**Marine Species Classification using convolutional Neural Networks**

***Abstract* —** The building of a model for the classification of marine species is one of the objectives of this research, which may be looked of as an illustration of multilabel classification. The classification of marine species will be accomplished through the development of this model. Python, TensorFlow, and keras are the three programming languages that will be used throughout the course of the process of building the model. keras is also expected to be used. Convolutional neural networks are going to be implemented into the model. In addition to this, the study makes use of data augmentation in order to increase the total amount of data to an adequate level in order for it to be appropriately incorporated into the model for the purpose of training. This ensures that the model can be properly utilised for the purpose of learning. This was done so that participants might gain a better understanding of how to utilise the model in their work. Because of this, it is ensured that the model may be properly utilised for the purpose of instruction in the appropriate method.

***Keywords —*** *Marine Species, Max Pooling, Zero Padding, TensorFlow, Keras, Image Processing, CNN, data augmentation*

# INTRODUCTION TO PROBLEM

Marine lif­e or sea life refers to the life under the water bodies like sea and oceans. Similar to the life on land, the marine body has its own ecosystem under the ocean. This means that they have their own food chain and Ib. The flora and fauna of the marine ecosystem is very rich. A massive number of 2,26,000 have been identified and described so far and on an average about 1200 new species are identified every year. The identification of marine life is very crucial for their study and to understand about the marine ecosystem. The variety and the range of the marine ecosystem is very high when compared to the ecosystem on land. This is because the marine life was the first form of life on earth years before the life on land started. Discovery in the marine provides us with so many information about the life before us. Even though on one hand the marine ecosystem is a rich ecosystem, unfortunately on the other hand, the life of marine animals has not been good in recent years. This is because of the pollution that is caused by the humans. Marine biologists are people who study about the marine ecosystem. Their work is to study the system and contribute to the enrichment of the ecosystem. For many years the research in this system has been taking place. HoIver, one major constraint in the marine ecosystem is that the data availability of the marine ecosystem is very low when compared to other ecosystems. This is one of the greatest barriers that the marine biologists face. Due to this there is a need for advanced technology that helps them in conducting the research. Even though various kinds of research take place, **correct identification of a species is the most basic step for the research**. So, there is a **need for an instantaneous classification and identification of marine species** that can assist the marine biologists, researchers, and students and help them to quickly identify them, conduct studies, and research. Machine learning, Image Processing and deep learning is the perfect field that can make this possible where a model is developed and deployed. A model is created by training it on a predefined dataset, validating it, using it for testing, and then deploying it to perform the task at hand. Convolution neural networks are one of the popular deep learning algorithms used for multilabel classification, despite the fact that there are many different classification techniques available. I preprocess every image in the dataset and use Python utilities to create a CNN model.. **The classifier is built to classify four differ marine species namely starfish, sea turtle, crab and dolphin.**

# OBJECTIVE

The main objective of the project is to build a multilabel classification model using CNN and to improve the accuracy of the model through augmenting enough data and by varying the activation functions and optimizing techniques. The paper also aims to study about the changes in the accuracy, precision and recall when the hyper parameters like number of epochs, data size and activation function changes.

