International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Fish Classification Using Deep Learning on Small Scale and Low-Quality Images

M. Sudhakara*[1], M. Janaki Meena[2], K. Reddy Madhavi[3], P. Anjaiah[4], LNC Prakash K[5]

Submitted: 06/06/2022 **Accepted:** 10/09/2022

Abstract: Fine-grained visual classification is one of the essential data science tasks with enormous datasets. The studies show that species composition and ample distribution of fishes notably impact the fishery industry, aquaculture, and marine ecosystem. Incredible work and analysis are required to state fish characteristics by classification. Lately, deep learning has helped to gain exceptional development in this area. Be that as it may, fine-grained fish classification is more complex than primary image classification, particularly with medium quality (i.e., underwater images) and small-scale (i.e., limited data). But traditional convolutional neural networks (CNNs) and other popular models like V.G.G., RESNET, DenseNet, etc., require high-quality and high-scale datasets. This paper presents another way to enhance the CNN models that best fit this fine-grained fish classification problem. Real-world underwater images have several issues, including noise, dominant colours, light attenuation, etc. Further, it isn't easy to get a large set of images of each category of species under the sea, and hence an imbalanced dataset is generated. These two problems are addressed in this paper. Then the quality of the raw images was improved by an Underwater Image Enhanced Generative Adversarial Network (UIEGAN), that CycleGAN trains over 6128 images of the ImageNet dataset. Conventional data augmentation helps increase the dataset size of the dataset by random transformations of the images (i.e., flipping, rotation), but it cannot handle the imbalanced class problem. We generated synthetic images of every class utilizing DCGAN to create a balanced dataset. Further, we used the SmallerVGG and SmallerRESNET models that best fit the Croatian dataset. Moreover, we compared our strategy with eight popular pre-trained transfer learning models trained on the ImageNet dataset. The exploratory outcomes show that the proposed techniques beat well-known CNNs, with high accuracy, demonstrating their possible applications in the real-time underwater fish image classification.

Keywords: *Fish classification, CycleGAN, DCGAN, SmallerVGG, SmallerRESNET.*

1. Introduction

In recent years, people have better explored the ocean due to technological advancement. Due to the continuous utilization of inadequate resources of the sea, biodiversity, mainly fish variety under the marine environment, is under threat. Hence, productive strategies must be proposed to find and estimate quantitative fish distribution. For example, fine-grained fish classification gives a better environment to fish and marine ecology [1]. There is a significant demand for tourists to find and observe underwater fish species in pond water. Besides, scientists and marine biologists need to keep track of different species of fish's behaviour. Business applications like fish cultivation rely on monitoring the fish species breeding similar fishes, and studying their life cycle. The classification of fish species is turning into a challenge for research. A few methodologies are employed for non-destructive sampling, automatic fish identification, and types characterization in underwater videos [2]. Nevertheless, challenges presented by variations regarding terrible light conditions, murkiness in water, occlusions, intra, and inter-species similarity, moving aquatic plants, and background confusion minimize utilization of those procedures in real-life situations because of low preciseness.

In the literature, there exist methodologies for fish detection and classification. Along with this, most of the research on fish detection is done based on pre-trained YOLO object detection models. For instance, in fish detection in [3], researchers have introduced live fish tracking framework utilizing YOLO and correlation filters and added detection and classification in an end-to-end manner. Like this, the authors in [4] prepared a YOLO model to distinguish the fish species with multivariate datasets, acquiring an average precision score of 0.5392. In [5], the authors elaborated their work to incorporate sea mammals like fish and utilized

[1] *Research Scholar, SCOPE, VITCC, Chennai, India*
[2] *Professor, SCOPE, VITCC, Chennai, India.*
[3] *Professor, C.S.E., Sree Vidyanikethan Engineering College, Tirupati*
[4]*Assistant Professor, Department of C.S.E, Institute of Aeronautical Engineering, Hyderabad, India*
[5]*Associate Professor, C.S.E. Department, CVR Engineering College, Mangalpalli, Hyderabad.*
**malla.sudhakara2015@vit.ac.in*, janakimeena.m@vit.ac.in, kreddymadhavi@gmail.com, anjaiah.pole@gmail.com
klnc.prakash@gmail.com
*Corresponding Author: *malla.sudhakara2015@vit.ac.in*

similar YOLO procedures. The common thing for these methodologies is that they trained their model completely. In [6], the authors proposed an alternate technique, specifically different deep learning-based methods for temperate fish identification and characterization. The authors used input images and videos captured from underwater cameras in typical habitats and utilized YOLOv3 for fish detection and squeeze-excitation networks (SE-net) for classification.

The other fish detection and classification categories use deep learning models like convolutional neural networks (CNN). Meng et al. [7] designed an underwater drone with a camera, which utilized LeNet [8], AlexNet [9], and GoogleNet [10] for fish identification. Qin et al. [11] designed a model composing convolutional and pooling layers and a linear SVM classifier for fish detection. Nonetheless, the previously mentioned works neglect that fish image classification is a fine-grained problem, which requires a human-labelled and large-scale fish dataset. Fine-grained image classification (FGIC), like fish, cat, bird, and flower species, remains challenging compared to conventional image classification due to the high intraclass and low inter-class similarities. In [12], the authors proposed another technique to improve transfer learning and squeeze and excitation networks to classify fine-grained fish images with better quality and small-scale datasets. The authors utilized the super-resolution reconstruction method for better image resolution and pre-pre trains, pre-trains analyzed domain knowledge and fine-tuned with some professional skills.
Moreover, refined squeeze with excitation blocks is intended to enhance bilinear CNNs for fine-grained classification. In [13], the researchers provide a compound way by combining optical flow and the Gaussian Mixture Model (G.M.M.) with the YOLO model in a dynamic environment rather than fish detection in static images. The authors in [14] also worked on FGIC, using Mask-CNN. The method localizes the object's parts and selects the descriptors for fine-grained bird classification on a large scale and high-quality datasets. Small-scale and low-quality FGIC remains challenging because CNNs can't create proficient skills from a limited set of images. Image distortion happens when pictures have lousy quality. For example, Figure 1 is a sample image of 12 fish categories from the Croatian fish dataset [15].

Fig. 1. Samples of different categories of Croatian datasets with high intra-class and low inter-class similarities.

Henceforth, it is hard to get better classification results with low-quality images. Nevertheless, many FGIG tasks in the real world frequently experience a limited number of poor-quality images. Not the same as the past works, this work introduces a strategy that ideally concentrates on addressing those issues. This paper primary contributes the following:

1. High-quality dataset through generative adversarial network design (like UIEGAN) trained upon underwater image dataset.
2. We have generated synthetic images by another deep convolutional GAN (DCGAN) along with conventional data augmentation to address a limited dataset. Step 2 also addresses the problem of class imbalance issue.
3. Prominent models such as V.G.G. and RESNET are modified by reducing the layers count, which is suitable for datasets with poor quality and low scale.
4. In addition, we have trained models from fundamental CNN to recent pre-trained transfer learning models on the Croatian dataset, and the results are analyzed.

