16. In addition, Table 7 shows a comparative score of all three models when trained using a fish dataset with and without image augmentation. Here, the pre-trained ViT model exceeds both state-of-art performing VGG16 and Inception V3 models.

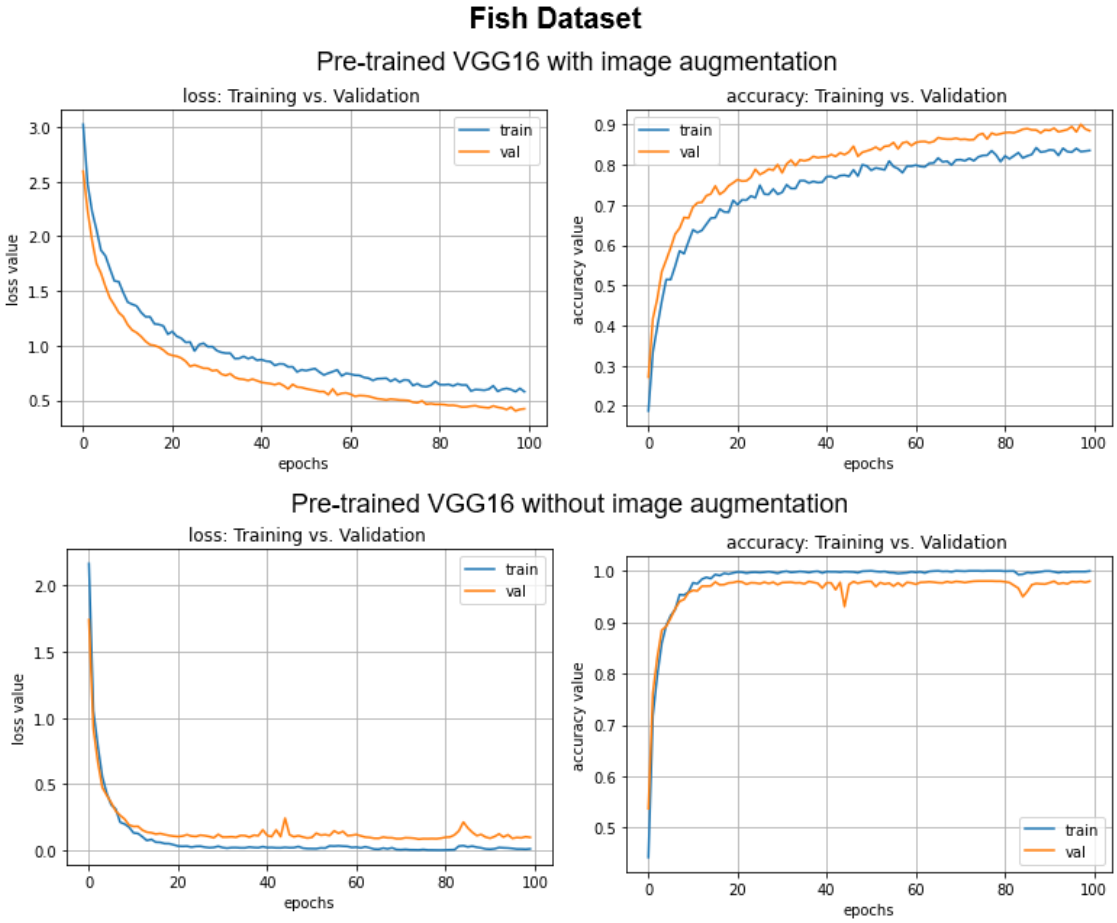


*Figure 20: Losses (on left) and accuracies (on right) of a pre-trained VGG16 model when using fish dataset with augmentation (top twos) and without augmentation (bottom twos).*

*Table 7: Overall train, test and valid losses and accuracies of pre-trained ViT, VGG16 and Inception V3 models when the fish dataset is used with and without augmentation.*

| **The fish dataset** | | | **Vision Transformer** | | **VGG16** | | **Inception V3** | |
|---|---|---|---|---|---|---|---|---|
| | | | Accuracy | Loss | Accuracy | Loss | Accuracy | Loss |
| Pre-Trained | With Augmentation | Train | 98.339 | 1.199 | 83.438 | 0.594 | 99.113 | 0.139 |
| | | Val | 98.11 | 1.211 | 88.481 | 0.421 | 99.055 | 0.154 |
| | | Test | 97.443 | 1.226 | 87.841 | 0.464 | 98.253 | 0.195 |
| | Without Augmentation | Train | 99.886 | 1.162 | 99.966 | 0.007 | 99.954 | 0.035 |
| | | Val | 99.1 | 1.178 | 98.001 | 0.097 | 99.055 | 0.079 |
| | | Test | 98.75 | 1.193 | 95.625 | 0.203 | 97.898 | 0.133 |

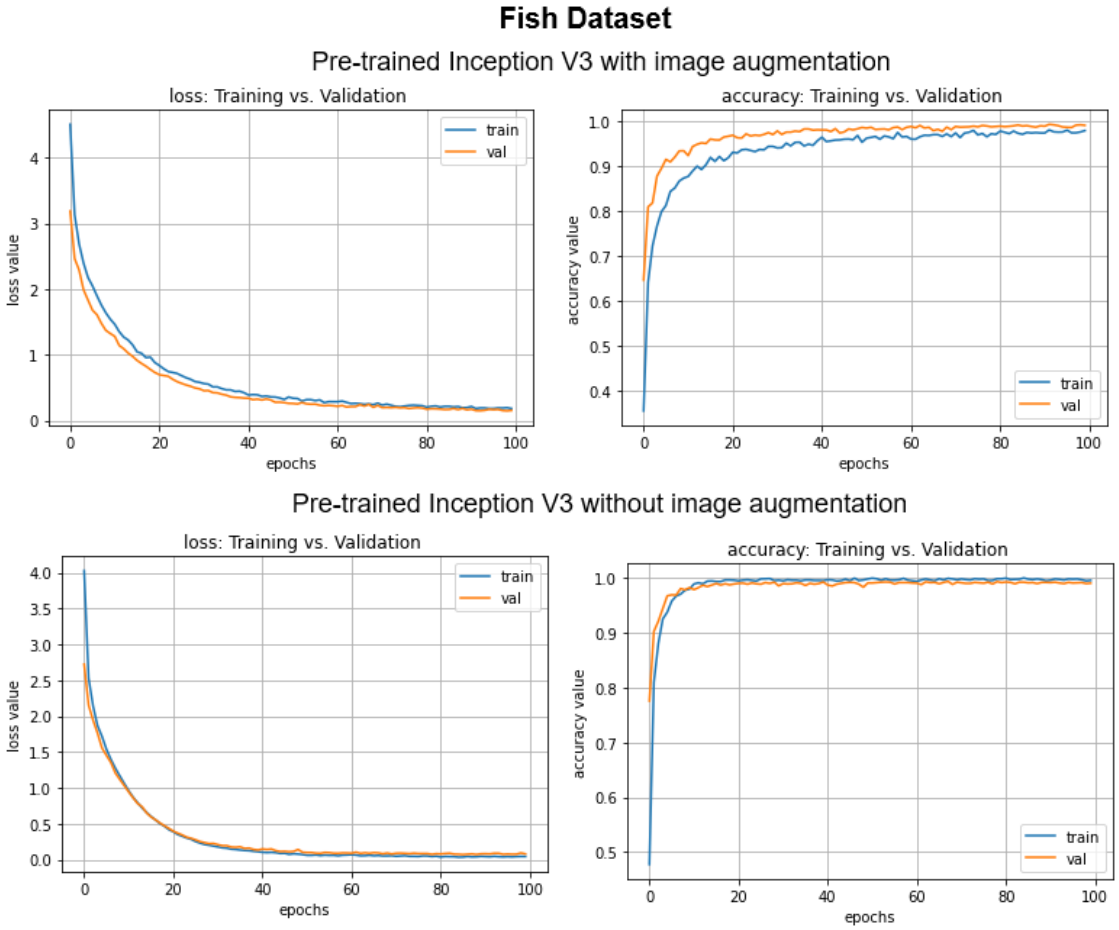


*Figure 21: Losses (on left) and accuracies (on right) of a pre-trained Inception V3 model when using fish dataset with augmentation (top twos) and without augmentation (bottom twos).*

### 5.3.1.3 On A large-scale fish dataset

Pre-trained VGG16 and Inception V3 models when trained on a large-scale fish dataset with image augmentation applied gave test accuracies of 99.778% and 100% respectively. A large-scale fish dataset being relatively small in number of classes and variations allows the deep CNN models like Inception V3 to classify images accurately. Also, when image augmentation was not applied pre-trained VGG16 and Inception V3 models gave test accuracy of 99.741% and 100% respectively. Their performance doesn't differ much when the large-scale fish dataset is augmented and not augmented. The overall performance of all three pre-trained models (ViT, VGG16 and Inception V3) on a large-scale dataset are similar. They all exhibit identical results which can be seen from the plots of losses and accuracies as shown in Figure 22 and Figure 23.