# RELATED WORKS

[1] The paper contained a thorough literature assessment of deep learning methods for identifying marine species. The primary focus of the study is image categorization for marine species because it serves as the foundation for other visual tasks and serves as the primary tool for mapping the distribution of marine species. As the foundation of picture classification, methods for feature extraction and feature combination that are specific to marine species have been discussed in detail. The advantages and disadvantages of both handmade and deep features for marine organisms have been considered. Moreover, results for marine species detection on movies have been studied in this chapter. The computer vision area has been given a fantastic opportunity, as well as a significant problem, by the increasing quantity and distinctive properties of marine data, such as colour degradation and significant morphological variety. It also makes the case that deep learning-based picture quality enhancement and semantic segmentation approaches can be used in the future to address issues with maritime data, such as poor image quality and confusing borders. [2] Two species of bacteria with different cell shapes—Staphlococcus aureus, which has a spherical or round shape, and Lactobacillus delbrueckii, which has a long-rod shape—were found to be capable of making automatic predictions using machine learning, image classification, and deep learning techniques. For accuracy in the bacterium prediction use case, the experimental findings compare outcomes utilising an existing research dataset versus a dataset that was produced on one's own. By using more training cycles of 4 Epochs and both high-resolution and standard resolution bacterium images, training accuracy and validation accuracy have increased to more than 75%. As a result, this research also aims to alter LeNET technique and track the development of additional training with more Epochs. The findings have demonstrated that future accuracy improvements for datasets of standard resolution bacterium photos are possible. It can be used to compare CNN methodologies like ResNET, AlexNET, and more. However, the scope of this initial investigation is simply two species of bacteria with various cell morphologies. [3] Image recognition is a crucial component of artificial intelligence, which has taken on a greater significance in recent years in the field of computer science research. Four convolutional layers and two common dense layers are used in the convolutional neural network model. The convolution layers also use the pooling technique, which may be applied to statistically lower the feature dimension with modest convolutional effects, eliminating time-consuming calculations with a high level of picture recognition. This model can mimic the neural network in the human brain and, with some learning, can output the results of picture recognition instantly. However, it has some restrictions and needs more in-depth research because it can only handle the static images in the format that has been supplied. [4] I suggested a straightforward Convolutional neural network for image categorization in this paper. Less computational work is required by this straightforward convolutional neural network. On the basis of the convolutional neural network, I also examined various approaches to choosing the learning rate and other optimization techniques for determining the best parameters that have an impact on picture categorization. Additionally, I confirm that the shallow network has a respectably strong recognition impact. The recognition rate of each method has increased with the amount of iterations, and multistep is the best among those algorithms. With more repetitions, SGD can therefore lower the mistake, but the declining speed and effect are universal. Although SGD with momentum and NAG are superior to SGD, the gradient decreases slowly in the initial stages before gradually stabilising with an increase in the number of iterations. SGD with momentum also outperforms NAG. The effect is better, the curve is smooth, and the error rate of the test set is declining rather steadily in ADAGRAD. [5] The deep learning system described in this paper categorises buildings and other objects in high-resolution multi-spectral satellite data. The system is made up of an ensemble of CNNs with post-processing neural networks that merge satellite metadata with the predictions from the CNNs. The system obtains an accuracy of 0.83 and an F1 score of 0.797 using the IARPA fMoW dataset, which consists of one million photos divided into 63 classes, including the false detection class. In the fMoW TopCoder competition, it outperforms the Johns Hopkins APL model by 4.3% and classifies 15 classes with an accuracy of 95% or higher. Our technology might search through a significant amount of satellite imagery for objects or facilities of interest when used in conjunction with a detecting component. It might be able to address the issues raised at the outset of this paper in this way. By keeping an eye on a collection of satellite images, law enforcement authorities could spot unauthorised mining operations or fishing boats, disaster relief teams could map the damage from hurricanes or mudslides, and investors could more efficiently track crop growth or oil well development. This system received a score of 722,985 points in the fMoW TopCoder competition, translating to an Iweighted F1 score of 0.723. With a score of 765,663 points and an Iweighted F1 score of 0.766, our system surpassed the APL system by a margin of 4.3%. Yet rather of an ensemble, their baseline employed a single model. It is possible that an LSTM coupled with our system would demonstrate a similar improvement in accuracy given that their use of an LSTM increased the accuracy by 1% over the CNN/metadata model. [6] This survey offers several Data Augmentation solutions to the issue of Deep Learning models overfitting owing to a lack of data. Big data is used by deep learning algorithms to prevent overfitting. The advantages of big data in the restricted data realm can be obtained by artificially inflating datasets using the techniques covered in this survey. Building better datasets can be done using a process called data augmentation. Several augmentations that fall within the categories of data warping or oversampling have been proposed. Data augmentation has a highly promising future. The potential for using search algorithms that combine data warping and oversampling techniques is immense. Deep neural networks' layered architecture offers numerous opportunities for data augmentation. The input layer is where the majority of the augmentations surveyed function. The space of intermediate representations and the label space, however, are under-explored regions of data augmentation with intriguing outcomes. Some are derived from hidden layer representations, and one way. While many of these methods and ideas can be applied to other data domains, this survey concentrates on applications for picture data. A small dataset's biases cannot all be eliminated by data augmentation. No augmentation technique, from Sample Pairing to Auto Augment to GANs, will produce a golden retriever, for instance, in a dog breed classification task where there are only bulldogs and no examples of golden retrievers. However, a number of biases, including those caused by lighting, occlusion, scale, backdrop, and many others, can be avoided or at the very least significantly reduced by data augmentation. Access to vast data generally makes overfitting less of a problem. Data augmentation transforms small datasets into those of big data to prevent overfitting.