2. Methodology

Results of the current works improved with the proposed methodology. The structure of our method is shown in Figure 2. First, 6128 underwater images were collected from the subset of the ImageNet dataset. Second, we have trained these images with the help of CycleGAN architecture to develop a model (i.e., UIEGAN). This UIEGAN architecture enhances the input images (in our case Croatian dataset). Second, the dataset was balanced by training DCGAN for creating many synthetic images (i.e., augmentation). We applied two GAN models to get the Augmented Enhanced Image dataset from the original dataset. Then, we trained three deep neural networks (basic CNN, Smaller RESNET, and SmallerVGG) on the Augmented Enhanced Image dataset to get classification results.

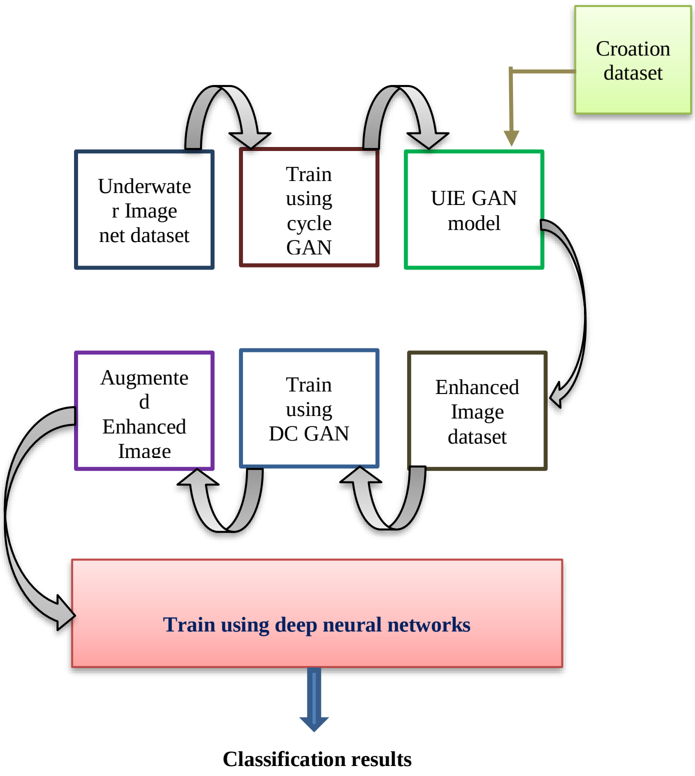


Figure 2. The flow of our proposed methodology

2.1 Image Enhancement using UIEGAN

The algorithms resize input images before sending them to the CNN models. Henceforth, the critical models require a greater size of the input image. The linear interpolation techniques applied in most CNN models might yield more distortion from the input image, mainly when dealing with low-quality images, for example, in the Croatian dataset. Hence, to address this issue, we formulate a strategy to increase the data quality with a trained model (i.e., UIEGAN). There exist many conventional techniques for improving the quality. The SRGAN [16] is the popular technique using generative adversarial networks. In [12], the authors used SRGAN to increase the quality of the Croatian dataset, but this pre-trained SRGAN model is suitable for terrestrial images. It is better to use a model which is trained on underwater images. In this work, a model is designed through underwater images gathered from the ImageNet dataset.

This model was trained using CycleGAN [17], mainly used to generate Zebra images from the Horse images. The primary objective of this UIEGAN model is to take the original input images and increase the quality of the images and call this Enhanced Image Dataset. This Enhanced Image Dataset is more helpful in extracting the better features in each layer in CNN to produce better results.

The sample enhanced images generated from the UIEGAN model are shown in Figure 3 vividly indicates the differences between an enhanced image dataset with the original sample images shown in Figure 1. Further, the improved image dimensions are 256 x 256-pixel width and height. If the original images are resized to 256 x 256-pixel width and height, we get more pixelated images, leading to severe distortion problems. By utilizing this UIEGAN, we can successfully increase original image quality and diminish the issue of distortion.

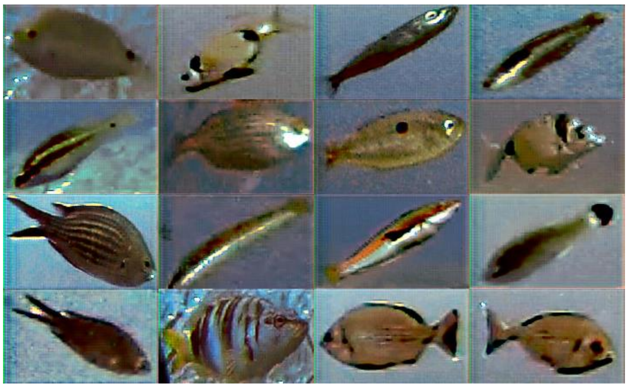


Figure 3. Sample images generated using the UIEGAN model.

2.2 Distribution of the dataset

The second issue addressed is the low-scale data for CNN models. The Croatian dataset contains about 794 images categorized into 12 classes. The data distribution of the original dataset is represented in Figure 4. It is clear the dataset is minimal, and each type is imbalanced. Each species category contains 500 to 1000 images for deep neural networks to get better accuracy. To deal with this class imbalance, we generated synthetic images of around 300 images for each class using DCGAN. The data distribution after training the model is shown in Figure 5. The augmented dataset is more balanced compared with the original dataset. In our work, 300 synthetic images are generated for each class, and the number of images can still be increased to get more images. The area plot of the dataset before and after the generation of synthetic images is shown in Figure 6. The blue and orange colors denote the distributions before and after the augmentation.

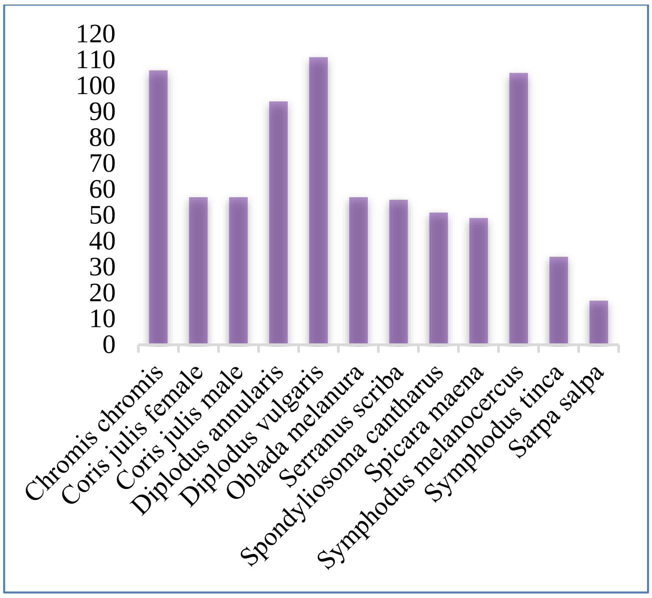


Figure 4. Distribution of the Croatian dataset (original)

