UNIVERSITY OF ILLINOIS SPRINGFIELD

Deep Learning Final Project

Topic

Sea Animal Image Classification 🌊

Kiran Ram Ganugula

kganu2@uis.edu

UIN:- 657959396

Prof:- Dr. Elham Buxton

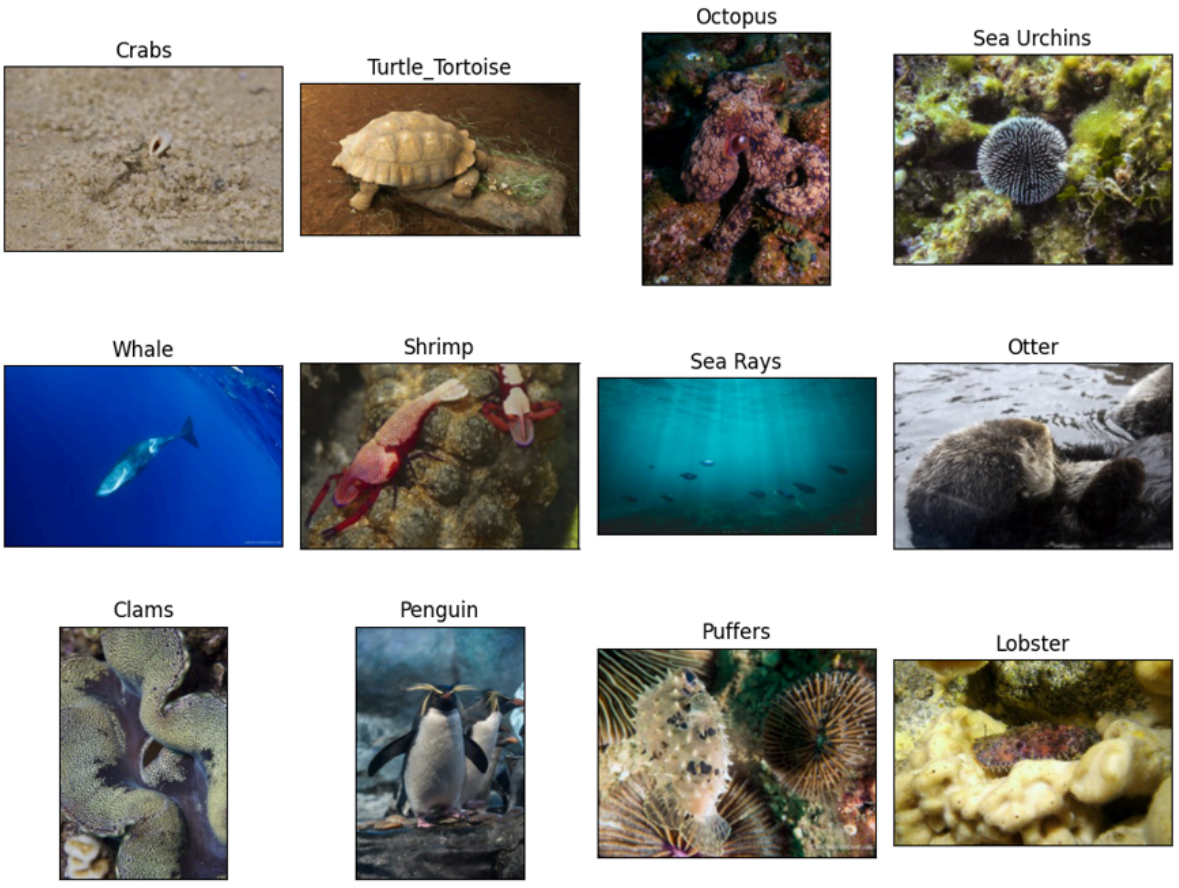
CONTENTS

Summary:-

Aquatic environments have been the birthplace of most life forms, with the oceans providing approximately 90% of the world's living space in terms of volume. The first known vertebrates were fish, exclusively found in water. Some fish evolved into amphibians, capable of surviving both on land and in water for parts of the day. Among amphibians, certain subgroups, including sea turtles, seals, manatees, and whales, further evolved into reptiles and mammals. Plant life, such as kelp and other algae, support some underwater habitats. The ocean's food chain relies on phytoplankton, crucial primary producers.

Aquatic environments are home to a diverse array of organisms, including microscopic phytoplankton, which serve as the base of the ocean food chain. Phytoplankton are important primary producers, using energy from the sun to produce organic matter through photosynthesis. They are an essential food source for a wide range of organisms, from zooplankton to large predators like sharks and whales. And We are in this project to detect the 23 different types of sea animals,using the Neural Network models.

Visualising images from the dataset



About Dataset

The dataset comprises various images of marine animals, sourced from different platforms. Images from pixabay.com are freely available for use without any licensing or attribution requirements. However, images sourced from flickr.com may require attribution to the original authors when used commercially. The dataset presently consists of 23 distinct classes of images, which may be expanded in the future.The images are resized to dimensions of either (300px, n) or (n, 300px), with n representing a pixel size less than 300px.