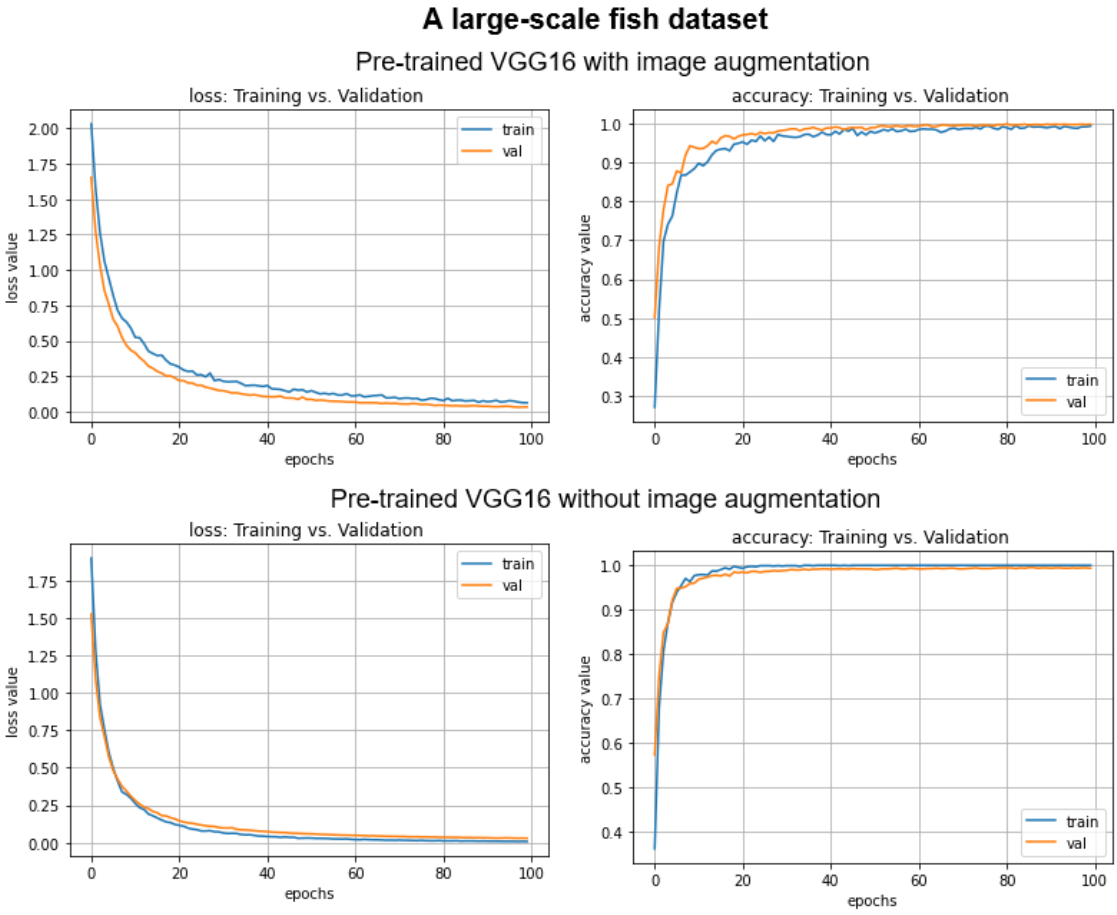


*Figure 22: comparison of losses (on left) and accuracies (on right) curves of pre-trained VGG16 model on large-scale fish dataset with augmentation (top twos) and without augmentation (bottom twos).*

Figure 22 shows performance of pre-trained VGG16 model with and without augmentation on large-scale dataset, where it shows no overfitting and model learns

quickly under both conditions. Whereas, Figure 23 shows performance of pre-trained Inception V3 model, where on both conditions the model's learning rate is faster than VGG16 and begins with high validation accuracy. The pre-trained deep learning models like VGG16 and Inception V3 are also exceptional at image classification but ViT can also be more prominent when it comes to a larger dataset.

Table 8 shows comparative scores of train, test and validation accuracies and losses of all three deep learning architectures while classifying images from large-scale fish dataset. The execution of all three models are high and homogeneous.

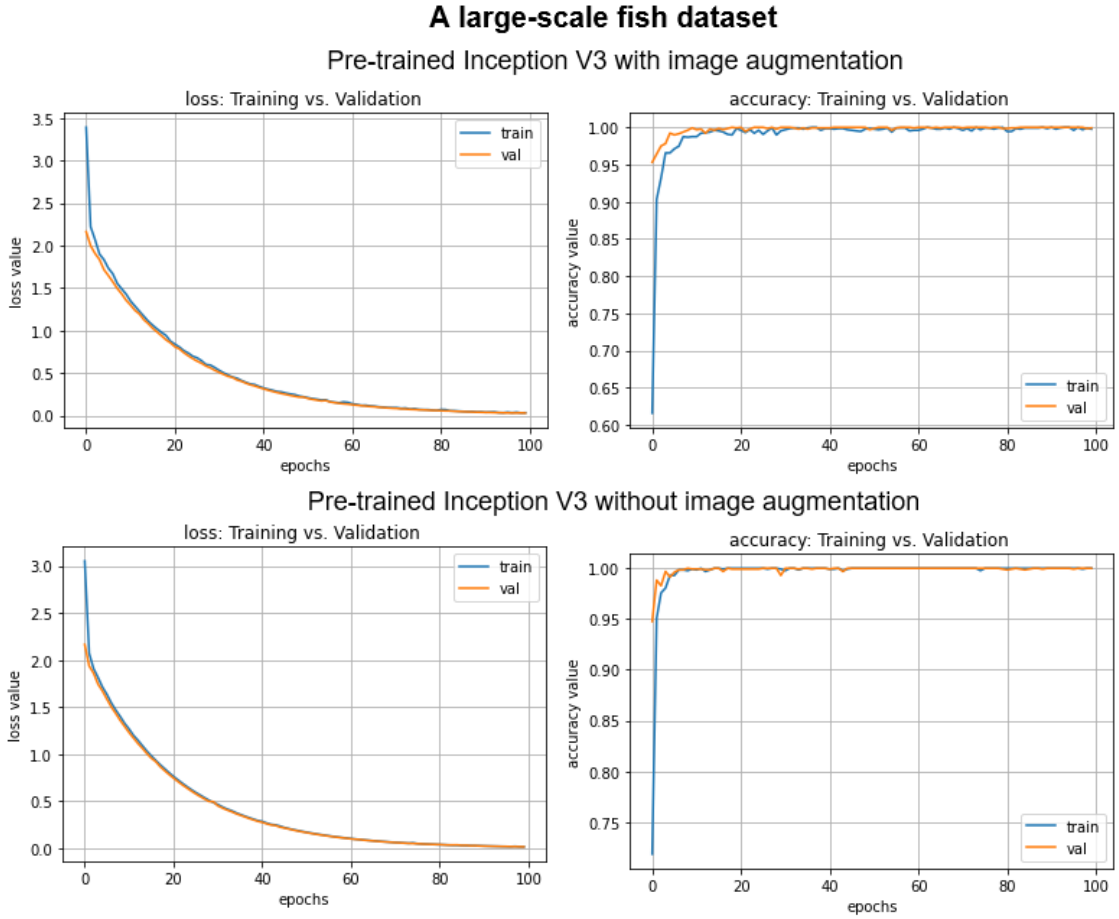


*Figure 23: comparison of losses (on left) and accuracies (on right) curves of pre-trained Inception V3 model on large-scale fish dataset with augmentation (top twos) and without augmentation (bottom twos).*

