Fish Identification through Convolutional Neural Network

Fish dataset for classification

ECS 171 - Group 6 Project Report

Justine Arjo Melvin Tong Evan Silvey Jordan Wong Eric Du

December 8, 2023

Contents

1 Introduction

Computer vision has revolutionized the way researchers collect and analyze data across various fields, including marine biology and ichthyology — the study of fish. A fish dataset typically comprises numerous images capturing various fish species in different settings. Studying such a dataset can unlock a multitude of benefits and insights, ranging from ecosystem monitoring to the advancement of machine learning techniques.

The goal of our research is to create a simple Convolutional Neural Network (CNN) model which can run on minimal hardware. Such use cases could be a fisherman out at sea, or a environmentalist documenting fish. We will compare our model to cutting edge architectures in classification and try to reduce parameter count and complexity while keeping a high accuracy.

This task may be of interest to the food or fishing sector and biodiversity for multiple reasons:

- **Biodiversity Monitoring**: By identifying and cataloging different fish species, researchers can monitor biodiversity in aquatic environments. This can be essential for ecological research and the protection of ecosystems.

- **Fisheries Management**: Accurate data on fish species and populations helps in managing fisheries. Overfishing is a significant problem, and computer vision can help regulate catches to sustainable levels.

- **Species Conservation**: Some fish species are endangered. Computer vision can help identify those species when fished, and alert fishermen when they should be released.

- **Aquaculture and anomaly detection**: The aquaculture industry can benefit from computer vision by monitoring the health and growth of fish in farms, leading to more efficient and healthy production. By analyzing images, it is possible to detect potential anomalies and prevent food intoxication or disease propagation.

- **Research and Education**: A visual dataset can be a valuable tool for educational purposes, helping students and researchers to learn about marine biology.

2 Literature Review

2.1 Image Classification

The main goal of our project is to recognize fishes in images. Image classification, a hot topic in deep learning, involves categorizing and labeling images into predefined classes. If models existed before the 90s to tackle this problem with numerical and categorical data, visual dataset represented a great challenge due to its significant complexity.

Early attempts at image classification predating modern deep learning relied on hand-engineered features and traditional machine learning algorithms. These methods required extensive domain knowledge to design features appropriate for specific image types and classification tasks. However, they often struggled with high dimensions of image data, leading to limitations in generalization and adaptability.

The landscape of image classification saw a revolutionary change with the introduction of Convolutional Neural Networks (CNNs). The foundational work by Yann LeCun in 1998, titled “Gradient-Based Learning Applied to Document Recognition”, introduced a specialized kind of neural network for processing data with a grid-like topology. This work, leveraging the backpropagation algorithm, laid the groundwork for modern CNNs.

The architecture of CNNs is inspired by the organization of the animal visual cortex and is specifically tailored to handle image data.

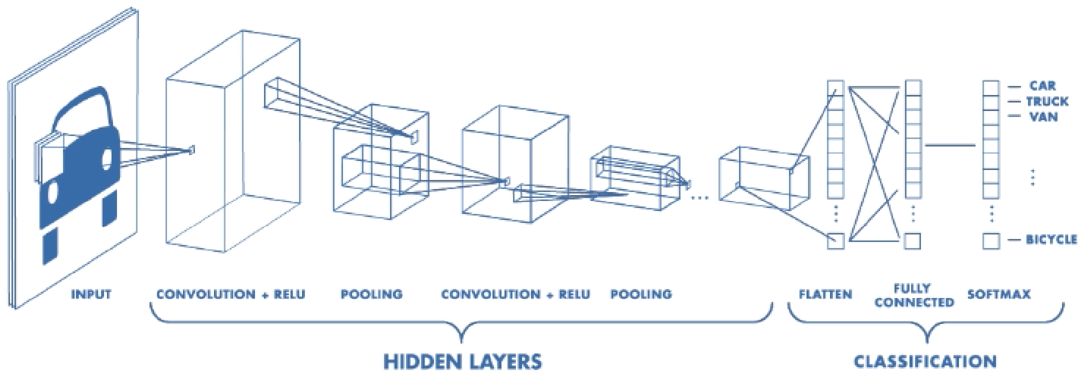


Figure 1: Convolutional Neural Network (CNN) Architecture for Image Classification

Unlike fully connected networks, where each input is connected to every neuron, CNNs preserve the spatial relationships between pixels by employing a series of convolutional layers. These layers apply a set of learnable filters to the input image, effectively capturing local features such as edges, shapes, and textures. Following the convolutional layers, pooling layers are used to reduce the spatial dimensions of the extracted features, decreasing the number of parameters and computation in the network. This reduction is not only beneficial for computational efficiency but also aids in making the network invariant to small shifts and distortions in the input image.