2.3 Synthetic Image Generation using DCGAN

Generative models have been analyzed in recent years and characterized into two classes (i.e., parametric and non-parametric). The main aim of these GANs is to generate synthetic (i.e., fake) images which are perceptually nearer to their basic authentic originals. The images generated using GAN [18] experience noise and are inconceivable. An extension to this method [19] uses a Laplacian pyramid approach to render the images with the best quality. Yet, they experienced the objects looking unstable due to noise chaining numerous models. Later, DCGAN was proposed in [20] to bridge the gap between supervised CNNs and unsupervised learning. The DCGAN exhibits a strong candidate for unsupervised learning, and the adversarial pair learns a hierarchy of representations from object parts in both generator and discriminator [21]. For generating synthetic images, we included *two* neural networks while training: (1). A generator that can take a randomly generated noise vector as an input and yields an output image is the same (i.e., fake image). (2).

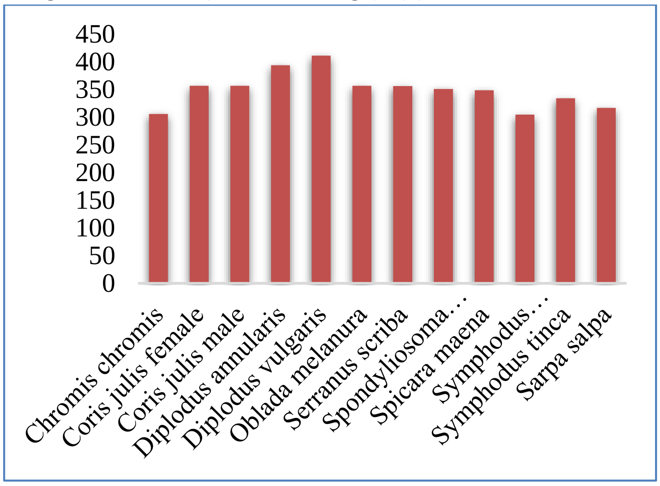


Figure 5- Distribution of the augmented Croatian dataset (after generating synthetic images)

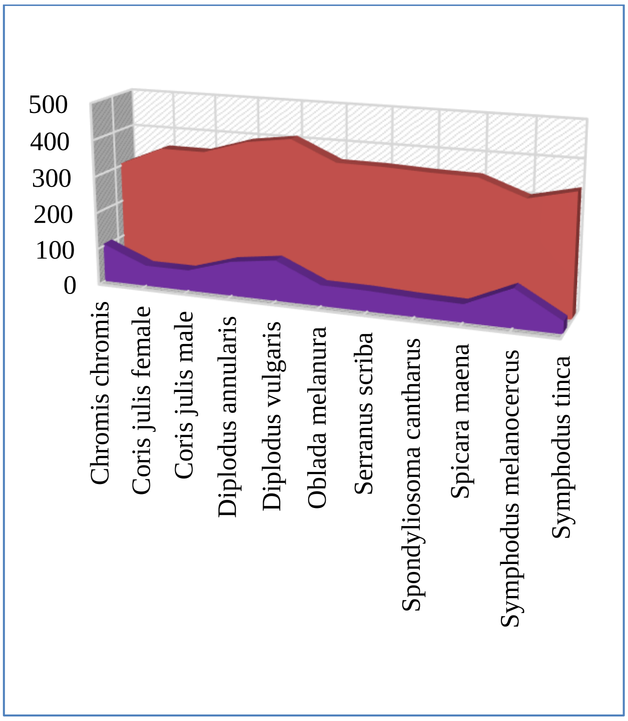


Figure 6. Comparison of the original data distribution with augmented data distribution.

A discriminator will try to mention that the image is "real" or "fake." Through training these networks at a similar interval, one provides feedback to the other. During training this GAN, the generator's objective is to improve in generating synthetic images. The discriminator cannot find the difference between real and artificial data. The steps involved in the training process of GAN are shown in Figure 7.

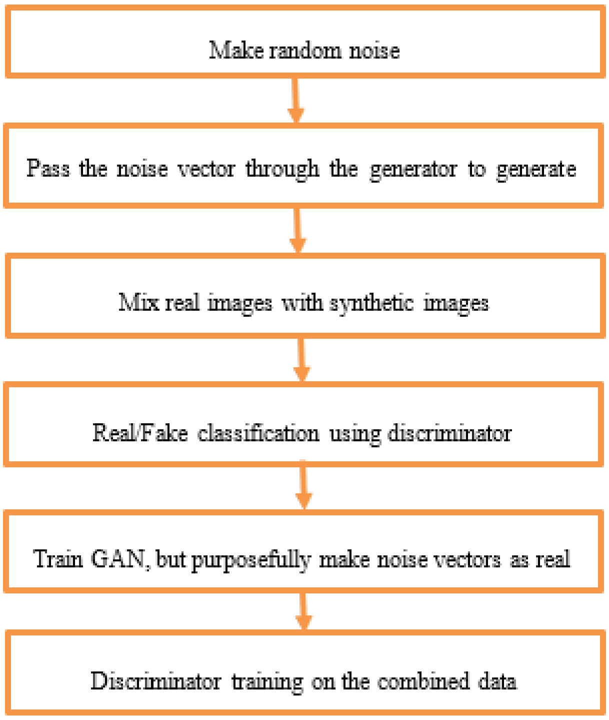


Figure 7- The steps in the training process of GANs

First, random noise is generated and passed through the generator to generate the fake image. Next, we sample the original images from the training data and mix them with the fake images. Now, the discriminator is trained to classify whether each image is real or fake. Once again, generate the noise, but we purposefully make this noise vector as a real image. Then, we train the noise vectors and real image labels to develop more synthetic images. The architecture of GAN creates synthetic images are represented in Figure 8. The sample of synthetic images generated through our GAN model is shown in Figure 9.

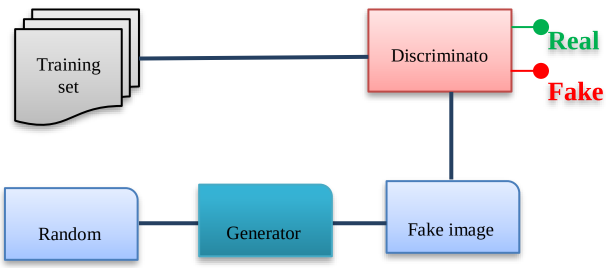


Figure 8- The architecture of GAN for generating synthetic images from the Enhanced Image Dataset.