*Table 8: Train, test, validation losses and accuracies of all three pre-trained models on large scale-fish dataset when image augmentation is applied and not applied.*

| **A large-scale fish dataset** | | | **Vision Transformer** | | **VGG16** | | **Inception V3** | |
|---|---|---|---|---|---|---|---|---|
| | | | Accuracy | Loss | Accuracy | Loss | Accuracy | Loss |
| Pre-trained | With Augmentation | Train | 99.905 | 0.824 | 99.127 | 0.067 | 99.984 | 0.027 |
| | | Val | 100 | 0.841 | 99.762 | 0.034 | 99.923 | 0.028 |
| | | Test | 99.852 | 0.842 | 99.778 | 0.03 | 100 | 0.027 |
| | Without Augmentation | Train | 100 | 0.841 | 100 | 0.007 | 100 | 0.017 |
| | | Val | 100 | 0.841 | 99.286 | 0.03 | 100 | 0.017 |
| | | Test | 100 | 0.841 | 99.741 | 0.024 | 99.926 | 0.018 |

### 5.3.2 Without transfer learning but with and without image augmentation

In order to train a VGG16 and Inception V3 without any transfer learning we used the keras model by setting weights as None. This means that the models will be trained with random weights rather than using pre-trained weights. Like in non-pretrained ViT models the VGG16 and Inception V3 models will also have the input datasets weights only.

#### 5.3.2.1 On fish species dataset

Training non-pretrained deep learning models is a time-consuming process as the models need to initialize and update a large number of parameters. It will also have very low performance in comparison to the pre-trained models. While training a non-pretrained VGG16 and Inception V3 models on augmented fish species dataset, we obtained test accuracies of 30.156% and 45.734% respectively. Correspondingly, while training the non-pretrained models on non-augmented fish species dataset we got test accuracies of 28.465% and 42.174% respectively.

We can compare the performances of these non-pretrained models from the plots shown in Figure 24 and Figure 25 too. The figures show that non-pretrained VGG16 and Inception V3 models overfit when trained with non-augmented images and with augmentation their accuracy is in increasing trend but no improvement of validation accuracy is seen when trained without augmentation. Their performances on non-augmented images leads to the conclusion that the deep learning models tend to overfit when images are not augmented. The reasons can be that the models when trained without any pre-trained weights need diversity with the input data in order to learn more complex features. Without such diversity the model simply memorizes the training set and no generalization will be seen.

We can also look at Table 9 to get information on train, test and validation accuracies and losses of non-pretrained ViT, VGG16 and Inception V3 models where the overall accuracy score for all three models on test data is lower than when pre-trained weights were used. All the models were trained upto 100 epochs but we can check further by increasing the number of epochs as the non-pretrained models were still learning.

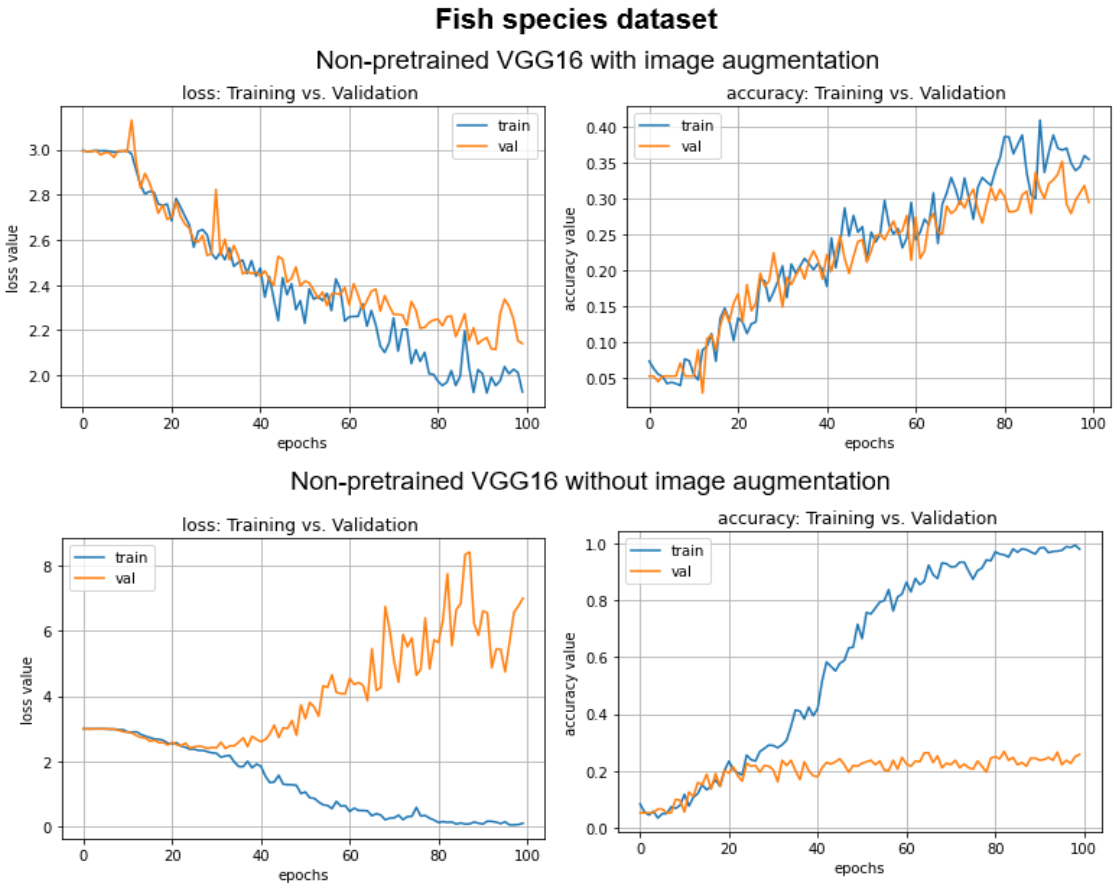


*Figure 24: Plots of losses (on left) and accuracies (on right) for a non-pretrained VGG16 model when trained with fish species dataset with augmentation (top twos) and without augmentation (bottom twos).*

*Table 9: Train, test and validation accuracies and losses of all three non-pretrained deep learning models when trained with augmented and non-augmented Fish species dataset*

| **Fish Species dataset** | | | **Vision Transformer** | | **VGG16** | | **Inception V3** | |
|---|---|---|---|---|---|---|---|---|
| | | | Accuracy | Loss | Accuracy | Loss | Accuracy | Loss |
| Non-Pre-trained | With Augmentation | Train | 42.35 | 2.379 | 37.6 | 1.896 | 59.813 | 0.728 |
| | | Val | 35.25 | 2.64 | 29.25 | 2.139 | 43.617 | 2.518 |
| | | Test | 34.02 | 2.98 | 30.156 | 2.556 | 45.734 | 2.967 |
| | Without Augmentation | Train | 62.65 | 2.117 | 98.85 | 0.061 | 98.663 | 0.249 |
| | | Val | 25.75 | 3.797 | 25.5 | 7.079 | 39.418 | 4.517 |
| | | Test | 25.02 | 3.93 | 28.465 | 3.523 | 42.174 | 3.936 |