In 1998, Yann LeCun and his team introduced the MNIST database, a comprehensive collection of handwritten digits sourced from American Census Bureau employees and high school students in the United States. This database has subsequently emerged as a standard benchmark for assessing the performance of handwriting recognition systems, and is a proof that CNNs are an highly efficient method. The MNIST database is still widely used today [4].

2.2 Other Model Architectures: VGG16 and ResNet50

We will be using transfer learning of two other models, VGG-16 and ResNet-50 with imagenet weights loaded in as a baseline to compare against our model. The additional layers we are adding in is a 64 filter Dense layer followed with Batch Normalization and a Dropout with 0.2 rate. We introduce briefly these models to better understand the following results in our code. ResNet-50 on this dataset have been reported to be have 100% accuracy [6], while VGG-Nets have reported a 98% accuracy [1].

VGG-16 is a CNN model created by a team in the Visual Geometry Group department at the University of Oxford in 2014 [8]. It is composed of 16 layers, featuring a uniform architecture of 3x3 convolutional filters designed for effective image feature learning and achieving strong performance in computer vision tasks [3].

ResNet-50 is also a CNN model introduced in the 2015 by He Kaiming, Zhang Xiangyu, Ren Shaoqing, and Sun Jian's [2]. Made of 50 layers, its key strength lies in its use of residual learning, which facilitates the training of very deep neural networks by addressing the van-

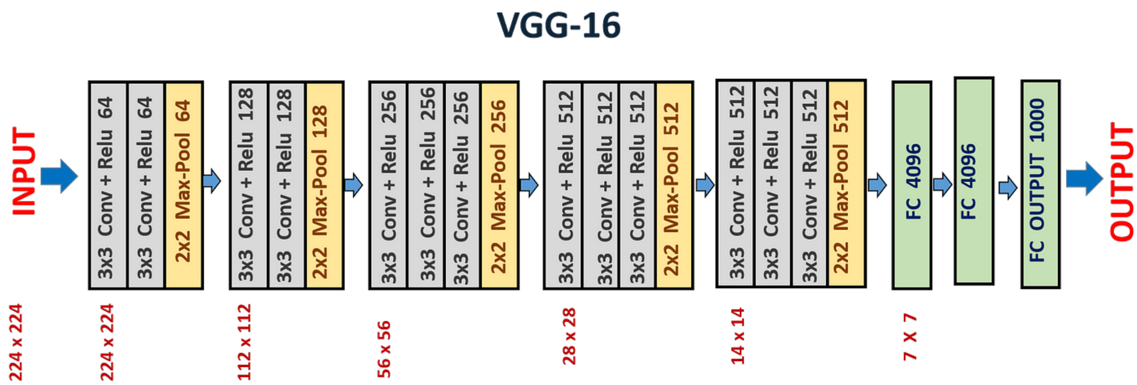


Figure 2: VGG16 (CNN) Model Architecture for Deep Learning

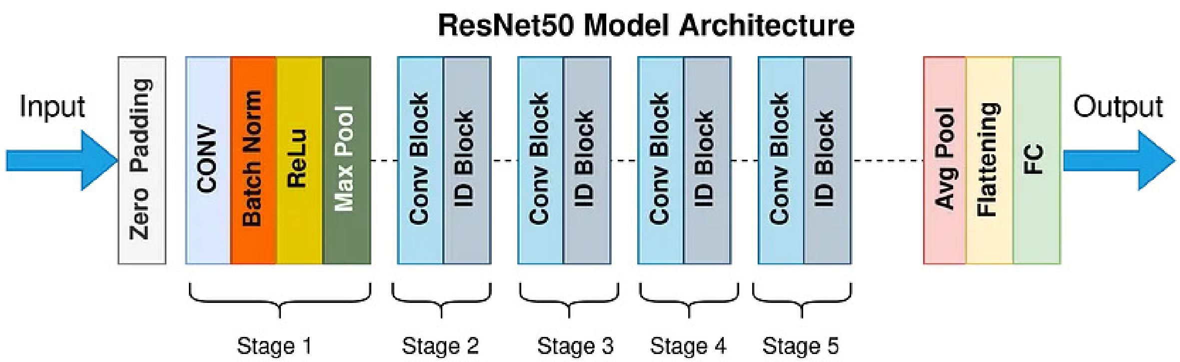


Figure 3: Resnet50 (CNN) Model Architecture for Deep Learning

ishing gradient problem and enables effective feature learning, leading to improved performance in image recognition tasks [5].