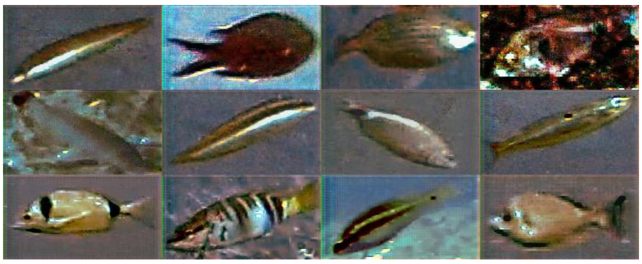


Figure 9- Synthetic images generated through the training process of GAN. All the images shown here do not exist in the original dataset. One sample image is shown for each category of 12 classes.

There are two loss functions in GAN, one is for the discriminator, and the other is for the generator. The discriminator must classify the real and not real images, which is like binary classification. The correct loss function is binary cross-entropy. The discriminator loss can be computed using Eq. (1).

$$J_D = -\frac{1}{N}\sum_{n=1}^{N}\{p_n \log \hat{p_n} + (1 - p_n) * \log(1 - \hat{p_n})\} \quad (1)$$

The generator loss can be computed using Eq. (2). Freeze the discriminator layers so only the generator is trained.

$$J_G = -\frac{1}{N}\sum_{n=1}^{N}\log \hat{p_n}\ ,\ \hat{p_n} = \text{fake image, target is always 1.} \quad (2)$$

J_D *is the discriminator loss and* J_G *is the generator loss* p_n', is scalar value concerning 'i'. p_n, is the target value.

3. Results and Discussion

This section provides our experimental study to classify fish instances from the given low-scale and low-resolution underwater image dataset (i.e., the Croatian dataset). The study includes three steps. First, we applied the basic convolutional networks (Basic-CNN), second customized RESNET model (SmallerRESNET), and third customized V.G.G. model (SmallerVGG) to the actual dataset. Further, we applied these three models to the augmented enhanced dataset to show that our proposed methodology gives better results after pre-processing. The difference between the three models concerns the architecture. The conventional RESNET and V.G.G. models are customized to get better results.

The basic CNN architecture includes three convolutional layers ((Conv2D+RELU => Max Pooling) * 3), along with a dense layer with a softmax layer of 12 classes. Here, "*" represents the same block repeated three times. The activate function is "Relu," and the activation function is "softmax" in the last layer. The pool and strides are of size (2 x 2), and the filters in each layer are 8, 16, and 32. Since we have three convolutional layers, our input size is 32 x 32 x 3. The architecture used for training the Basic-CNN model is shown in Figure 10. The purpose of taking a minimum number of layers is to show the effectiveness of our methodology. The results may vary by increasing the number of layers.

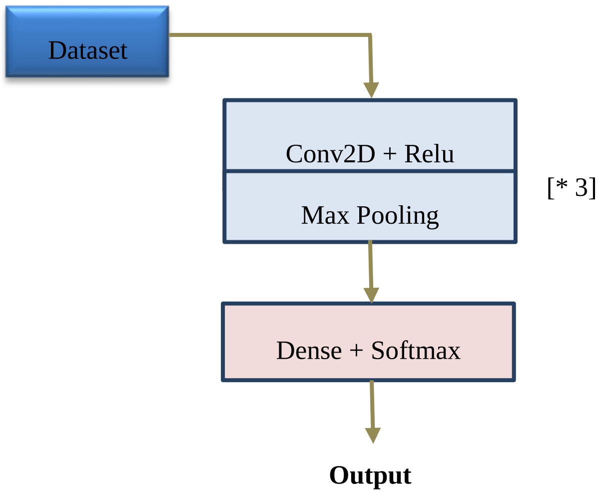


Figure 10- The architecture used for training a basic CNN model

However, we have tried the transfer learning with eight different models (INCEPTIONRESNETV2, RESNET50, VGG16, INCEPTIONV3, XCEPTION, DENSENET, MOBILENET, and NASNET), but the results are poor due to the limited size of the data set.

Table 1: Training results of Transfer learning on the Croatian dataset with the popular pre-trained models

Architecture	Train Accuracy	Train loss	Validation Accuracy	Validation Loss
INCEPTIONRESNETV2 [22]	99.81	0.0086	77.38	5.5050
RESNET50 [23]	34.13	2.4359	29.17	2.9025
VGG16 [24]	95.38	0.1253	71.43	1.0380
INCEPTIONV3 [25]	98.82	0.0636	73.21	10.7376
XCEPTION [26]	97.86	0.2110	69.64	8.8683
DENSENET [27]	99.04	0.0518	79.76	5.4835
MOBILENET [28]	99.96	4.5862	86.90	5.6054
NASNET [29]	99.26	0.1550	68.45	34.5343

It is observed that models like INCEPTION and MOBILENET give ~100 % of training accuracy, but the test accuracy is around ~77 % only. Besides, the test loss is big, and it is clear that the problem of overfitting to the given data. The raw dataset is split into 531 training images and 138 test images. Each transfer learning-based model is trained with 100 epochs—the results of these models over the raw dataset are presented in Table 1.

Basic-CNN model is trained (train and test split as 75% and 25%) with Adam optimizer, rather than using S.G.D., and the evaluation measures (i.e., precision, recall, and f1-score) are mentioned in Figure 11 and Figure 12. Figure 11 shows the measures on the original dataset, and Figure 12 shows the measures after our pre-processing (augmented enhanced dataset) method.

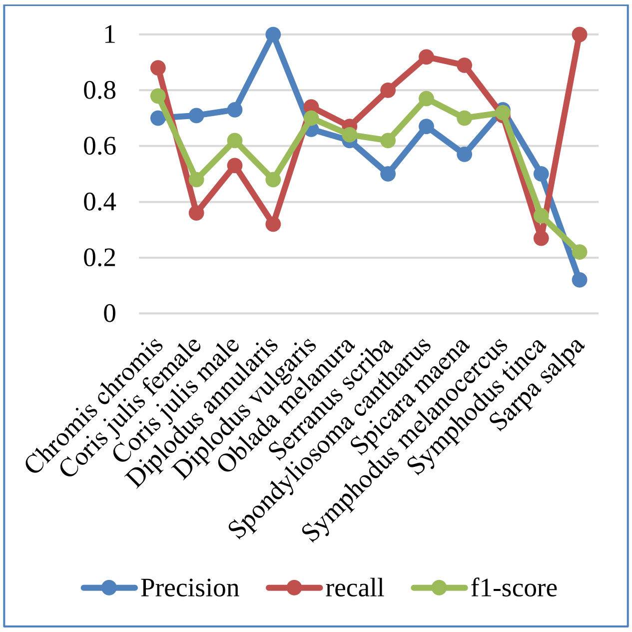


Figure 11- The evaluation results of basic CNN on the original dataset.