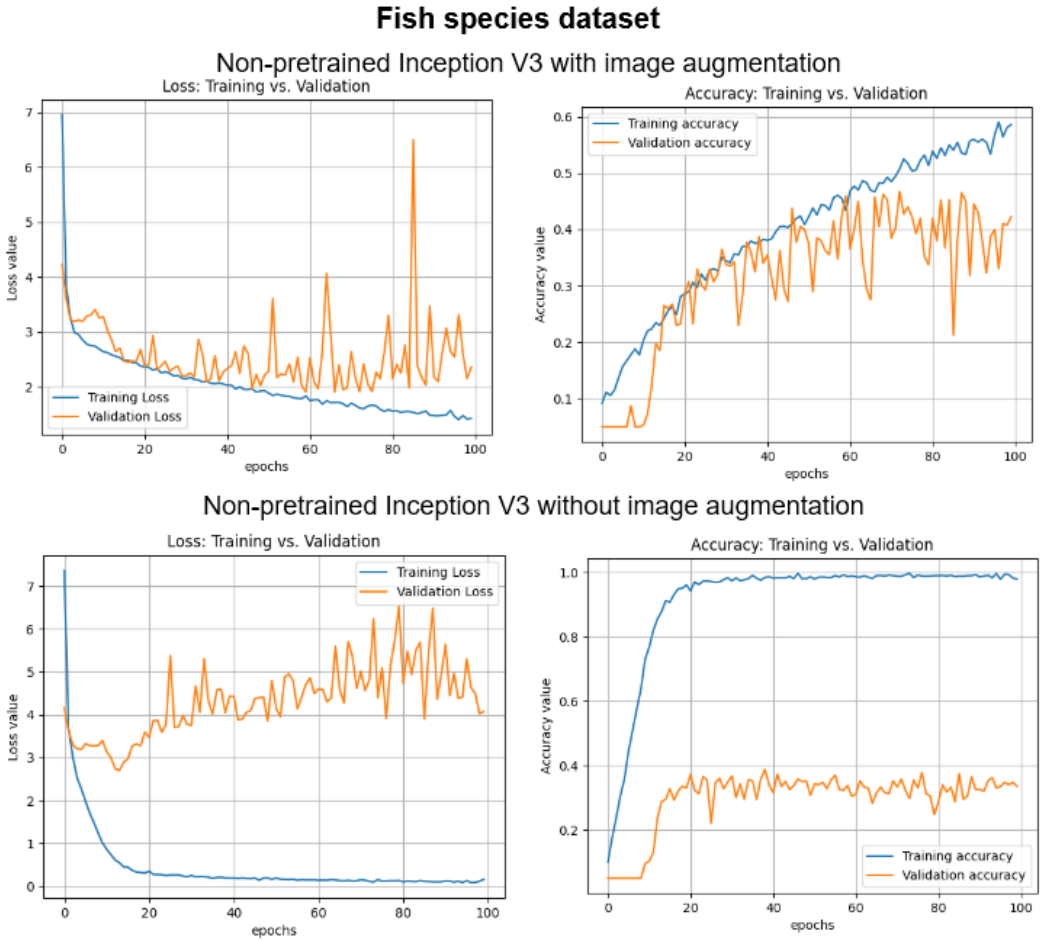


*Figure 25: Plots of losses (on left) and accuracies (on right) for a non-pretrained Inception V3 model when trained with fish species dataset with augmentation (top twos) and without augmentation (bottom twos).*

### 5.3.2.2 On the fish dataset

When using the fish dataset with augmentation on a non-pretrained VGG16 and Inception V3 deep learning models, test accuracies of 87.102% and 39.886% was obtained. Correspondingly, when using the same dataset without augmentation we got 90.909% and 79.148% of test accuracies. Here we can observe that the models are performing better when data is non-augmented but the non-pretrained VGG16 model has a little overfitting than Inception V3. Since there are no pre-trained weights involved, the models need to train from scratch using the weights of input images. Considering the size of the dataset, we can expect overfitting to occur.

In addition, we can look at the plots shown in Figure 26 to evaluate the overall performance of a non-pretrained VGG16 model both with and without image

augmentation applied. The model overfits in both cases but with augmentation the learning rate is a bit slower than without augmentation. For models without augmentation, the training can be halted at around 30$^{th}$ epochs as we are not seeing any improvements in performance but on the other hand, the model with augmentation seems

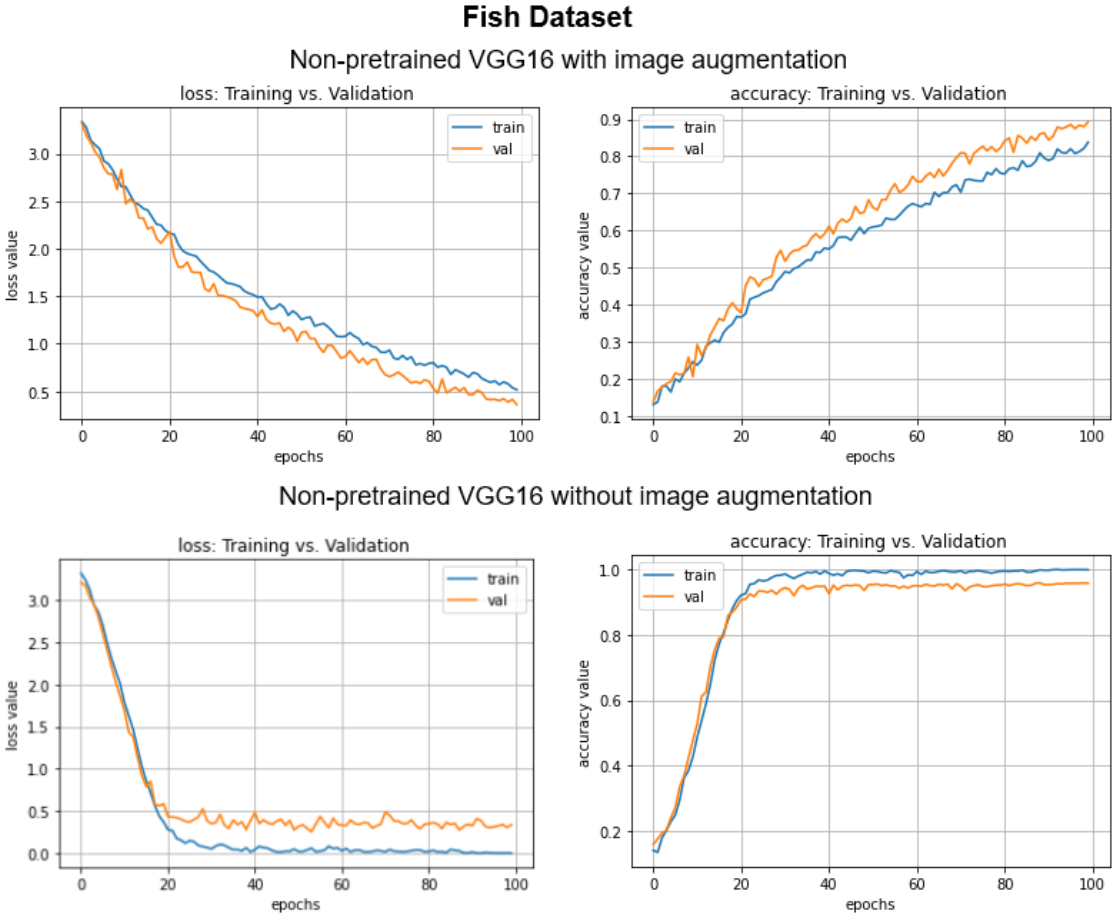


*Figure 26: Accuracies (on right) and losses (on left) plots of non-pretrained VGG16 model when trained on augmented (top twos) and non-augmented (bottom twos) fish dataset.*

to be learning continuously. The execution rate with both conditions are slower than that with transfer learning.

Likewise, in Figure 27 we can see the performance of a non-pretrained Inception V3 model. The model doesn't overfit as much as that of VGG16 but performs better when images are not augmented. The model is memorizing the features as no other general forms are present to extract additional features. Learning trend is also higher when the images are not augmented and pick increases and decreases validation loss and accuracy can be seen when augmented images are used. The results can be compared with non-pretrained ViT models as shown in Table 10 where we can see that the deep CNN

networks outperform ViT on this dataset. In contrast, ViT model doesn't show overfitting hence it can be considered while pre-training weights are not available.