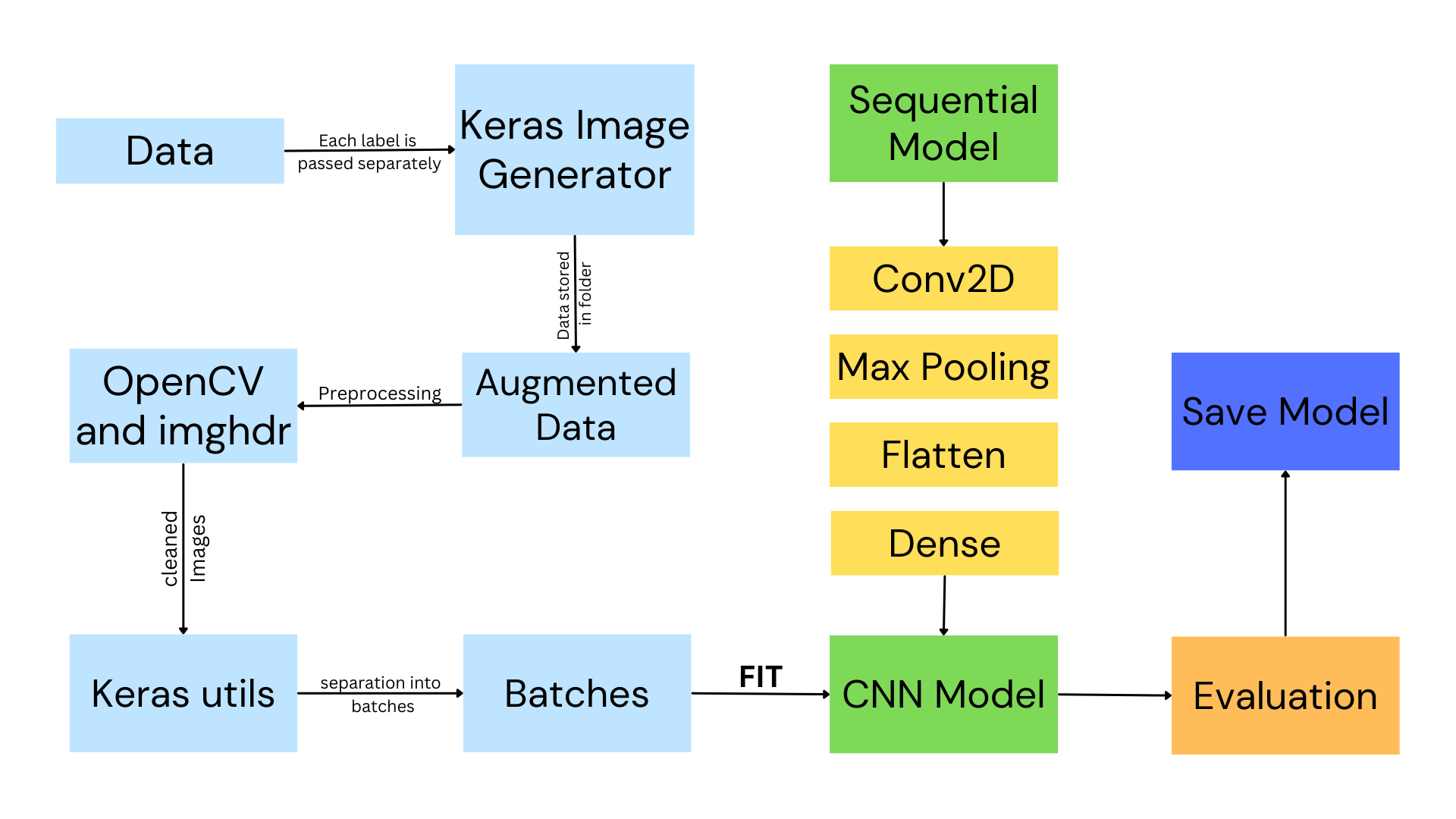
# PROBLEM DEFINITION

The problem that is dealt with in the paper is the instantaneous classification of marine species for study and research purposes. This comes under the problem of multi label classification. The accuracy of such existing classification models is low and have a scope to be improved. I have to analyze the consequences by changing the hyper parameters to achieve better accuracy.