Variable or Sea AnimalNames:-

- Clams
- Corals
- Crabs
- Dolphin
- Eel
- Fish
- JellyFish
- Lobster
- Nudibranchs
- Octopus
- Otter
- Penguin
- Puffers
- Sea Rays
- Sea Urchins
- Seahorse
- Seal
- Sharks
- Shrimp
- Squid
- Starfish
- Turtle_Tortoise
- Whale

Models Performed

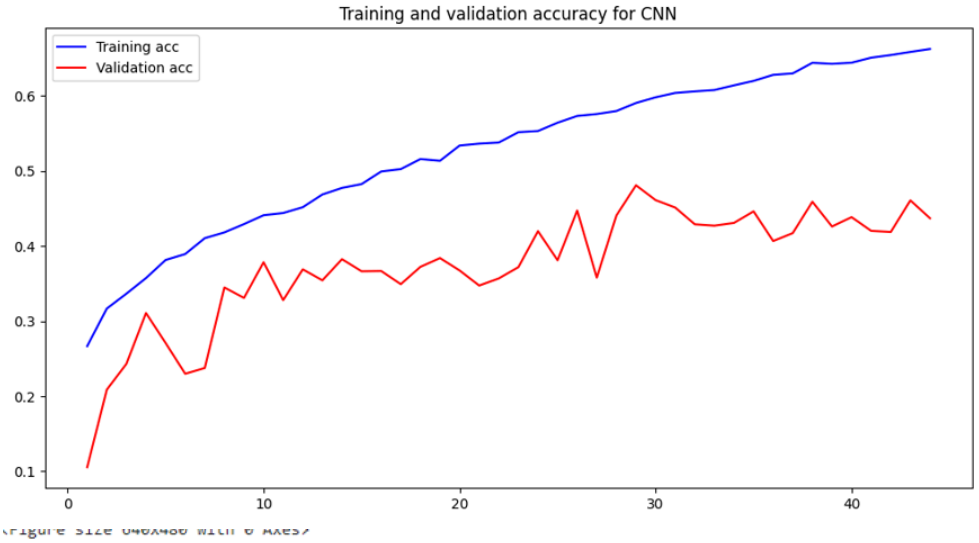
We actually have a large dataset with 12000+ images in which we divided into the train 8000 test 2000 and validation dataset 2000 and good number of images for each variable, so we did not perform the “Data Argumentation” in our dataset.so i directly performed the models on our dataset. Below is the list of Models I did on my project.

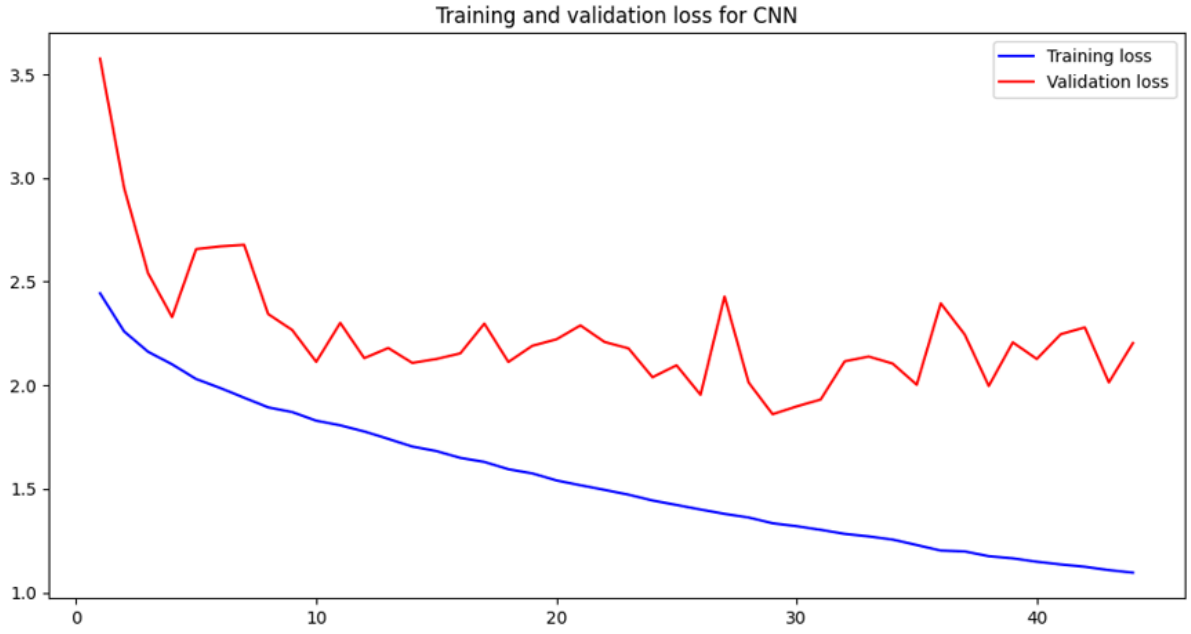
1. **Baseline Model**
2. **Baseline with Multilayer and Dropout layer**
3. **Depth wise separable and Residual Block**
4. **Transfer Learning Using the feature extension (DenseNet169)**
5. **Transfer Learning Using the “ResNet101”**

1. Baseline Model

To begin constructing the CNN model, we start with an input layer that contains information regarding the shape and type of data that the model will be processing. Unlike the Dense layer, the image does not require flattening for the conv2D layer. We incorporated multiple blocks that feature progressively larger filters, each consisting of a convolutional layer, a ReLU activation function, a batch normalisation layer, and a max pooling layer. Following these blocks is a GlobalAveragePooling layer, culminating in an output layer that utilises the softmax activation function.

Output Graph:-

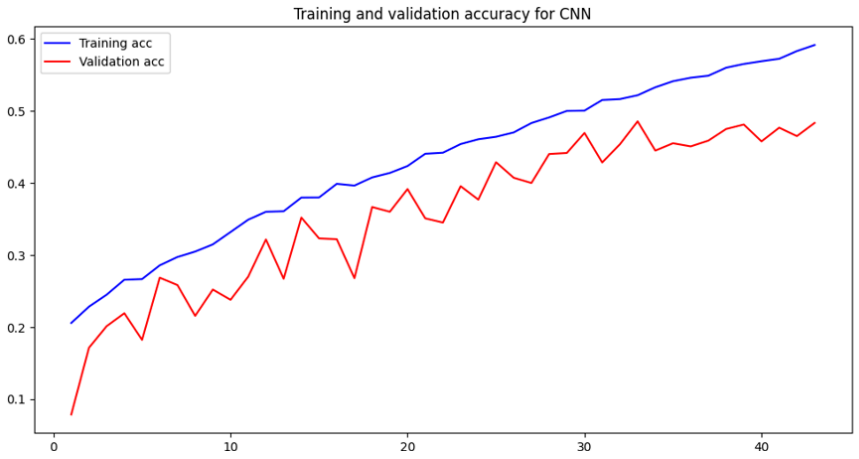


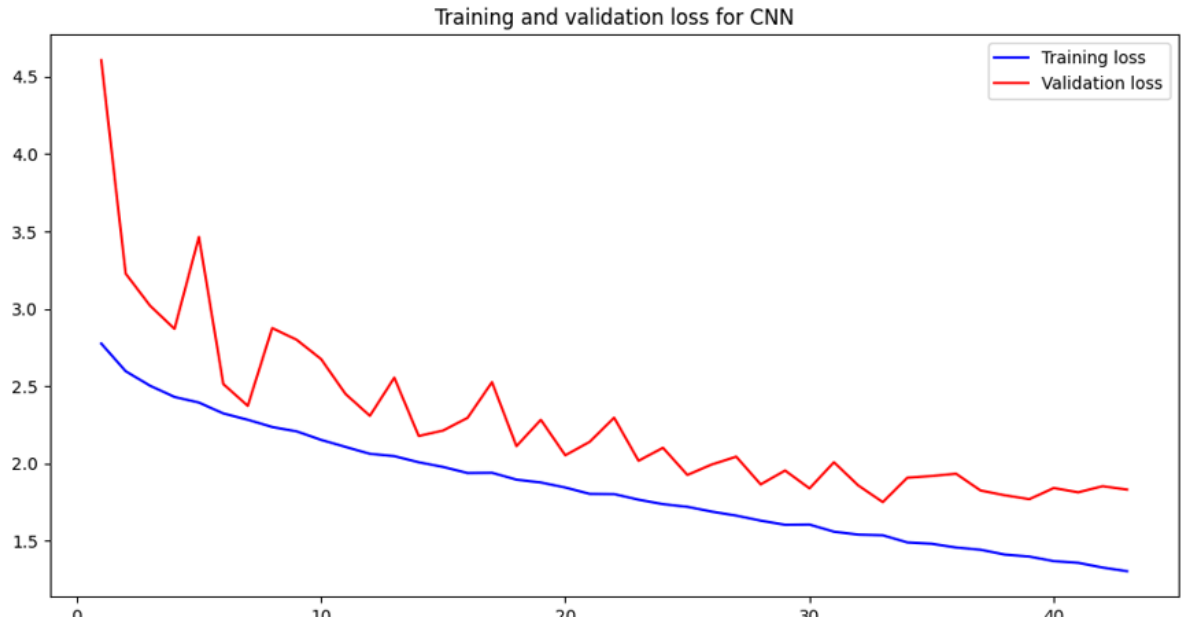


2. Baseline with Multilayer and Dropout Layer

Adding more layers can help extract additional features from the data, but only to a certain extent. Beyond a certain point, adding more layers can lead to overfitting, where the model becomes too closely fitted to the training data and loses its ability to generalise to new data. Overfitting can result in issues such as false positives.Dropout layers are typically more effective when placed between fully connected/Dense layers. In this case, dropout can be applied after the GlobalAveragePooling layer.

Output Graph:-

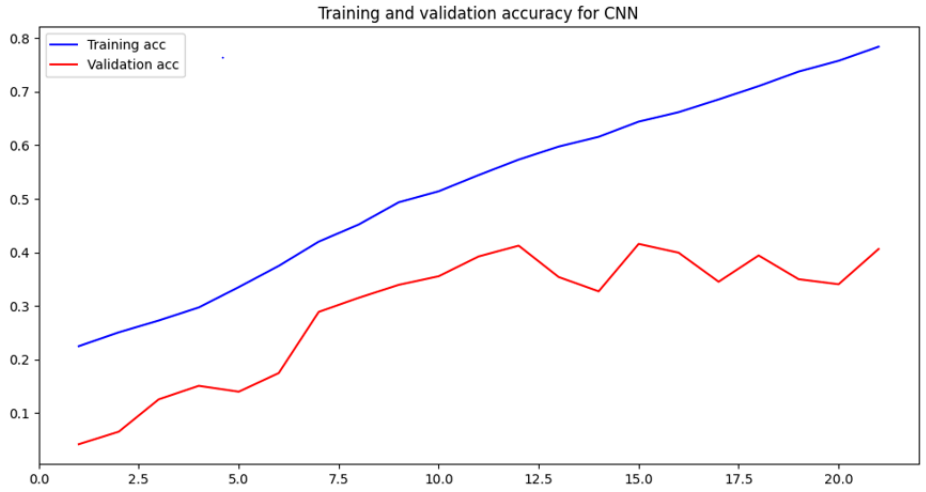




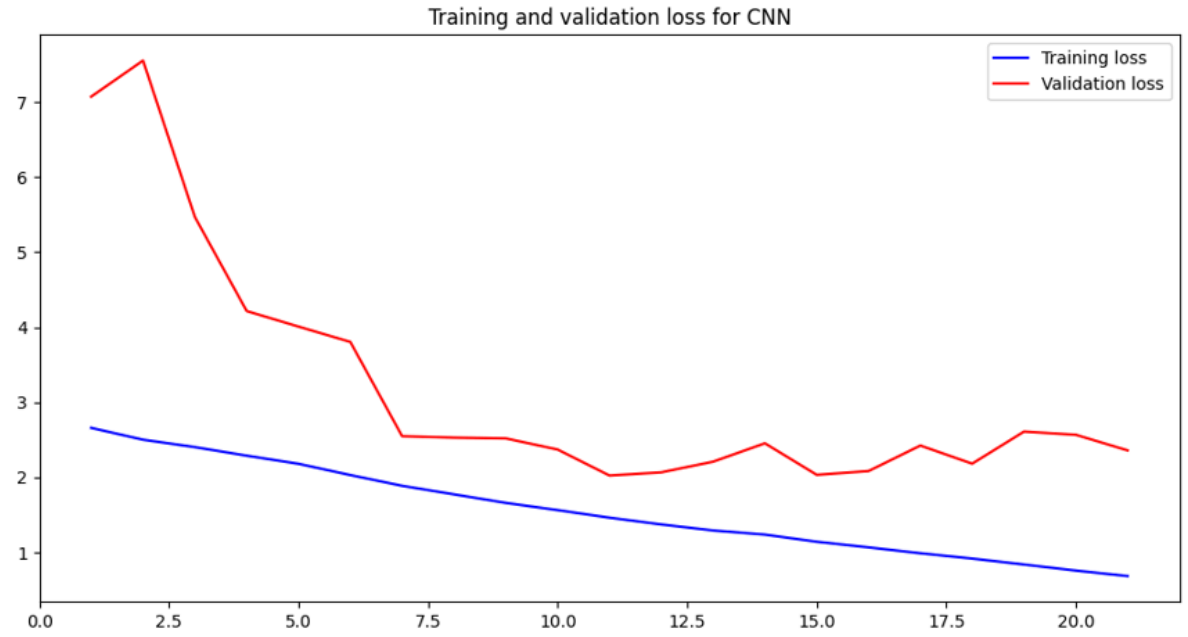
3. Depth wise separable and Residual Block

The ResNet architecture introduced the residual connection technique in 2015, which adds the input of a layer or block of layers back to its output, allowing the model to bypass the block if it increases training error. This residual link acts as a data shortcut around potentially harmful or noisy blocks, enabling error gradient information from early layers to pass through a deep network without noise.