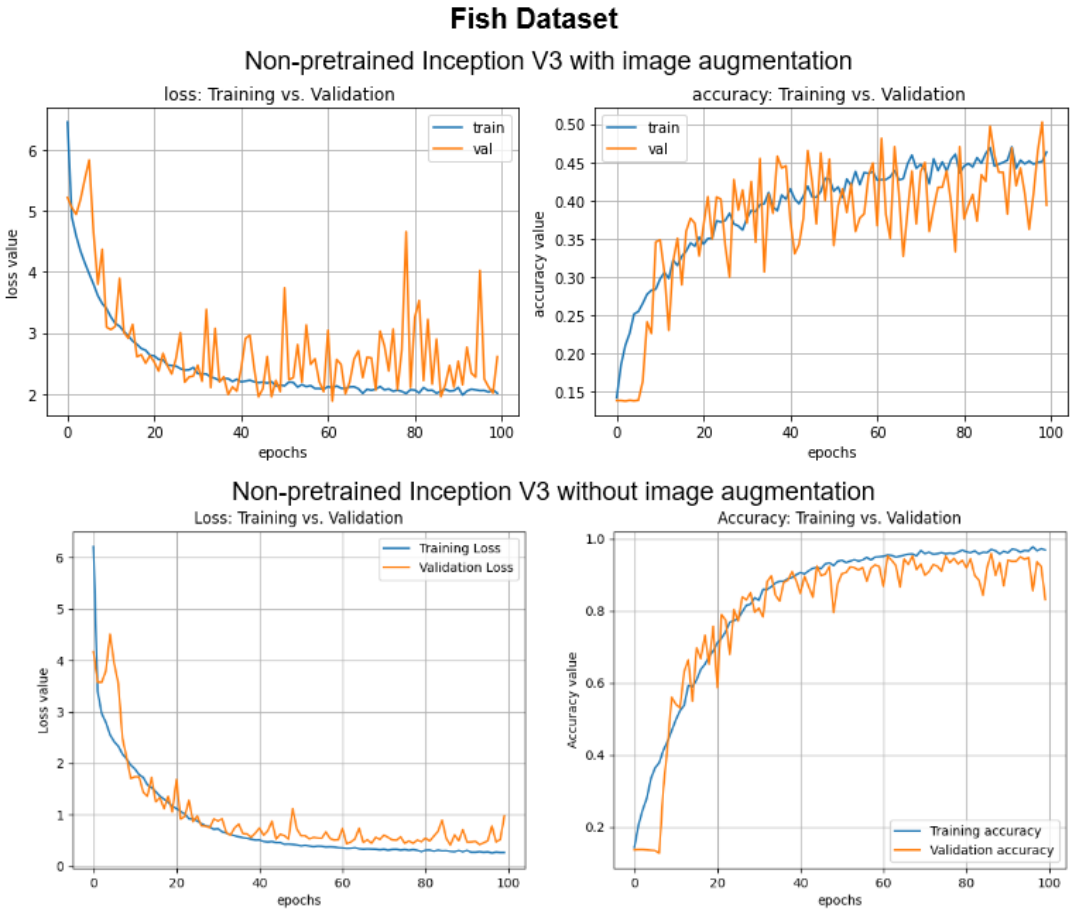


*Figure 27: Accuracies (on right) and losses (on left) plots of non-pretrained Inception V3 model when trained on augmented (top twos) and non-augmented (bottom twos) fish dataset.*

*Table 10: Train, test and validation losses and accuracies of non-pretrained VGG16 and Inception V3 models when using Fish dataset with and without augmentation.*

| **Fish Dataset** | | | **Vision Transformer** | | **VGG16** | | **Inception V3** | |
|---|---|---|---|---|---|---|---|---|
| | | | Accuracy | Loss | Accuracy | Loss | Accuracy | Loss |
| Non-pretrained | With Augmentation | Train | 56.66 | 2.202 | 83.403 | 0.524 | 42.919 | 2.336 |
| | | Val | 57.143 | 2.195 | 89.24 | 0.368 | 39.513 | 2.602 |
| | | Test | 34.02 | 2.98 | 87.102 | 0.47 | 39.8886 | 2.589 |
| | Without Augmentation | Train | 96.769 | 1.278 | 99.977 | 95.856 | 99.813 | 0.081 |
| | | Val | 92.039 | 1.407 | 95.856 | 0.335 | 46.25 | 3.418 |
| | | Test | 25.02 | 3.93 | 90.909 | 0.755 | 79.148 | 1.29 |

### 5.3.2.3 On A large-scale fish dataset

A non-pretrained VGG16 and Inception V3 models gave test accuracies of 98.296% and 72.333% when training with augmented large-scale fish dataset. In the same way, the models had test accuracies of 95.852% and 86.963% when trained without augmentation. In comparison to non-pretrained ViT model, the VGG16 model executes similar results but it excels Inception V3. Like other models discussed above, we can visualize the results on plots of graphs shown in Figure 28 and Figure 29.

As evidenced by Figure 28, the non-pretrained VGG16 model has some overfitting when training with non-augmented images which was not present in non-pretrained ViT model. On the side of training with image augmentation, the validation accuracy is higher than training accuracy. On the other hand, as depicted by plot in Figure 29, the non-pretrained Inception V3 model overall overfits when trained with both augmented and non-augmented images.

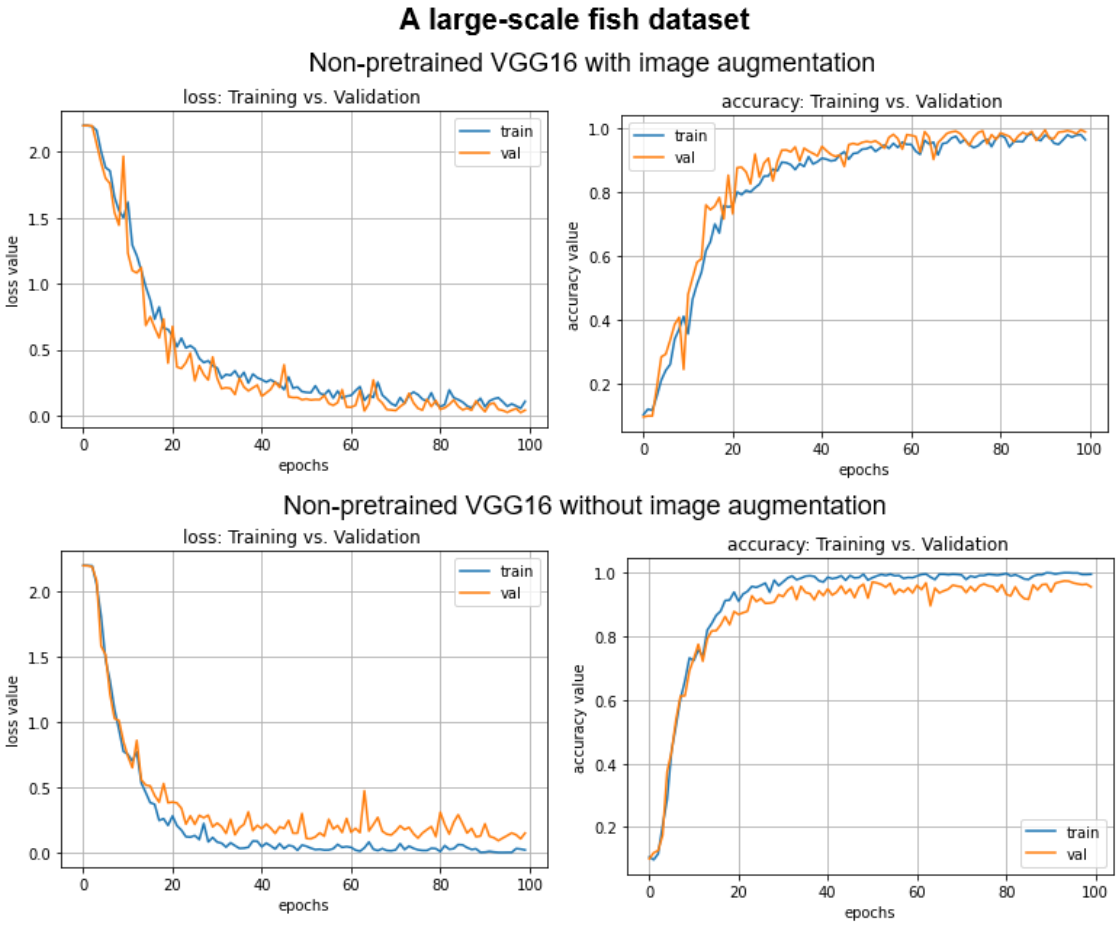


*Figure 28: Comparative plots of accuracies (on right) and losses (on left) for non-pretrained VGG16 model when trained on augmented (top twos) and non-augmented (bottom twos) large-scale fish dataset.*