# PROPOSED SYSTEM

I propose a deep learning model based on convolution neural networks. The model is built in python by making use of the python libraries like TensorFlow, Keras and OpenCV. I obtained a dataset from Kaggle that has four directories for four different species namely starfish, sea turtle, dolphin and crab. Before processing of the data, the images files are preprocessed. The preprocessing would include the removal of corrupted images using OpenCV and imghdr. Then the images are scaled. The data is taken and data augmentation is done using Keras image generator. The batch sizes and iterations are changed to obtain a different number of images. Then I separate the dataset into several batches of numpy arrays using Keras. These batches are further separated into train, test and validation sets. Now I **build a sequential model with several conv2D and max pooling layers**. Then this is folloId by the flatten and dense layers. Then the training images are it. The accuracy changes for different data sizes are analyzed to find the optimum data size required. Similarly, all the other hyper parameters like activation function, number of epochs, optimization parameter and kernel size of the convolution neural networks are changed to notice the changes in accuracy, precision and recall.

# BLOCK DIAGRAM



**Figure 1. Block Diagram**

# TOOLS USED

Python is the programming language used. Python libraries used are OpenCV, TensorFlow, Keras, imghdr, matplotlib, OS. VS code is the code editor used along with the jupyter notebook extension. Keras Image generator is used for data augmentation and keras utilities are used for separating the data into batches.

# TECHNIQUES USED

Initially the dataset is taken and augmented using the image generator in Keras that gives us the desired number of images by making small changes in terms of angles, coloring, aspect ratio etc. in the already available images. Keras is used to obtain the images and convert them into batches of NumPy arrays. Now the arrays are scaled to values betIen 0 and 1. Once the data is ready, I build the model using TensorFlow and Keras. I use the sequential model to add various convolution and max pooling layers to the model. I specify various kernel sizes and activation functions to study the accuracy changes. The optimizer is also changed during model compilation. I then split the data into three different sets namely train, validation and test. The training batches fit the model for various epochs. The trained model is then used to test and evaluate the accuracy.

# RESULTS

Different sizes of training data Ire fit in the model with tInty epochs and rmsprop optimizer. The importance of data augmentation was clearly reflected in the accuracy changes. Given that the dataset size was of 400 images, augmentation of the data improved the accuracy in a tremendous way. Table 1 shows the difference in precision and recall values as the data size changes.

**Table 1. Precision and Recall values for differing data size**

|  |  |  |
| --- | --- | --- |
| **Dataset Size** | **Precision** | **Recall** |
| 400 | 0.987 | 0.979 |
| 1500 | 0.991 | 0.983 |
| 3000 | 0.990 | 0.920 |
| 9000 | 0.900 | 0.986 |
| **20000** | **0.996** | **0.993** |

I can see the precision and recall values are consistent and above 90% for all the data sizes. This shows us the proper fitting of the models.

Table 2 shows the different parameters used in the model for analyzing the accuracy changes.

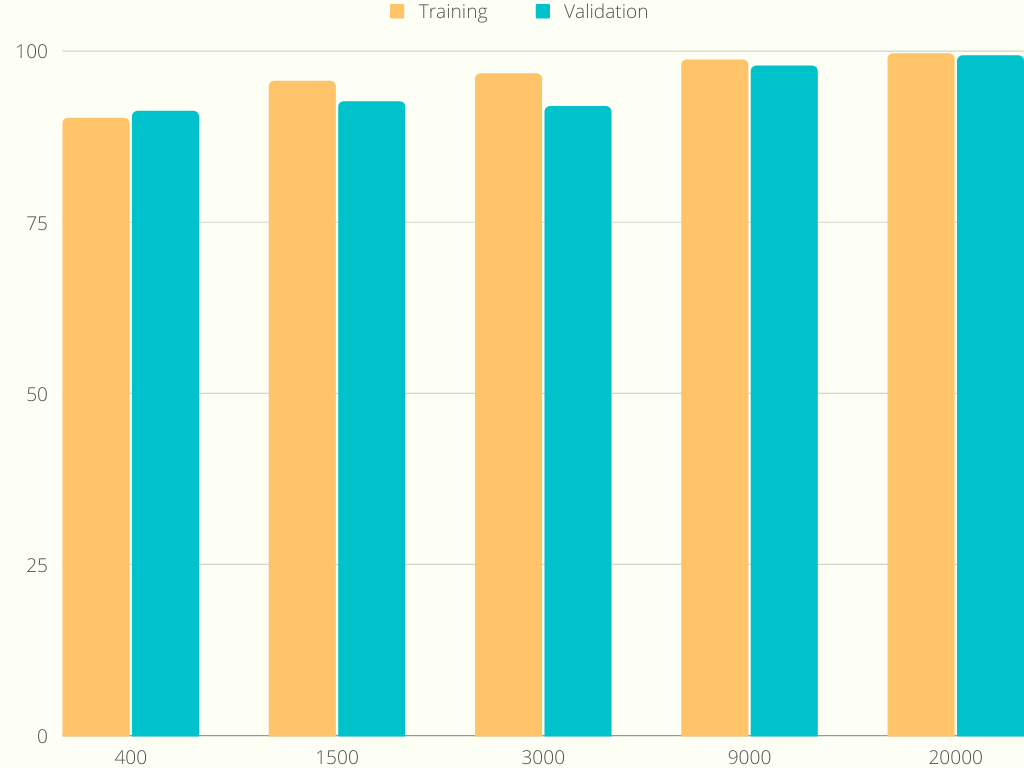
**Table 2. Parameters used for finding the best accuracy**

|  |  |
| --- | --- |
| **Parameter Name** | **Parameters** |
| Kernel Size | 3x3, 5x5, 7x7 |
| Data size | 400, 1200, 3000, 9000, 20000 |
| Activation Function | Relu, Selu, Sigmoid |
| Epochs | 5, 10, 20, 30 |

Table 3 shows the changes in test accuracy with increasing epochs. I can clearly see that the highest accuracy is obtained when epochs is tInty. When I increase the epoch more than that the accuracy tends to become less due to overfitting.

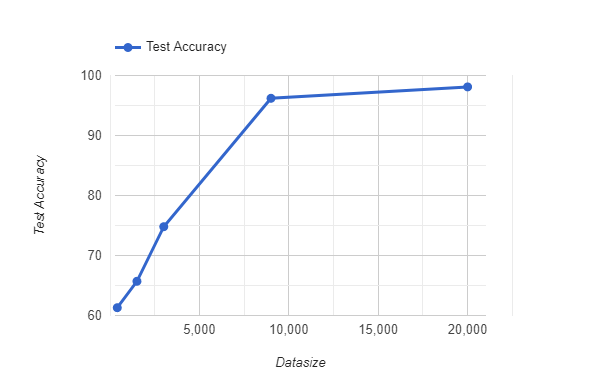
**Table 3. Test accuracy and epochs**

|  |  |
| --- | --- |
| **Epochs** | **Test Accuracy** |
| 5 | 49.7 |
| 10 | 68.1 |
| **20** | **98.1** |
| 30 | 88.3 |



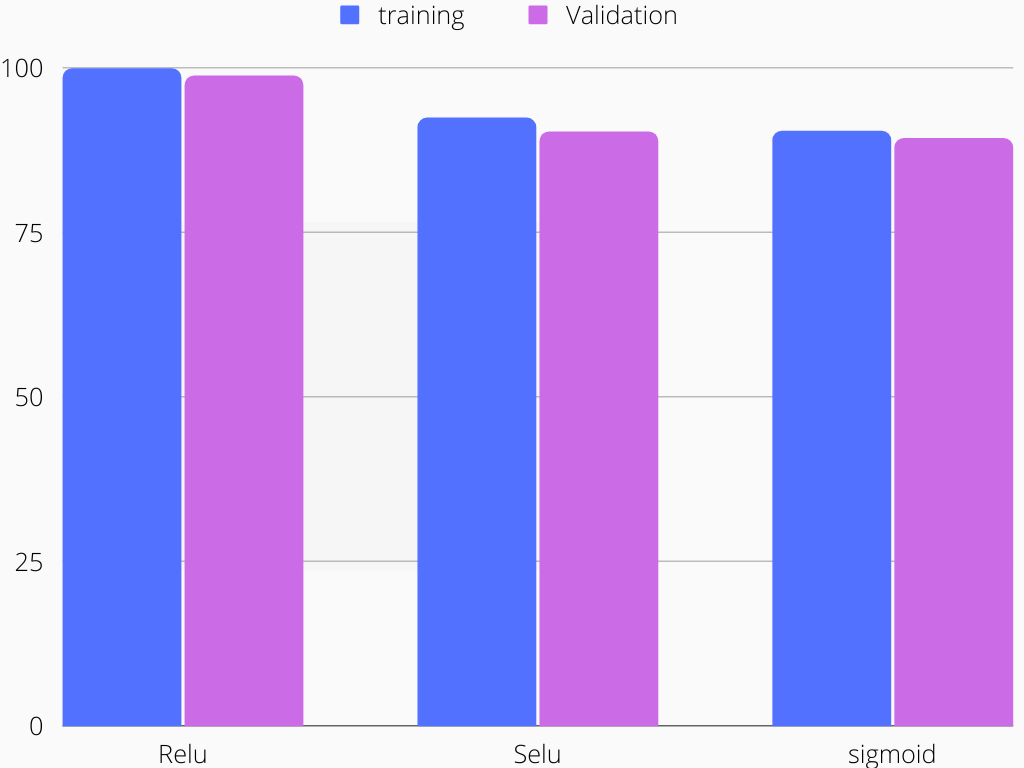
**Fig 2. Training and Validation accuracy**