Output Graph:-



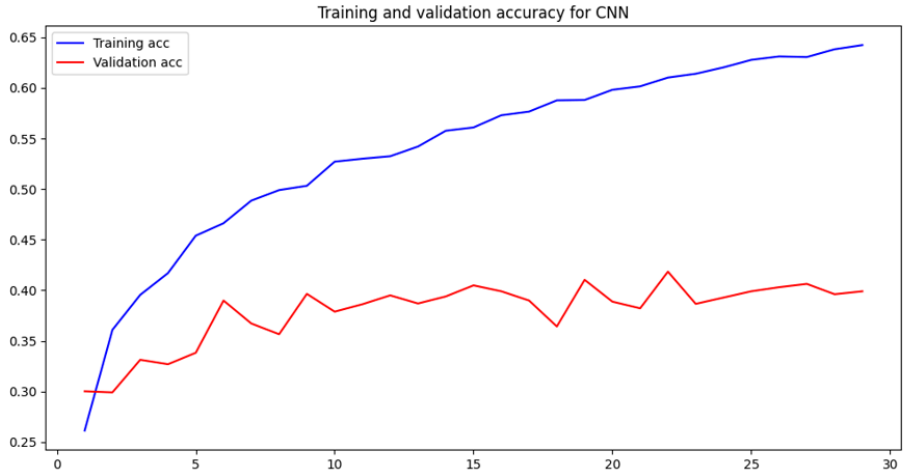
<Figure size 640x480 with 0 Axes>

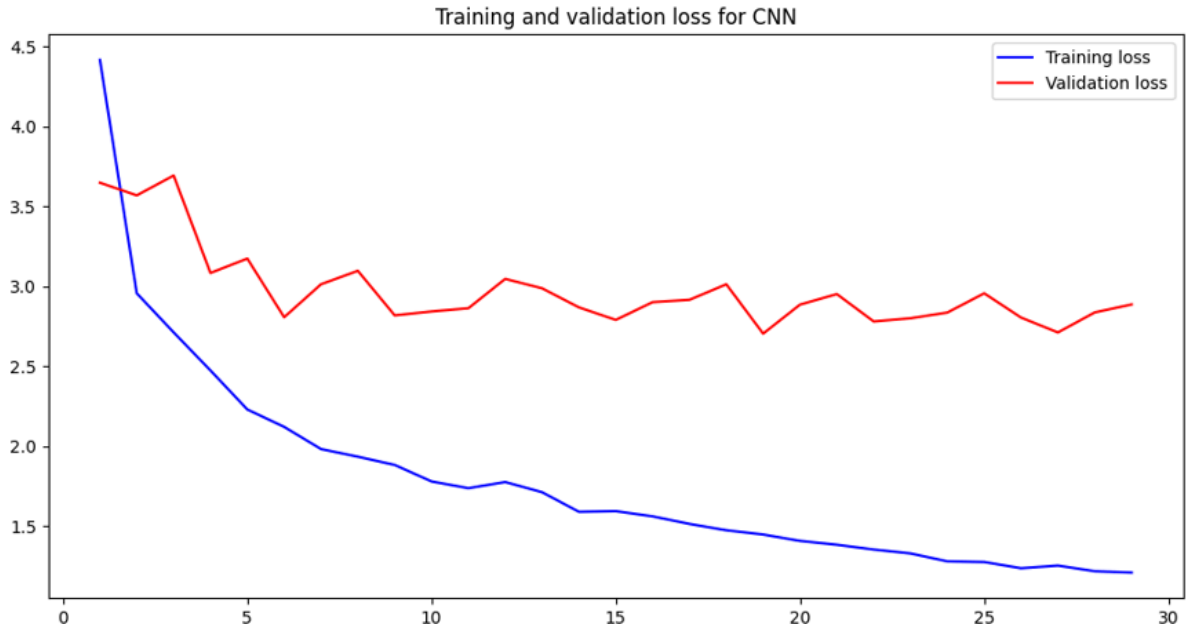


4. Transfer Learning Using the feature extension (DenseNet169)

One common technique for using a pretrained network is feature extraction. This involves taking the convolutional base (i.e., the convolutional layers and possibly batch normalization layers) of a well-known architecture (such as VGG, ResNet, Inception, DenseNet, etc.) that has been pretrained on ImageNet, running the new dataset through it to extract features, and then training a new classifier using these extracted features.

Output Graph:-

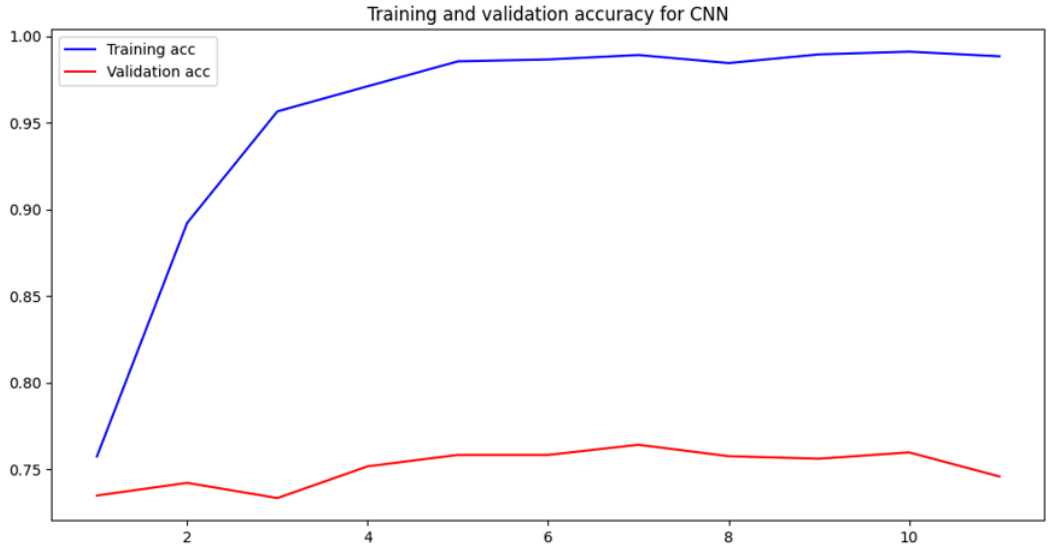




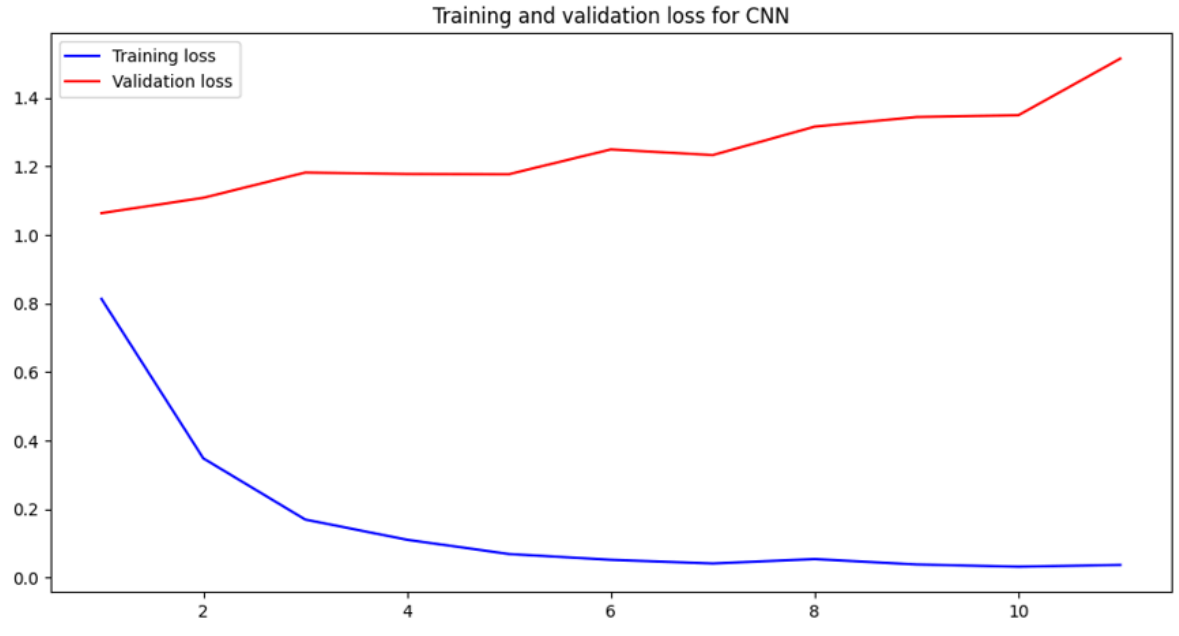
5. Transfer Learning Using the “ResNet101”

Using transfer learning with ResNet101 involves taking the pretrained ResNet101 model and removing the top layers (typically the fully connected layers) that were trained on ImageNet, a large-scale dataset of images with over 1,000 classes. The remaining layers of ResNet101 are then used as a convolutional feature extractor for the new task.

Output Graph:-



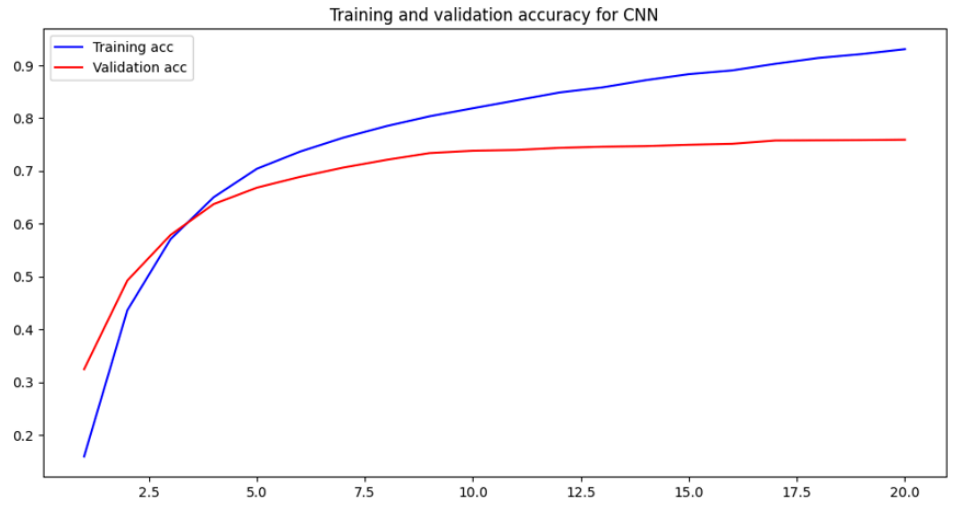
<Figure size 640x480 with 0 Axes>



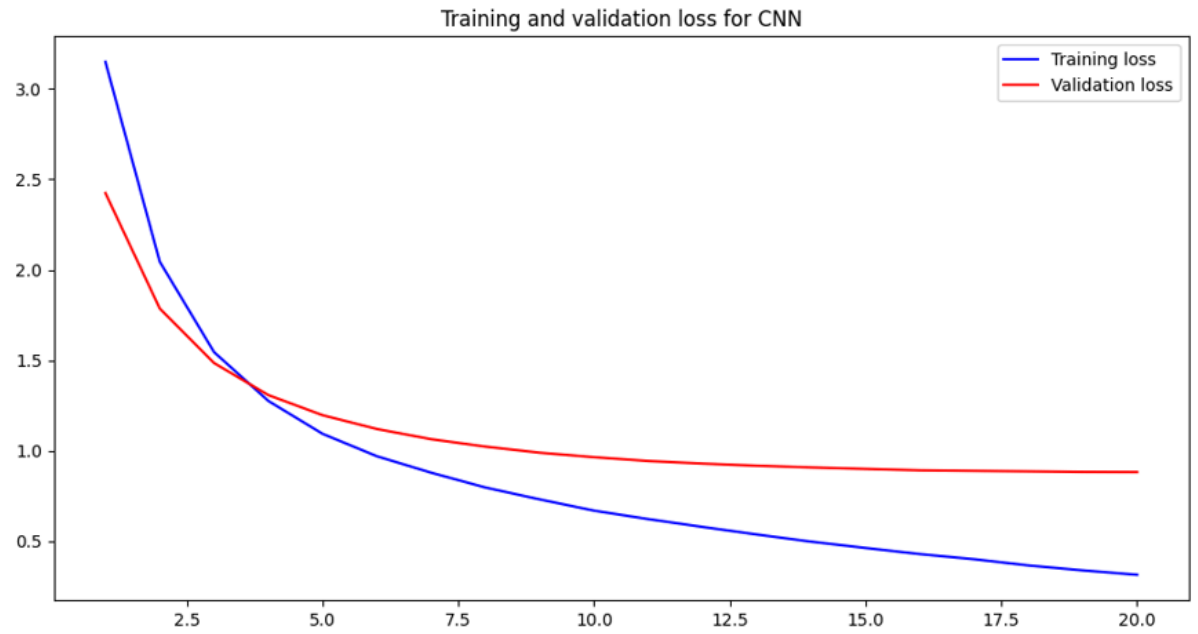
Fine Tuning

this fine-tuning technique allows the pre-trained model to adapt better to the new dataset by leveraging the features learned from the large-scale dataset used for pre-training.To achieve this, the code sets trainable to False for all layers except the last 9 layers of the conv_base network. Then, it sets trainable to True for only the convolutional layers in the last 9 layers.

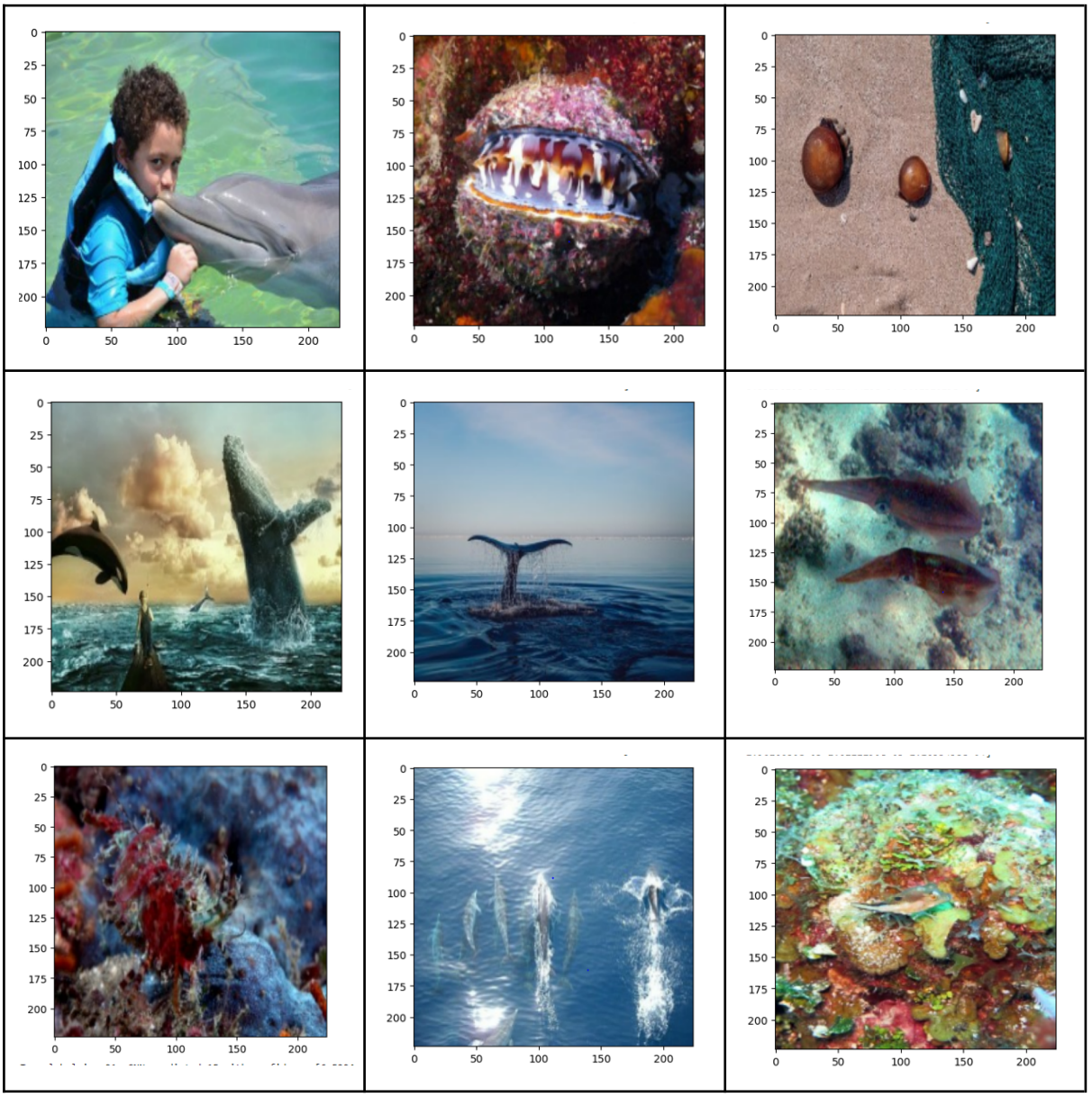
Output Graph:-



<Figure size 640x480 with 0 Axes>



Showing misclassified images in the validation data



Classification Reports and Confusion Matrix

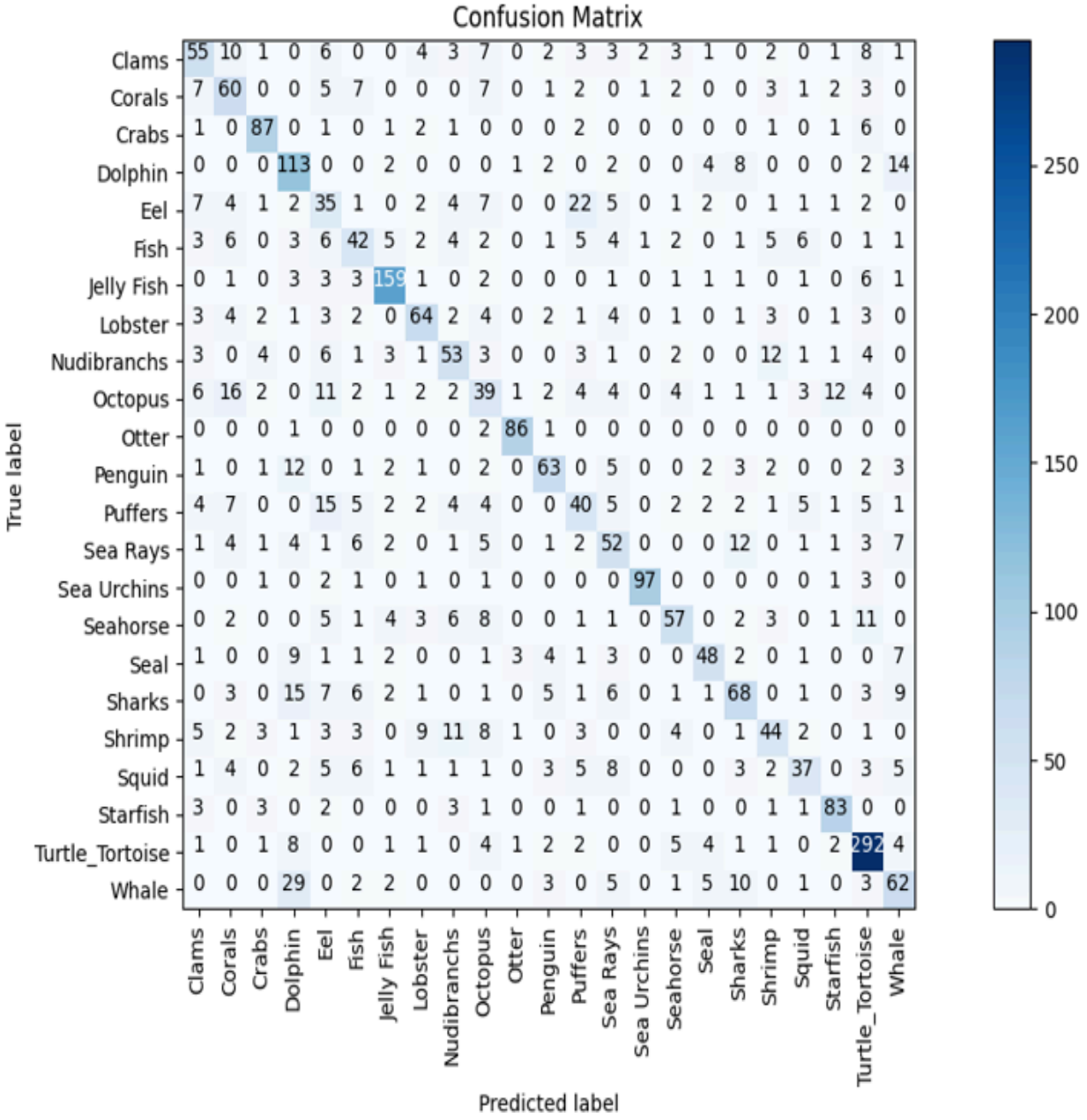
The output is a classification report which provides a summary of the performance of a machine learning model on each class of a multi-class classification problem.

For each class, the classification report shows the precision, recall, f1-score, and support. Precision is the ratio of true positives to the sum of true positives and false positives. Recall is the ratio of true positives to the sum of true positives and false negatives. The f1-score is the harmonic mean of precision and recall. The support is the number of samples in the true class.

The values in the output indicate the performance of the machine learning model on each class. For example, the precision of Clams is 0.54, which means that 54% of the samples predicted to be Clams are actually Clams. The recall of Clams is 0.49, which means that 49% of the actual Clams are correctly identified by the model. The f1-score of Clams is 0.51, which is the harmonic mean of precision and recall. The support of Clams is 112, which is the number of samples in the true class.

Overall, the classification report provides a good summary of the performance of the machine learning model and helps in evaluating the model's ability to classify samples into different classes.

	precision	recall	f1-score	support
Clams	0.539216	0.491071	0.514019	112.000000
Corals	0.487805	0.594059	0.535714	101.000000
Crabs	0.813084	0.844660	0.828571	103.000000
Dolphin	0.556650	0.763514	0.643875	148.000000
Eel	0.299145	0.357143	0.325581	98.000000
Fish	0.466667	0.420000	0.442105	100.000000
Jelly Fish	0.841270	0.864130	0.852547	184.000000
Lobster	0.659794	0.633663	0.646465	101.000000
Nudibranchs	0.557895	0.540816	0.549223	98.000000
Octopus	0.357798	0.330508	0.343612	118.000000
Otter	0.924731	0.955556	0.939891	90.000000
Penguin	0.684783	0.630000	0.656250	100.000000
Puffers	0.408163	0.373832	0.390244	107.000000
Sea Rays	0.477064	0.500000	0.488263	104.000000
Sea Urchins	0.960396	0.906542	0.932692	107.000000
Seahorse	0.655172	0.542857	0.593750	105.000000
Seal	0.676056	0.571429	0.619355	84.000000
Sharks	0.586207	0.523077	0.552846	130.000000
Shrimp	0.536585	0.435644	0.480874	101.000000
Squid	0.596774	0.420455	0.493333	88.000000
Starfish	0.768519	0.838384	0.801932	99.000000
Turtle_Tortoise	0.800000	0.884848	0.840288	330.000000
Whale	0.539130	0.504065	0.521008	123.000000


Confusion Matrix
True label
Predicted label
Clams
Corals
Crabs
Dolphin
Eel
Fish
Jelly Fish
Lobster
Nudibranchs
Octopus
Otter
Penguin
Puffers
Sea Rays
Sea Urchins
Seahorse
Seal
Sharks
Shrimp
Squid
Starfish
Turtle_Tortoise
Whale
0
50
100
150
200
250

True Labels and Predicted Labels

If the predicted label is correct, the title will be displayed in green, and if it's incorrect, the title will be displayed in red.

Conclusion

- Fine Tuning helped our model to improve a lot, it got the accuracy of 93% and validation loss 88%, this is the best output results we got.
- The overall accuracy of the model on the test set is 0.636, which means that it correctly predicts the class label for 63.6% of the instances in the test set.
- Looking at the individual class metrics, it appears that the model performs well on some classes, such as Sea Urchins, Otter, and Turtle_Tortoise, with high precision, recall, and F1-score. On the other hand, the model performs poorly on some classes, such as Eel, Octopus, and Sharks, with low precision, recall, and F1-score.
- The macro average, which is the unweighted mean of precision, recall, and F1-score across all classes, is 0.617 for precision, 0.605 for recall, and 0.608 for F1-score. The weighted average takes into account the support for each class, and is therefore more representative of the overall performance of the model. The weighted average precision, recall, and F1-score are 0.634, 0.636, and 0.632, respectively.

References

https://www.kaggle.com/code/mohameddeeb/sea-animals-cnn-with-scratch-pretrained-models

https://www.kaggle.com/code/vencerlanz09/sea-animals-classification-using-efficeintnetb7

https://www.kaggle.com/datasets/vencerlanz09/sea-animals-image-dataste

Thank You