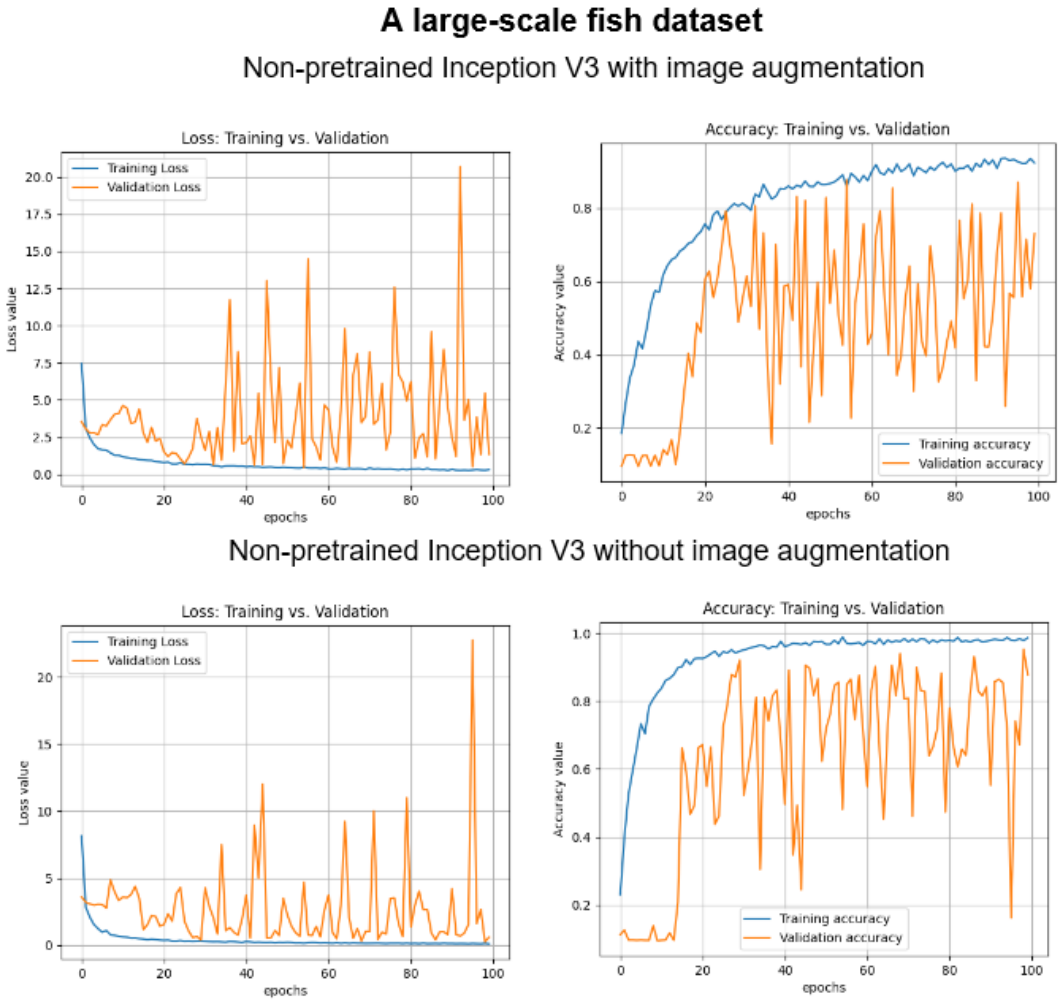


*Figure 29: Comparative plots of accuracies (on right) and losses (on left) for non-pretrained Inception V3 model when trained on augmented (top twos) and non-augmented (bottom twos) large-scale fish dataset.*

We can also compare the results of all three deep learning models on the large-scale fish dataset with the defined conditions. As indicated by Table 11, non-pretrained ViT and VGG16 have high test accuracy scores but as discussed above ViT does not exhibit overfitting conditions.

*Table 11: Accuracies and losses of non-pretrained deep learning models on a large-scale fish dataset with and without image augmentation.*

| **A large-scale fish dataset** | | | **Vision Transformer** | | **VGG16** | | **Inception V3** | |
|---|---|---|---|---|---|---|---|---|
| | | | Accuracy | Loss | Accuracy | Loss | Accuracy | Loss |
| Non-pretrained | With Augmentation | Train | 91.731 | 1.014 | 96.897 | 0.088 | 76.389 | 1.217 |
| | | Val | 91.929 | 1.011 | 98.616 | 0.043 | 73.016 | 1.342 |
| | | Test | 92.074 | 1.012 | 98.296 | 0..059 | 72.333 | 1.419 |
| | Without Augmentation | Train | 98.598 | 0.889 | 99.433 | 0.02 | 92.738 | 0.321 |
| | | Val | 96.618 | 0.022 | 95.542 | 0.148 | 87.778 | 0.592 |
| | | Test | 95.926 | 0.933 | 95.852 | 0.122 | 86.963 | 0.622 |

# 6 Discussion and further work

The primary purpose of this thesis is to study the performance of deep learning neural network models on three different image datasets under varying conditions like with, without transfer learning and image augmentation and conclude their influence and propose a suitable deep learning architecture. After going through the results from all three models we can discuss the core outcome of the study. Transfer learning has been supportive for several deep learning models to give state-of-the-art performance [34]. In this study, we have implemented transfer learning with all three deep learning architectures and also compared their performance with each other and also with the models that do not use transfer learning. Along with transfer learning, we have also included image augmentation methods. That is, models with and without transfer learning were trained with and without data augmentation. The effects if these methods on all the datasets are discussed in this chapter.

## 6.1 On fish species dataset

In section 5.3.1.1 and section 5.3.2.1 we explained about the results from training deep CNNs like VGG 16 and Inception V3 with conditions like with and without transfer learning and image augmentation. The models were mostly overfitted when transfer learning was involved. The fish species dataset was comparatively small in size as we selected small number of images for training and testing purpose and as per our observation, the pre-trained models tend to overfit while being trained on a small dataset. A reason for this can be, as the model is already pre-trained with a huge image dataset like ImageNet, it fits the limited number of inputs using the pre-trained features.

In addition, on a dataset like fish species dataset where the number of classes are high but the number of training data is few, the pre-trained models overfit with and without image augmentation. This might be due to the large number of parameters that the model learns from large dataset like ImageNet. The models might be memorizing the small training data instead of learning. The learning process is fast when transfer learning is used as the models already have pre-trained weights.

Vision Transformer on the other hand displayed less overfitting than the other two models. Results from section 5.2.1.1 and section 5.2.2.1 shows the performance of ViT on fish dataset with varying conditions applied. The cause behind this output can be due to the presence of self-attention mechanism in ViT. ViT models use the self-attention mechanism to learn the features from the input images. Hence, maximum overfitting is seen with deep CNN networks than in ViT, which makes ViT more preferable.

Table 12 lists the overall train, test and validation accuracy and loss of all three models on fish species dataset. We can observe that the test accuracy is different as per the condition and learning model. Like in the study conducted by Parnav et al. [21] where two different versions of VGG model was experimented on dataset of 23 classes of fish, the result of scale-down VGG was better, in our case deep CNN performs well but ViT shines through by excluding the overfitting issue seen on those models. As seen in Table 12, the ViT has high test accuracy in compare to other two models when pre-trained weights are used for training.