Fig 2 shows the changes in the training and validation accuracy for different data sizes and I can notice that both these accuracies have been good and above 90% for all the data sizes.

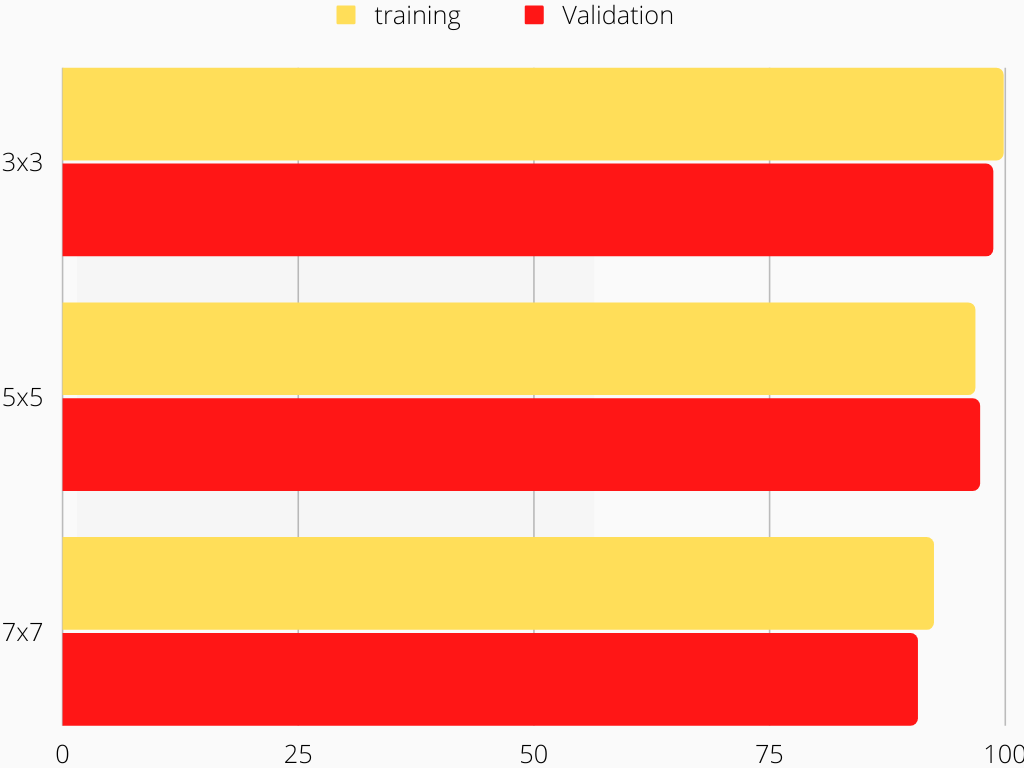


**Fig 3. Test accuracy**

Fig 3 shows the test accuracy for varying data sizes. I can see that the accuracy for loIr data sizes are very low and about 60 to 70 percent whereas as I increase the data size, the accuracy is as high as 98 percent. This figure clearly demonstrates the importance of data size and data augmentation in deep learning classification.



**Fig 4. Accuracy for different activation functions**

Fig 4 shows us that even though the change in then accuracy is not too much for different activation function, the Relu activation function gives us the highest accuracy required. 

**Fig 5. Kernel size and accuracy**

Three different kernel sizes 3x3 5x5 and 7x7 are used to build the model and 3x3 is more suited for the model as shown in fig 5.

# SOURCE CODE

# STEPS TO USE THE PROGRAM

1. Download/run the code from the google colab repository
2. Navigate to the respective folder in PC and create 4 seperate folders named "logs", "models", "data" and "augmented". The model will be saved in the "models" folder and logs in the "logs" folder.
3. Download the data from the dataset and put them in the "data" folder by creating subfolders with their label name. For eg: in the data folder create a folder called "crab" and inside the crab folder create another folder by the same name "crab" and put all the downloaded images of crab in here. Repeat the similar process for all the other labels like starfish, dolphin etc.

**NOTE: The data augmentation program would work properly only if this is done properly**.

1. Once after the data folder is ready with all the data files go to the augmented folder and create separate folders for separate labels, for eg: one folder named crab, one named dolphin etc.

**NOTE: It is the data in the augmented folder that is used as the main data for training and testing the model**.

1. Now we can successfully run the augmentation by just changing the number of iterations to get the desired number of data generated.

Following is the google colab link for the working model,

<https://colab.research.google.com/drive/1GS2ouHnu5QN34irJuOTZkr-xSsPA42LJ?usp=sharing>

# DATASET USED

CNN is used in conjunction with keras and tensorflow to classify four different marine species labels. Data augmentation is done using keras image data generator Dataset Link :

<https://www.kaggle.com/datasets/andrea2727/dataset-of-aquatic-animals>

# CONCLUSION

A multi label deep learning classification model was built and tested using various parameter changes to record the results in terms of accuracy, precision and recall to find the best model parameters to achieve the highest accuracy. The accuracy was the highest when I used a dataset of size 20000 images belonging to four different classes. The kernel size was (3,3) and Relu activation function is used for convolution and max pooling layers. The Softmax activation is the most accurate activation function to use when attempting to forecast the likelihood of each data point in relation to the several classes. rmsprop optimiser is what's being used here. The ratings for precision and recall range from 0.98 to 1.0 respectively. It is possible to acquire a training accuracy of 99.8 percent and a validation accuracy of 98.7 percent. The accuracy of the test came out to be approximately 98%.

# FUTURE WORKS

The potential for multilabel categorization by means of deep learning is really high, and the model that I have constructed has a very significant amount of room for advancement. The accuracy of the model, along with its other assessment parameters, has the potential to be enhanced in the future thanks to developments in deep learning libraries and packages, as well as to the introduction of new parameters and methods.

##### **REFERENCES**

[1] Xu, Lian & Bennamoun, Mohammed & An, Senjian & Sohel, Ferdous & Boussaid, Farid. (2019). Deep Learning for Marine Species Recognition: Proceedings of International Conference on Computational Intelligence 2015. 10.1007/978-3-030-11479-4\_7.

[2] T. Treebupachatsakul and S. Poomrittigul, "Bacteria Classification using Image Processing and Deep learning," 2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), 2019, pp. 1-3, doi: 10.1109/ITC-CSCC.2019.8793320.

[3] Li, A., Li, Y., & Li, X. (2017). TensorFlow and Keras-based Convolutional Li, A., Li, Y., & Li, X. (2017). TensorFlow and Keras-based Convolutional Neural Network in CAT Image Recognition. *DEStech Transactions on Computer Science and Engineering*.Neural Network in CAT Image Recognition. *DEStech Transactions on Computer Science and Engineering*.

[4] T. Guo, J. Dong, H. Li and Y. Gao, "Simple convolutional neural network on image classification," 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA), 2017, pp. 721-724, doi: 10.1109/ICBDA.2017.8078730.

[5] Pritt, M.D., & Chern, G. (2017). Satellite Image Classification with Deep Learning. *2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, 1-7.

[6] Shorten, Connor & Khoshgoftaar, Taghi. (2019). A survey on Image Data Augmentation for Deep Learning. Journal of Big Data. 6. 10.1186/s40537-019-0197-0.