3 Dataset Description and exploratory data analysis

The dataset contains 9.000 images. It has been resized to 590 × 445 by its owner, while preserving the aspect ratio. The dataset has also been augmented using flipping and rotating. There are 9 different seafood types: Black Sea Prat, Gilt-Head Bream, Hourse Mackerel, Red Mullet, Sea Bass, Shrimp, Striped Red Mullet and Trout.

For each species, there are 1000 augmented images. None of the images is corrupted or contains artifacts, and they are all readable.

4 Proposed Methodology

4.1 Data preprocessing

The dataset was retrieved from Kaggle [7]. As explained before, it contains augmented images with rotation and flipping. To avoid compromising data integrity, no further augmentation was performed on the dataset. We divide the dataset into 3 parts: training set that constitute 70% of the data, validation set that constitute 20% of the data, and test sets that constitute 10% of the data.

4.2 Design the CNN Architecture

We propose a simple model that comprises of 3 major CNN layers and a dense layer for regularization and output. The first layer is a 2D convolution layer that handles an input shape of 400 (w) x 267 (h) x 3 (c) with 32 filters and ReLU as activation function. After the convolution, the data went though a batch normalization, a 2D maximum pooling with a 2 by 2 pool size , and a dropout layer with a 0.2 dropout rate.

The second CNN layer have a similar structure as the first CNN layer with the exception that 64 layers are used instead of only 32. The third layer follow suit but without the dropout layer.

4.3 Configure the Learning Process

The models were all trained with validation accuracy early stopping with patience 5 along with 10% learning rate reduction on patience 3. We loaded our dataset into batches of 32 and trained 30 epochs for our CNN and 15 for ResNet and VGG. ADAM is used starting at 0.001 learning rate and categorial cross entropy is used as the loss function. Parameters were tuned manually based on classification report metrics due to each model taking 30+ minutes to train with GPU Acceleration.

4.4 Train the Model

Once the model is established, we give the training data into the model, allowing it to learn from the data. This is where the CNN adjusts its weights through back-propagation. During this crucial step, we monitor our results. To do so, we use the validation set to observe the model's performance and try avoiding over-fitting according to the results.

4.5 Evaluate the Model

During each epoch of our training, we use different metrics to evaluate how well our model is performing. We decided to check metrics seen in class, such as accuracy, precision and recall, represented in a confusion matrix.

5 Experimental results

5.1 Simple CNN Model Results

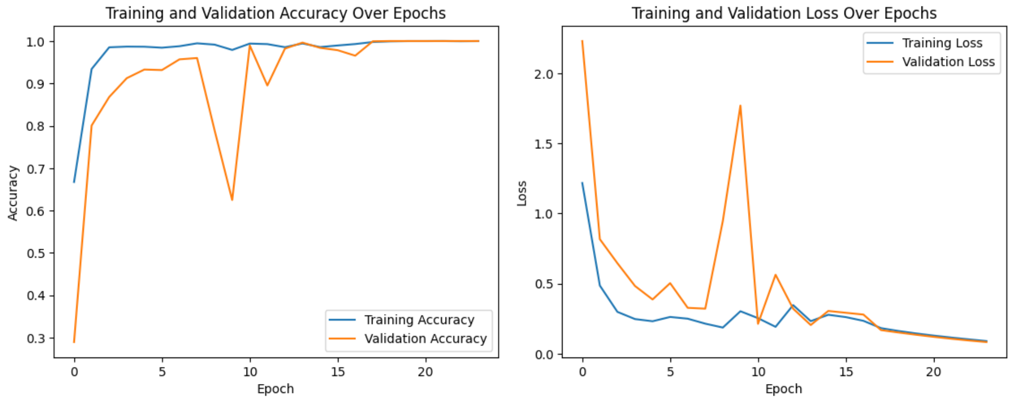
We have achieved 100% test accuracy on testing for our best model and kept the parameter count to 6152713 total parameters.

5.2 ResNet-50 Model Results

ResNet-50 also gets to 100% test accuracy, however the parameter count is 23719689, 3.85 times more parameters than our simple CNN.