*Table 12: Overall train, test and validation accuracy and loss of all three deep learning models on fish species dataset with and without image augmentation.*

| **Fish species dataset** | | | **Vision Transformer** | | **VGG16** | | **Inception V3** | |
|---|---|---|---|---|---|---|---|---|
| | | | Accuracy | Loss | Accuracy | Loss | Accuracy | Loss |
| Pre-Trained | With Augmentation | Train | 99.9 | 1.073 | 73.3 | 1.023 | 99.937 | 0.059 |
| | | Val | 96.75 | 1.147 | 61 | 1.314 | 93.75 | 0.379 |
| | | Test | 96.29 | 1.575 | 65 | 1.178 | 94.75 | 0.339 |
| | Without Augmentation | Train | 100 | 1.068 | 99.55 | 0.246 | 100 | 0.639 |
| | | Val | 96 | 1.148 | 58.75 | 1.298 | 83.3 | 0.604 |
| | | Test | 94.133 | 1.987 | 60.598 | 1.799 | 82.75 | 0.639 |
| Not Pre-Trained | With Augmentation | Train | 42.35 | 2.379 | 37.6 | 1.896 | 59.813 | 0.728 |
| | | Val | 35.25 | 2.64 | 29.25 | 2.139 | 43.617 | 2.518 |
| | | Test | 34.02 | 2.98 | 30.156 | 2.556 | 45.734 | 2.967 |
| | Without Augmentation | Train | 62.65 | 2.117 | 98.85 | 0.061 | 98.663 | 0.249 |
| | | Val | 25.75 | 3.797 | 25.5 | 7.079 | 39.418 | 4.517 |
| | | Test | 25.02 | 3.93 | 28.465 | 3.523 | 42.174 | 3.936 |

## 6.2 On Fish dataset

With datasets like fish dataset where the number of classes and training images are high, the models are less overfitted on both cases. The cause for this can be since the input images were already high in number and contained variations, addition of further transformation might have added noise. Although image augmentations add variations to input data allowing models to learn more features, sometimes it is subjected to add noise to input data as well, which leads to overfitting.

When factors like execution cost and period are considered, ViT architecture is more suitable than the deep CNNs. ViT having transformer architecture with attention mechanism as its core can be favorable in varying conditions like addition of more data and classes during the image classification task. Moreover, attention mechanism of ViT allows it to acknowledge the complex relationships and patterns of an image [3]. Moreover, ViT has less parameters comparison to CNN models which decreases its capacity to memorize data.

We can also look at Table 13 which shows performance report of all three architectures under defined circumstances and conditions. The performances of models when pre-trained does not show much difference but like mentioned when taking abilities and conditions on to account ViT is preferable.

*Table 13: Overall train, test and validation accuracy and loss of all three deep learning models on fish dataset with and without image augmentation.*

| Fish Dataset | | | Vision Transformer | | VGG16 | | Inception V3 | |
|---|---|---|---|---|---|---|---|---|
| | | | Accuracy | Loss | Accuracy | Loss | Accuracy | Loss |
| Pre-Trained | With Augmentation | Train | 98.339 | 1.199 | 83.438 | 0.594 | 99.113 | 0.139 |
| | | Val | 98.11 | 1.211 | 88.481 | 0.421 | 99.055 | 0.154 |
| | | Test | 97.443 | 1.226 | 87.841 | 0.464 | 98.253 | 0.195 |
| | Without Augmentation | Train | 99.886 | 1.162 | 99.966 | 0.007 | 99.954 | 0.035 |
| | | Val | 99.1 | 1.178 | 98.001 | 0.097 | 99.055 | 0.079 |
| | | Test | 98.75 | 1.193 | 95.625 | 0.203 | 97.898 | 0.133 |
| Not Pre-Trained | With Augmentation | Train | 56.66 | 2.202 | 83.403 | 0.524 | 42.919 | 2.336 |
| | | Val | 57.143 | 2.195 | 89.24 | 0.368 | 39.513 | 2.602 |
| | | Test | 34.02 | 2.98 | 87.102 | 0.47 | 39.8886 | 2.589 |
| | Without Augmentation | Train | 96.769 | 1.278 | 99.977 | 95.856 | 99.813 | 0.081 |
| | | Val | 92.039 | 1.407 | 95.856 | 0.335 | 46.25 | 3.418 |
| | | Test | 25.02 | 3.93 | 90.909 | 0.755 | 79.148 | 1.29 |

## 6.3 On a large-scale fish dataset

Further, the third dataset had fewer classes and training images. The outcomes of all three models under pre-trained and non-pretrained conditions were resembling. ViT and deep CNN models demonstrate quite similar results when they are trained under a smaller dataset with few numbers of classes. Since the models were pretrained on a large dataset like ImageNet which contains millions of images with thousands of classes, the models will have learned a large set of features allowing them to classify input images more precisely. Also, for the not-pretrained models, although they don't have any pre-trained features, the deep architectures with number of layers allows them to learn the abstract features from a relatively smaller dataset as well. The performance of the non-pretrained models also depends on the size of dataset being used.

Rauf, H. T. et al. [18] also included a comparative study of their 32-layerd CNN architecture with other eight traditional CNN models. In their study, they compared and concluded their architecture to outperform all the other models. To compare the 32-layerd

with other traditional CNN models they tested with varying conditions like changing number of epochs, learning rate and momentum rate. An overall performance was considered rather than results from a specific condition.

Similarly, in our case although VGG16 and Inception V3 have good performances on certain condition, overall performance of ViT has been exceptional. In a large-scale fish dataset, it shows similar stability, which can be seen in

Table 14 which shows the list of train, test and validation accuracy and loss of all three models under the applied conditions.

Table 14: Overall train, test and validation accuracy and loss of all three deep learning models on a large-scale fish dataset with and without image augmentation.

| **A large-scale fish dataset** | | | **Vision Transformer** | | **VGG16** | | **Inception V3** | |
|---|---|---|---|---|---|---|---|---|
| | | | Accuracy | Loss | Accuracy | Loss | Accuracy | Loss |
| Pre-Trained | With Augmentation | Train | 99.905 | 0.824 | 99.127 | 0.067 | 99.984 | 0.027 |
| | | Val | 100 | 0.841 | 99.762 | 0.034 | 99.923 | 0.028 |
| | | Test | 99.852 | 0.842 | 99.778 | 0.03 | 100 | 0.027 |
| | Without Augmentation | Train | 100 | 0.841 | 100 | 0.007 | 100 | 0.017 |
| | | Val | 100 | 0.841 | 99.286 | 0.03 | 100 | 0.017 |
| | | Test | 100 | 0.841 | 99.741 | 0.024 | 99.926 | 0.018 |
| Not Pre-Trained | With Augmentation | Train | 91.731 | 1.014 | 96.897 | 0.088 | 76.389 | 1.217 |
| | | Val | 91.929 | 1.011 | 98.616 | 0.043 | 73.016 | 1.342 |
| | | Test | 92.074 | 1.012 | 98.296 | 0..059 | 72.333 | 1.419 |
| | Without Augmentation | Train | 98.598 | 0.889 | 99.433 | 0.02 | 92.738 | 0.321 |
| | | Val | 96.618 | 0.022 | 95.542 | 0.148 | 87.778 | 0.592 |
| | | Test | 95.926 | 0.933 | 95.852 | 0.122 | 86.963 | 0.622 |

## 6.4 Further work

Deep CNN models like VGG16, Inception V3, DenseNet and others are already popular and used for computer vision for image classification. However, as studied earlier, the changing conditions can change the performance of these models. ViT has shown promising results in all the diverse conditions and can be considered for image classification tasks.

The study can be extended in future with more study of ViT model under other different conditions. We can study the effects of mutual hyperparameters of ViT and other deep learning models. Patch size is one of the hyper parameters that can affect the results of ViT transformer architecture. Num layers which denote the number of layers in transformer architecture can be changed and tested. Besides image classification, the ViT architecture can also be tested for other tasks like fish identification, detection and behavior analysis. We can further study the performances of other high performing deep learning models that are popular.

# 7 Conclusion

Automatic classification of fish is advantageous to modern aquaculture as it promotes efficiency, provides better accuracy, allows enhanced monitoring of fish population and can support better decision making. This study proposed an effective deep learning model for fish classification which can be more flexible than popular deep CNNs. We also compared the proposed vision transformer model with other two deep learning models with changing conditions and three different datasets. Such a test and study of ViT model shows its versatility on changing conditions of an aquaculture. We trained and compared the ViT model with and without transfer learning with deep CNNs like VGG16 and Inception V3 models. In addition, we also applied conditions like input images being augmented and not augmented to all these models. The three models with such varying conditions were then trained on three different fish image datasets that contained different numbers of images and species of fish.