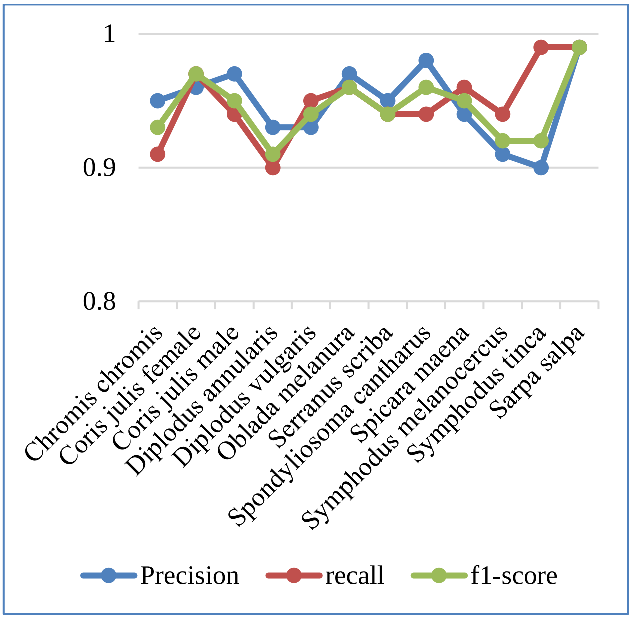


Figure 12- The evaluation results of basic CNN on the augmented enhanced dataset.

The second architecture is custom RESNET (i.e., SmallerRESNET), which accepts the shape of the input image and the number of classes. The architecture includes eight blocks along with an output layer. The first block consists of a convolutional layer followed by batch normalization with max pooling and 0.25. The second block convolution layer is followed by batch normalization. The same layers are repeated till block five consecutively. Block 6 uses a dense layer with 1024 layers followed by batch normalization with a drop out of 0.5. In the 7th block, the dense layer with 512 layers is utilized with batch normalization. In each block, we utilized the activation function as RELU and padding as the same. The filters used in block 1 to block 5 are 32, 64, 64, 128, and 128. We utilized the dense layer with 12 layers with a softmax activation function in the output block.

The architecture of SmallerRESNET is shown in Figure 13 (a). For simplicity, all blocks are shown in the diagram, and "*" indicates that the block's number is repeated. For instance, "* 2" is the block repeated two times. Further, we also applied RESNET50 architecture to our input dataset from scratch. The results of the RESNET50 model on the raw dataset and our enhanced dataset are shown in Figure 14 and Figure 15. The results of our customized RESNET50 model on the raw dataset and our enhanced dataset are shown in Figure 16 and Figure 17. The difference between the conventional RESNET and SmallerRESNET exists in blocks and the layers of every block.

The third architecture is a custom V.G.G. network (i.e., a SmallerVGG), which accepts the shape of the input image and the number of classes. For our case, the input image is resized to 96 X 96 with a depth of three channels. The architecture of SmallerVGG is shown in Figure 13(b). The first convolutional layer has 32 filters with 3 X 3 kernels, and we used 'relu' activation followed by batch normalization.

The pooling layer uses 3 X 3 pool sizes to minimize the spatial dimensions instantly from 96 X 96 to 32 X 32. We also added drop-out in our network model. Drop-out works by randomly disconnecting nodes from the current layer to the next layer. Likewise, added (CONV => RELU) * 2 layers are added before applying another POOL layer. Stacking multiple CONV and RELU layers together allows us to learn better features. Next (CONV => RELU) * 2 => POOL layer is added with dropout of 0.25. Eventually, we had a set of fully connected layers and a softmax activation function (F.C. => RELU).

Further, we have added conventional data augmentation method for generating images using random transformations to avoid overfitting problems. A few changes include rotation, shift range horizontally and vertically, shearing, zooming, flipping, and the fill mode is taken as nearest. The Adam optimizer is utilized for training a network. The model is introduced with a batch size of 32 for 100 epochs. The raw dataset is split into 531 samples (80%) for training and 138 examples (20%) for the test set. The augmented enhanced dataset is partitioned into 2961 samples (80%) to train and 988 (20%) samples to test. The training and validation loss of the model during the initial dataset training is shown in Figure 18(a). The training and validation loss of the model during the augmented enhanced dataset training is shown in Figure 18(b). The graph shows that the training and validation accuracy is increased, and the training/validation loss is decreased after pre-processing data. In the three models, the loss is evaluated through categorical cross-entropy from Eq. (3).

$$\text{Loss} = -\sum_{i=1}^{n} p_i * \log p_i' \qquad (3)$$

Here, p_i', is scalar value concerning 'i'. $'p_i'$ is the target value, and 'n' is the number of classes in the classification.

The comparison results of the three models are shown in Table 2. Basic-CNN architecture is giving training 77.98% and 64.82% validation accuracy. This architecture's training and validation loss are 0.6455 and 1.2589, respectively.

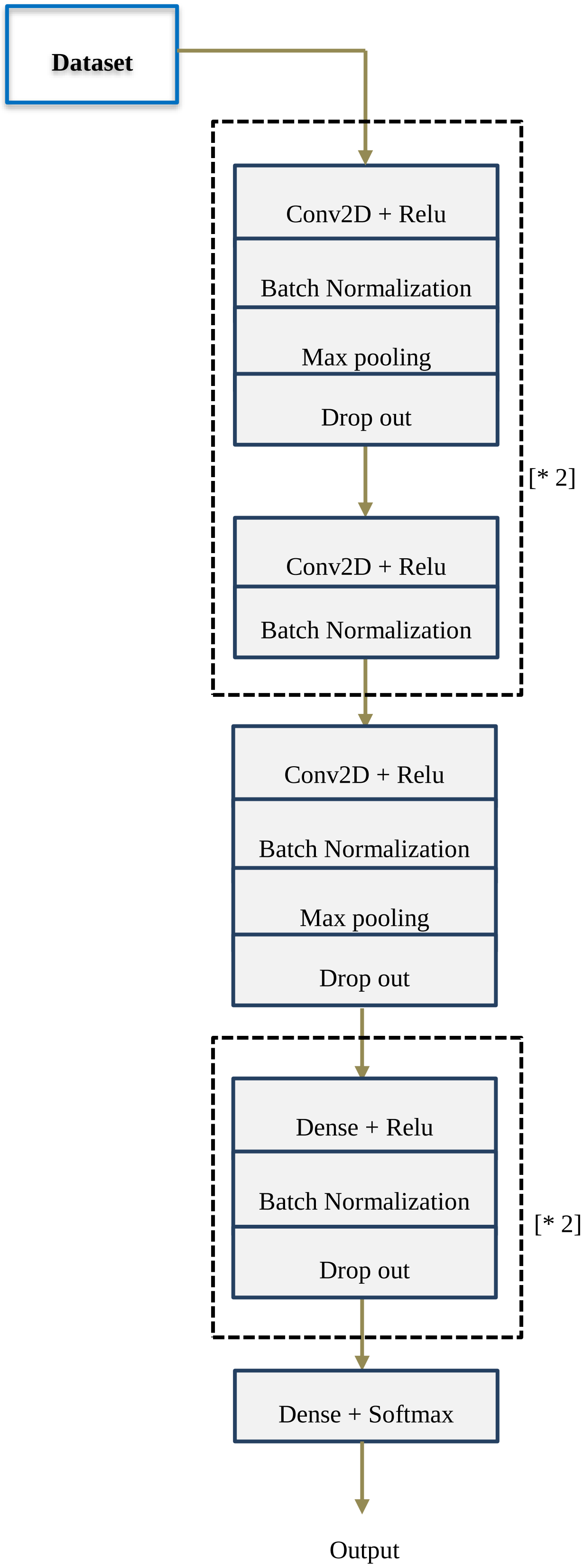


Figure 13- (a) SmallerRESNET

There is a significant improvement observed in the augmented enhanced dataset. We tried to improve the accuracy with SmallerRESNET, but this architecture does not yield better results. We got only 41.24% training accuracy and 33.12% validation accuracy. The loss is also a little high compared with the Basic-CNN. This model's training and validation loss are 1.7447 and 2.2870, respectively, on the original data. But we got satisfactory results on the augmented enhanced dataset by training the same model. We achieve 82.56% of training accuracy and 96.48% of validation accuracy.

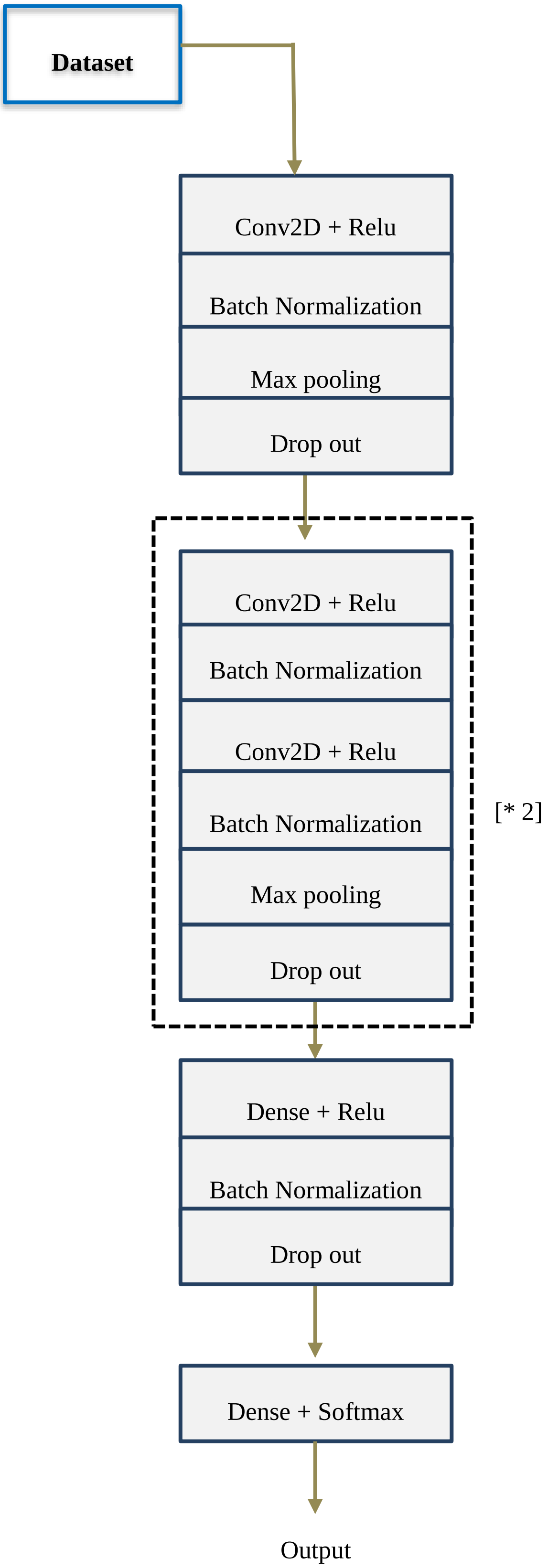


Figure 13- (b). SmallerVGG.

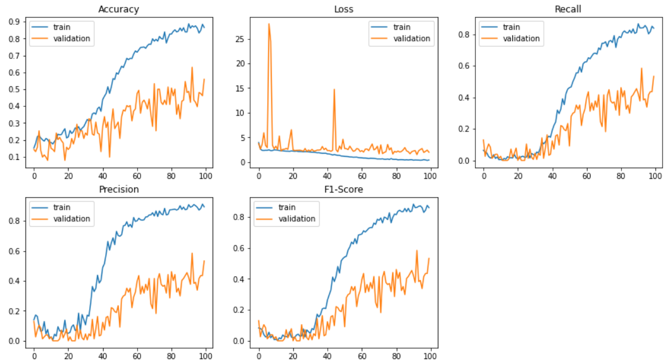


Figure 14- Visualized results of RESNET50 on the original Croatian dataset after training.

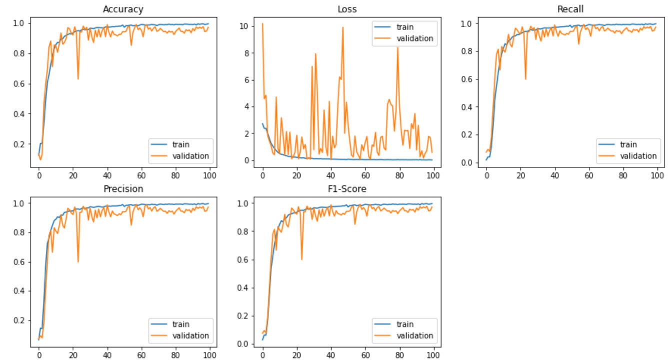


Figure 15- Visualized results of RESNET50 on the Augmented Enhanced Croatian dataset after training

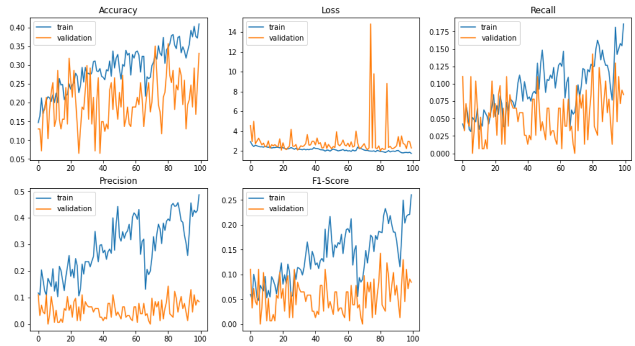


Figure 16- Visualized results of SmallerRESNET on the original Croatian dataset after training.

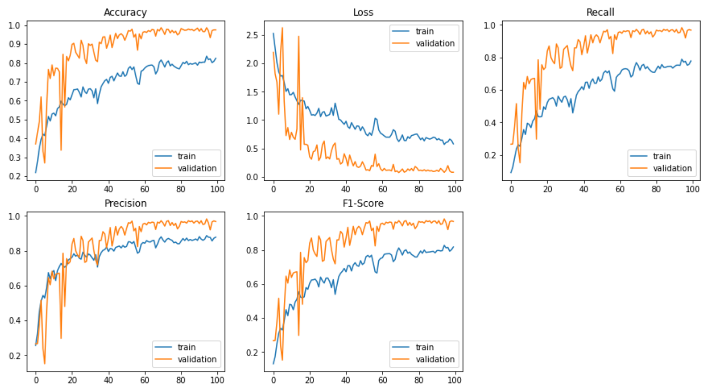
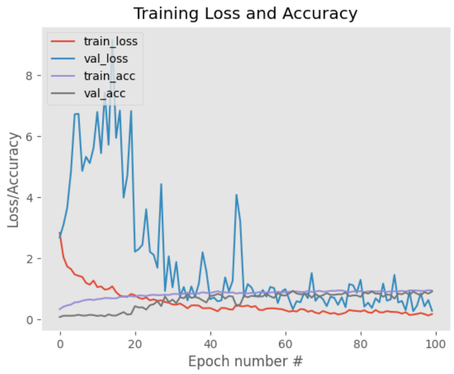
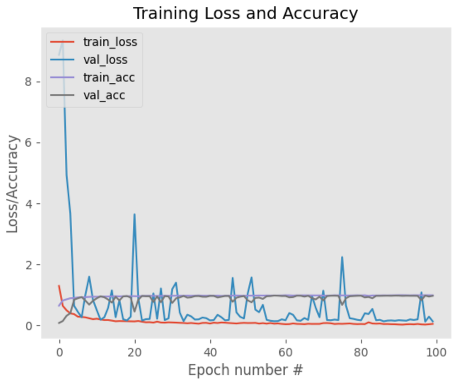


Figure 17- Visualized results of SmallerRESNET on the Augmented Enhanced Croatian dataset after training



(a)



(b)

Figure 18- Experimental results of SmallerVGG model training and validation accuracy and training and validation loss on the original and augmented enhanced datasets. (a). SmallerVGG results on the original dataset. (b). SmallerVGG outcomes over the expanded enhanced dataset.

The training and validation loss of the model are 0.5729 and 0.0811, respectively. The other evaluation metrics, precision, recall, and f1-score, are also recorded and are shown in Figure 14 to Figure 17. The SmallerVGG architecture got 94.73% of training accuracy, 83.19% of validation accuracy and training loss is 0.1455, and validation loss is 0.2814 on the original dataset. The model gives 98.85% of accuracy in training and 97.04% of validation accuracy. The training loss and validation of the model are 0.0442 and 0.1319, respectively. After analyzing these experiments, the SmallerVGG network gives better results and gets the highest accuracy for training and validation data. Hence, we considered this model and compared the results of this architecture with and without using the enhancement technique. Table 3 shows the comparative analysis of training, validation accuracy, and training and validation loss. We used two data augmentation methods; one is a fundamental augmentation, and another is DCGAN. The primary augmentation (Data Generator) uses fundamental transformations such as flipping, rotation, scaling, etc. The SmallerVGG with only Data Generator and no augmentation techniques give 94.73% training accuracy and 83.19% validation accuracy.

Using multi-layer perceptron using multi-scale fusion to enhance the dataset and the primary augmentation gives 95.43% of training accuracy and 85.39% validation accuracy. Then we used the primary augmentation and enhanced the images using UIEGAN. The training and validation loss is increased to 97.68% and 85.53%. The loss is also reduced compared with M.L.P. using a Multi-scale fusion strategy. Further, our method (DCGAN for augmentation and UIEGAN for improvement) gives 98.85% training accuracy and 97.04% validation accuracy. Hence, we consider this model and test it on the ignored images during the training and validation dataset. The model gives an average of 95.64% classification accuracy over test data. The comparison results are shown in Table 4.

Table 2, Summarizes the training results of three architectures (Basic CNN, Smaller RESNET, and SmallerVGG) on the original and augmented enhanced datasets.

Data	Architecture	Train Accuracy	Train loss	Validation Accuracy	Validation Loss
Original dataset	Basic CNN	77.98	0.6455	64.82	1.2589
	SmallerRESNET	41.24	1.7447	33.12	2.2870
	SmallerVGG	94.73	0.1455	83.19	0.2814
Augmented Enhanced dataset	Basic CNN	91.35	0.4225	90.42	0.4421
	SmallerRESNET	82.56	0.5729	96.48	0.0811
	SmallerVGG	98.85	0.0442	97.04	0.1319

Table 3. Comparison of results of SmallerVGG architecture before and after the pre-processing using data augmentation and image enhancement.

Model	Augmentation	Enhancement	Train Accuracy	Train loss	Validation Accuracy	Test loss
SmallerVGG	Data Generator	-	94.73	0.1455	83.19	0.2814
	Data Generator	MLP using Multiscale fusion [30]	95.43	0.1528	85.39	0.9603
	Data Generator	UIEGAN	97.68	0.0842	85.53	0.5896
	DCGAN + Data Generator	UIEGAN	98.85	0.0442	97.04	0.1319

Table 4. Comparison of results of the proposed work with related works

Author(s)	Model/Architecture	Accuracy
Jaeger et.al [15]	CNNs + SVM	66.78%
Alex Krizevsky et al. [31]	AlexNet	62.35%
K. Simonyan et al. [24]	VGG-16	72.07%
Christian Szegedy [32]	Inception-v4	78.25%
K. He et.al. [23]	RESNET-50	80.15%
Baseline [33]	Baseline	66.78%
	B-CNNs [34]	83.52%
QIU et.al [12]	B-CNNs + S.E. blocks [35]	83.78%
	BCNNs + refined S.E. blocks	83.92%
Our method	UIEGAN + DCGAN + SmallerVGG	95.64%

4. Conclusion

Applying recently trained deep neural network architectures gives research and the community many advantages. Transfer learning provides less work in preparing large models for a particular classification task. VGG16 is a prominent model developed but could not classify the fish categories with limited data with low quality. We proposed a model SmallerVGG (minimized version of V.G.G.) that gives better results than traditional CNN models in this work. The model is customized because of the poor quality of data since the CNN models require high-quality images. We disclosed two issues, such as handling the quality and imbalanced distribution of the data. The first issue is addressed by enhancing the images using UIEGAN, trained on a subset ImageNet dataset using CycleGAN. The second issue is addressed by generating the synthetic data using DCGAN to get a more balanced distribution. This will empower the network to analyze the features from the dataset more accurately to achieve better classification results. The exploratory outcomes show that our strategy gives excellent results compared with the other popular CNN models, with a classification accuracy of 95.64%. However, the accuracy can improve further by fine-tuning the CNN architecture. In future works, we will combine this method with other recently developed architectures and work with other fine-grained fish classification datasets. Further, we intend to work on fish identification and classification and fish tracking in real-time underwater videos

References

[1] Zhuang, Peixian, Chongyi Li, and Jiamin Wu. "Bayesian retinex underwater image enhancement." Engineering Applications of Artificial Intelligence 101 (2021): 104171.

[2] Shortis, M., Abdo, E.H.D., 2016. A review of underwater stereo-image measurement for marine biology and ecology applications. In: Oceanography and Marine Biology. C.R.C. Press, pp. 269–304.

[3] Liu, Shasha, et al. "Embedded online fish detection and tracking system via YOLOv3 and parallel correlation filter." *OCEANS 2018 MTS/IEEE Charleston*. IEEE, 2018.

[4] Xu, Wenwei, and Shari Matzner. "Underwater fish detection using deep learning for water power applications." *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2018.

[5] Gupta, D. J. . (2022). A Study on Various Cloud Computing Technologies, Implementation Process, Categories and Application Use in Organisation. International Journal on Future Revolution in Computer Science &Amp; Communication Engineering, 8(1), 09–12. https://doi.org/10.17762/ijfrcsce.v8i1.2064

[6] Yadav, P. ., S. . Kumar, and D. K. J. . Saini. "A Novel Method of Butterfly Optimization Algorithm for Load Balancing in Cloud Computing". International Journal on Recent and Innovation Trends in Computing and Communication, vol. 10, no. 8, Aug. 2022, pp. 110-5, doi:10.17762/ijritcc.v10i8.5683.

[7] Pedersen, Malte, et al. "3D-ZEF: A 3D zebrafish tracking benchmark dataset." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

[8] Knausgård, Kristian Muri, et al. "Temperate fish detection and classification: A deep learning-based approach." *Applied Intelligence* (2021): 1-14.

[9] L. Meng, T. Hirayama, and S. Oyanagi, "Underwater-drone with panoramic camera for automatic fish recognition based on deep learning," IEEE Access, vol. 6, pp. 17880–17886, 2018.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, Vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1097–1105.

[12] C. Szegedy et al., "Going deeper with convolutions," in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit., Jun. 2015, pp. 1–9.

[13] H. Qin, X. Li, J. Liang, Y. Peng, and C. Zhang, "DeepFish: Accurate underwater live fish recognition with a deep architecture," Neurocomputing, vol. 187, pp. 49–58, Apr. 2016.

[14] Qiu, Chenchen, et al. "Improving transfer learning and squeeze-and-excitation networks for small-scale fine-grained fish image classification." *IEEE Access* 6 (2018): 78503-78512.

[15] Zhao, Zhenxi, et al. "Composited FishNet: Fish Detection and Species Recognition from Low-quality Underwater Videos." *IEEE Transactions on Image Processing* (2021).

[16] X.-S. Wei, C.-W. Xie, J. Wu, and C. Shen, "Mask-CNN: Localizing parts and selecting descriptors for fine-grained bird species categorization," Pattern Recognit., vol. 76, pp. 704–714, Apr. 2018.

[17] J. Jaeger, M. Simon, J. Denzler, V. Wolff, K. Fricke-Neuderth, and C. Kruschel, "Croatian fish dataset: Fine-grained classification of fish species in their natural habitat," in Proc. Mach. Vis. Animals Behav., 2015, pp. 1–7.

[18] C. Ledig et al. (Sep. 2016). "Photo-realistic single image super-resolution using a generative adversarial network." [Online]. Available: https://arxiv. org/abs/1609.04802.

[19] Cho, Se Woon, et al. "Semantic segmentation with low light images by modified CycleGAN-based image enhancement." *IEEE Access* 8 (2020): 93561-93585.

[20] Goodfellow, Ian J., Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial networks." *arXiv preprint arXiv:1406.2661* (2014).

[21] Denton, Emily, Chintala, Soumith, Szlam, Arthur, and Fergus, Rob. Deep generative image models using a laplacian pyramid of adversarial networks. arXiv preprint arXiv:1506.05751, 2015.

[22] Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." *arXiv preprint arXiv:1511.06434* (2015).

[23] Yu, Yang, et al. "Unsupervised representation learning with deep convolutional neural network for remote sensing images." *International Conference on Image and Graphics*. Springer, Cham, 2017.

[24] Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. No. 1. 2017.

[25] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[26] M. J. Traum, J. Fiorentine. (2021). Rapid Evaluation On-Line Assessment of Student Learning Gains for Just-In-Time Course Modification. Journal of Online Engineering Education, 12(1), 06–13. Retrieved from http://onlineengineeringeducation.com/index.php/joee/article/view/45

[27] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

[28] Szegedy, Christian, et al. "Rethinking the inception architecture for computer vision." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[29] Chollet, François. "Xception: Deep learning with depthwise separable convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

[30] Kumar, S., Gornale, S. S., Siddalingappa, R., & Mane, A. (2022). Gender Classification Based on Online Signature Features using Machine Learning Techniques. International Journal of Intelligent Systems and Applications in Engineering, 10(2), 260–268. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2020

[31] Huang, Gao, et al. "Densely connected convolutional networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[32] Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." *arXiv preprint arXiv:1704.04861* (2017).

[33] Saraireh, J., & Joudeh, H. (2022). An Efficient Authentication Scheme for Internet of Things. International Journal of Communication Networks and Information Security (IJCNIS), 13(3). https://doi.org/10.17762/ijcnis.v13i3.3422

[34] Zoph, Barret, et al. "Learning transferable architectures for scalable image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

[35] M. Sudhakara and M. Janaki Meena, "Multi-scale fusion for underwater image enhancement using multi-layer perceptron," IAES International Journal of Artificial Intelligence (IJ-AI), Vol. 10, No. 2, June 2021, pp. 389~397. DOI: 10.11591/ijai.v10.i2.pp389-397.

[36] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems* 25 (2012): 1097-1105.

[37] Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. No. 1. 2017.

[38] Singh, S. ., Wable, S. ., & Kharose, P. . (2022). A Review Of E-Voting System Based on Blockchain Technology. International Journal of New Practices in Management and Engineering, 10(04), 09–13. https://doi.org/10.17762/ijnpme.v10i04.125

[39] K. Anantharajah et al., "Local inter-session variability modelling for object classification," in Proc. IEEE Winter Conf. Appl. Comput. Vis., Mar. 2014, pp. 309–316

[40] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear CNN models for fine-grained visual recognition," in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2015, pp. 1449–1457.

[41] J. Hu, E. Wu, L. Shen, and G. Sun. (Sep. 2017). "Squeeze-and-excitation networks." [Online]. Available: https://arxiv.org/abs/1709.01507.