5.3 VGG-16 Model Results

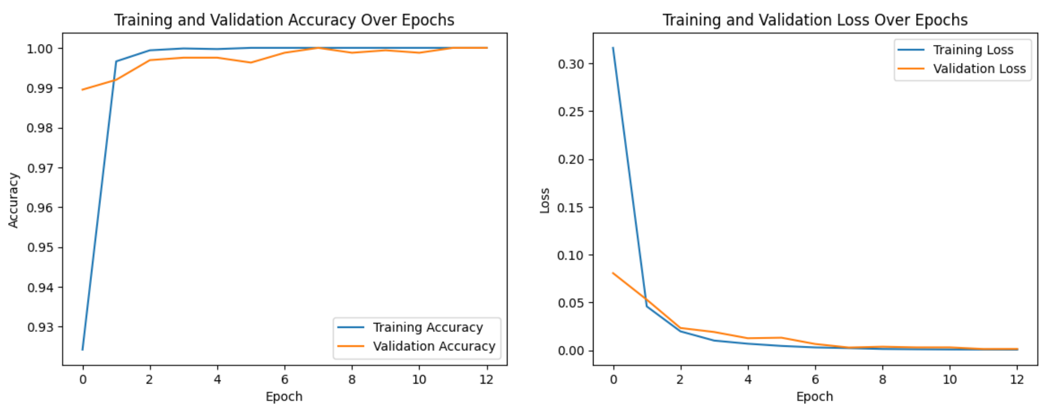
The VGG-16 Model did pretty good with 99.44% test accuracy, however it looks like it was over-fitting and wasn't able to capture some features of the fish. Despite having more parameters at 14748361 total parameters, it seems to not be able to capture some features.



Black Sea Sprat	1.00	1.00	1.00	91
Gilt-Head Bream	1.00	1.00	1.00	121
Hourse Mackerel	1.00	1.00	1.00	99
Red Mullet	1.00	1.00	1.00	108
Red Sea Bream	1.00	1.00	1.00	103
Sea Bass	1.00	1.00	1.00	100
Shrimp	1.00	1.00	1.00	96
Striped Red Mullet	1.00	1.00	1.00	79
Trout	1.00	1.00	1.00	103
accuracy			1.00	900
macro avg	1.00	1.00	1.00	900
weighted avg	1.00	1.00	1.00	900

```
================================================================
Total params: 6152713 (23.47 MB)
Trainable params: 6152265 (23.47 MB)
Non-trainable params: 448 (1.75 KB)
________________________________________________________________
```

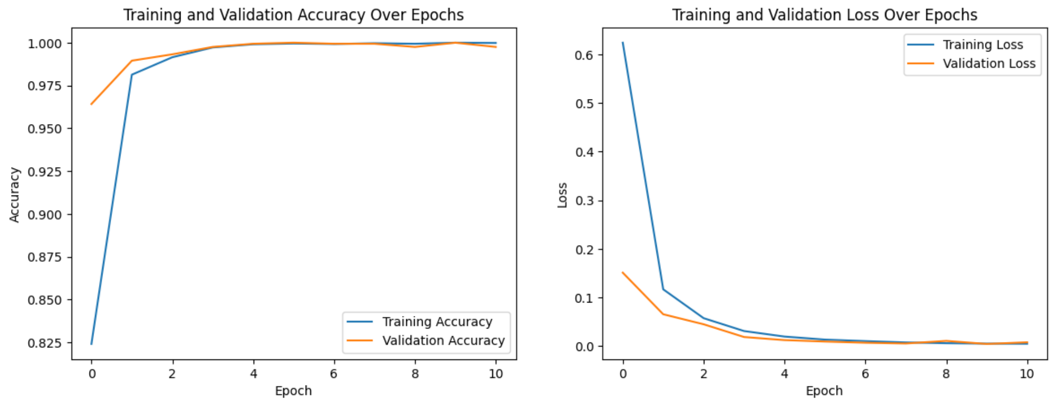
Figure 4: Complete results for our CNN model : Accuracy, Loss, Classification report and Parameters



Black Sea Sprat	1.00	1.00	1.00	103
Gilt-Head Bream	1.00	1.00	1.00	100
Hourse Mackerel	1.00	1.00	1.00	91
Red Mullet	1.00	1.00	1.00	96
Red Sea Bream	1.00	1.00	1.00	108
Sea Bass	1.00	1.00	1.00	99
Shrimp	1.00	1.00	1.00	121
Striped Red Mullet	1.00	1.00	1.00	103
Trout	1.00	1.00	1.00	79
accuracy			1.00	900
macro avg	1.00	1.00	1.00	900
weighted avg	1.00	1.00	1.00	900

```
================================================================
Total params: 23719689 (90.48 MB)
Trainable params: 131849 (515.04 KB)
Non-trainable params: 23587840 (89.98 MB)
________________________________________________________________
```

Figure 5: Complete results for Resnet50 model : Accuracy, Loss, Classification report and Parameters



	precision	recall	f1-score	support
Black Sea Sprat	1.00	1.00	1.00	103
Gilt-Head Bream	0.99	0.97	0.98	100
Hourse Mackerel	1.00	1.00	1.00	91
Red Mullet	1.00	0.99	0.99	96
Red Sea Bream	0.97	0.99	0.98	108
Sea Bass	1.00	1.00	1.00	99
Shrimp	1.00	1.00	1.00	121
Striped Red Mullet	0.99	1.00	1.00	103
Trout	1.00	1.00	1.00	79
accuracy			0.99	900
macro avg	0.99	0.99	0.99	900
weighted avg	0.99	0.99	0.99	900

```
================================================
Total params: 14748361 (56.26 MB)
Trainable params: 33545 (131.04 KB)
Non-trainable params: 14714816 (56.13 MB)
________________________________________________
```

Figure 6: Complete results for VGG16 model : Accuracy, Loss, Classification report and Parameters

6 Conclusion and discussion

6.1 Project Road Map

Throughout the timeline of the project, we initially had four large checkpoints for the road map to creating the CNN. Each checkpoint had it's respective due date, listed below.

- **10/29 - Pre-processing and Data Visualization**: We expect to be able to pre-process images such that we remove outliers and make images generic for the computer to handle. Within this, we allow for re-scaling, gray-scaling, and other augmentations to the data before we feed it into the CNN.

- **11/12 - Image Classification**: This is where the CNN was meant to be built and iterated upon. The model would be able to classify images from the data set and provide characteristics and correct classification. We were not sure of it's accuracy, leading to object detection as a way to hopefully improve the model.

- **11/26 - Object Detection using Segmentation**: This goal would allow the model to essentially differentiate the edges of the fish from the surroundings. Additionally it would theoretically allow for multiple fish to be in the same frame, although such scenario was not present in our data set. Overall, we hoped for this to improve our model by providing a more succinct image.

- **12/01 - Anomaly Detection**: This was a stretch goal which did not align much with our data set. However, if we had time, this detection would showcase any anomalies present between fish of the same breed (i.e mutations).

However, instead of segmentation and anomaly detection tasks, we decided to work on existing CNN models to get a better comparison of our work.

We did not stick too closely to the road map. Rather, the map was a guideline for goals to complete and things to achieve. We did create functions like gray-scaling that did not end up in our final product, as we found the CNN to do well just being fed the data set. It also did not take into account deliverable such as the documentation and website, but communication and flexibility allowed for these tasks to be completed.

6.2 Conclusions and Takeaways

The CNN built in this project works well within the confines of the data set provided, but is much less sophisticated than the models compared to. Some things we could do to improve upon this is to process more data, which we did not focus too heavily on. Based on our goals set for this project, we believe they were achieved based on the resources provided. We hope for the takeaways of creating a CNN to follow into future projects.

References

[1] Chaitana. *Classifying Fishes ‖ VGG Nets 99% acc.* https://www.kaggle.com/code/chaitanya99/classifying-fishes-vgg-nets-99-acc [Accessed: (11.22.2023)]. 2021.

[2] Kaiming He. *Deep Residual Learning for Image Recognition.* https://arxiv.org/pdf/1512.03385.pdf. 2015.

[3] Vaibhav Khandelwal. *The Architecture and Implementation of VGG-16.* https://pub.towardsai.net/the-architecture-and-implementation-of-vgg-16-b050e5a5920b [Accessed: (11.22.2023)]. 2020.

[4] Yann LeCun. *Convolutional Networks and Applications in Vision.* http://yann.lecun.com/exdb/publis/pdf/lecun-iscas-10.pdf.

[5] Suvaditya Mukherjee. *The Annotated ResNet-50.* https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758 [Accessed: (11.22.2023)]. 2022.

[6] Mrinal Tyagi. *Fish Classification ‖ ResNet50 ‖ 100% accuracy.* https://www.kaggle.com/code/tr1gg3rtrash/fish-classification-resnet50-100-accuracy [Accessed: (11.22.2023)]. 2022.

[7] Oguzhan Ulucan, Diclehan Karakaya, and Mehmet Turkan. "A Large-Scale Dataset for Fish Segmentation and Classification". In: *2020 Innovations in Intelligent Systems and Applications Conference (ASYU).* IEEE. 2020, pp. 1–5.

[8] Karen Simonyan & Andrew Zisserman. *Very deep Convolutional Networks for large-scale image recognition.* https://arxiv.org/pdf/1409.1556.pdf. 2015.