The proposed Vision transformer is effective and stable on large to small datasets as it reveals stable output in such conditions applied. It can be tuned as per requirement by changing the hyper parameters. More study needs to be done regarding how the hyper parameters influence the performance of the model. Another study regarding fusion of other models on the vision transfer is also possible. The application and scope of the Vision transformer in aquaculture is hopeful and carries an optimistic future.

In summary, Vision transformers are effective to all sets of datasets and can execute in changing conditions. It surpasses deep CNNs in the majority of conditions and has versatile characteristics.

# 8 References


1. Montalbo, F.J.P. and A.A. Hernandez. *Classification of fish species with augmented data using deep convolutional neural network*. in *2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)*. 2019. IEEE.
2. Szegedy, C., et al. *Rethinking the inception architecture for computer vision*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
3. Dosovitskiy, A., et al., *An image is worth 16x16 words: Transformers for image recognition at scale.* arXiv preprint arXiv:2010.11929, 2020.
4. Jamshidi, A., et al., *Advantages of techniques to fortify food products with the benefits of fish oil.* Food Research International, 2020. **137**: p. 109353.
5. Khalili Tilami, S. and S. Sampels, *Nutritional value of fish: lipids, proteins, vitamins, and minerals.* Reviews in Fisheries Science & Aquaculture, 2018. **26**(2): p. 243-253.
6. Nations, F.a.A.O.o.t.U. *Record fisheries and aquaculture production makes critical contribution to global food security*. 2022 20 Feburary 2023]; Available from: https://www.fao.org/newsroom/detail/record-fisheries-aquaculture-production-contributes-food-security-290622/en.
7. Department, F.F.a.A. *The state of world fisheries and aquaculture*. 2006 20 Feburary 2023]; Available from: https://www.fao.org/3/a0699e/a0699e.pdf.
8. Yang, X., et al., *Deep learning for smart fish farming: applications, opportunities and challenges.* Reviews in Aquaculture, 2021. **13**(1): p. 66-90.
9. Sun, M., X. Yang, and Y. Xie, *Deep learning in aquaculture: A review.* J. Comput, 2020. **31**(1): p. 294-319.
10. Mizuta, D.D., H.E. Froehlich, and J.R. Wilson, *The changing role and definitions of aquaculture for environmental purposes.* Reviews in Aquaculture, 2023. **15**(1): p. 130-141.
11. Vo, T.T.E., et al., *Overview of smart aquaculture system: Focusing on applications of machine learning and computer vision.* Electronics, 2021. **10**(22): p. 2882.
12. Kassem, T., et al., *Smart and Sustainable Aquaculture Farms.* Sustainability, 2021. **13**(19): p. 10685.
13. Imai, T., K. Arai, and T. Kobayashi. *Smart aquaculture system: A remote feeding system with smartphones*. in *2019 IEEE 23rd International Symposium on Consumer Technologies (ISCT)*. 2019. IEEE.
14. Olyaie, E., H.Z. Abyaneh, and A.D. Mehr, *A comparative analysis among computational intelligence techniques for dissolved oxygen prediction in Delaware River.* Geoscience Frontiers, 2017. **8**(3): p. 517-527.
15. LeCun, Y., Y. Bengio, and G. Hinton, *Deep learning.* nature, 2015. **521**(7553): p. 436-444.

16. Rathi, D., S. Jain, and S. Indu. *Underwater fish species classification using convolutional neural network and deep learning*. in *2017 Ninth international conference on advances in pattern recognition (ICAPR)*. 2017. Ieee.
17. Lee, D.-J., et al., *Contour matching for fish species recognition and migration monitoring.* Applications of Computational Intelligence in Biology: Current Trends and Open Problems, 2008: p. 183-207.
18. Rauf, H.T., et al., *Visual features based automated identification of fish species using deep convolutional neural networks.* Computers and electronics in agriculture, 2019. **167**: p. 105075.
19. Yusup, I., M. Iqbal, and I. Jaya. *Real-time reef fishes identification using deep learning*. in *IOP Conference Series: Earth and Environmental Science*. 2020. IOP Publishing.
20. Wolff, L.M. and S. Badri-Hoeher. *Imaging sonar-based fish detection in shallow waters*. in *2014 Oceans-St. John's*. 2014. IEEE.
21. Thorat, P., R. Tongaonkar, and V. Jagtap. *Towards designing the best model for classification of fish species using deep neural networks*. in *Proceeding of International Conference on Computational Science and Applications: ICCSA 2019*. 2020. Springer.
22. Iqbal, M.A., et al., *Automatic fish species classification using deep convolutional neural networks.* Wireless Personal Communications, 2021. **116**: p. 1043-1053.
23. Abinaya, N., D. Susan, and R. Kumar, *Naive Bayesian fusion based deep learning networks for multisegmented classification of fishes in aquaculture industries.* Ecological Informatics, 2021. **61**: p. 101248.
24. Iqbal, U., D. Li, and M. Akhter, *Intelligent Diagnosis of Fish Behavior Using Deep Learning Method.* Fishes, 2022. **7**(4): p. 201.
25. Zhao, J., et al., *Modified motion influence map and recurrent neural network-based monitoring of the local unusual behaviors for fish school in intensive aquaculture.* Aquaculture, 2018. **493**: p. 165-175.
26. Yu, C., et al., *Segmentation and measurement scheme for fish morphological features based on Mask R-CNN.* Information Processing in Agriculture, 2020. **7**(4): p. 523-534.
27. Garcia, R., et al., *Automatic segmentation of fish using deep learning with application to fish size measurement.* ICES Journal of Marine Science, 2020. **77**(4): p. 1354-1366.
28. Alshdaifat, N.F.F., A.Z. Talib, and M.A. Osman, *Improved deep learning framework for fish segmentation in underwater videos.* Ecological Informatics, 2020. **59**: p. 101121.
29. Álvarez-Ellacuría, A., et al., *Image-based, unsupervised estimation of fish size from commercial landings using deep learning.* ICES Journal of Marine Science, 2020. **77**(4): p. 1330-1339.
30. Petrellis, N., *Measurement of fish morphological features through image processing and deep learning techniques.* Applied Sciences, 2021. **11**(10): p. 4416.

31. Spampinato, C., et al., *Detecting, Tracking and Counting Fish in Low Quality Unconstrained Underwater Videos.* VISAPP (2), 2008. **2008**(514-519): p. 1.
32. French, G., et al., *Convolutional neural networks for counting fish in fisheries surveillance video.* 2015.
33. Bhavsar, D. *Dispelling Myths: Deep Learning vs. Machine Learning*. 2020 3 March 2023]; Available from: https://www.merkle.com/blog/dispelling-myths-deep-learning-vs-machine-learning.
34. Marcelino, P. *Transfer learning from pre-trained models*. 2018 [cited 2023 6th March]; Available from: https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751.
35. Li, F.-F. *ImageNet Database*. 2006 [cited 2023 7th March]; Available from: https://image-net.org/index.php.
36. Mueller, V. *Deep transfer learning: The art of reusing models trained by others*. 2021 [cited 2023 7th March]; Available from: https://towardsdatascience.com/transfer-learning-3e9bb53549f6.
37. Joshi, N. *Exploring the limits of transfer learning*. 2020 [cited 2023 10th March]; Available from: https://www.allerin.com/blog/exploring-the-limits-of-transfer-learning.
38. Keras. *Image data preprocessing*. [cited 2023; Available from: https://keras.io/api/preprocessing/image/.
39. Morales, F. *vit-keras*. 2020 [cited 2023 25th Feburary]; Available from: https://github.com/faustomorales/vit-keras.
40. Cristina, S. *The Vision Transformer Model*. 2022 2023]; 24th Feburary]. Available from: https://machinelearningmastery.com/the-vision-transformer-model/.
41. Georgiou, G., *Fish species*. 2020: Kaggle.
42. Mark Daniel Lampa, R.C.L., Mary Mae Calamba, *Fish Dataset*. 2022: Kaggle.
43. Ulucan, O., D. Karakaya, and M. Turkan. *A large-scale dataset for fish segmentation and classification.* in *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*. 2020. IEEE.


**Norges miljø- og biovitenskapelige universitet**
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway