Journal of Computer Science

Original Research Paper

Underwater Fused Image Classification Using Deep Learning Based Resnet and Hybrid PSO + HHO Model

[1]Devika Sarath and [2]M. Sucharitha

[1]Department of Electronics and Communication Engineering,
Noorul Islam Centre for Higher Education Kumaracoil, Tamil Nadu, India
[2]School of Electronics Engineering, VIT-AP University, Amaravathi, Andra Pradesh, India

Article history
Received: 01-11-2022
Revised: 23-01-2023
Accepted: 02-02-2023

Corresponding Author:
Devika Sarath
Department of Electronics and Communication Engineering, Noorul Islam Centre for Higher Education Kumaracoil, Tamil Nadu, India
Email: devikagokul@gmail.com

Abstract: In image fusion, multiple images are combined into one with minimal distortion and data loss. Image fusion forms a highly configured image with good results. The image fusion technique is employed in various domains, including remote sensing, robotics, medical applications, and underwater image processing. The focus of this study is on a unique underwater image fusion approach that allows for greater flexibility in the construction of fusion criteria. The characteristics of the input images are extracted using a Modified Tetrolet Transform (MMT), which may be employed together or independently. Our aim is to apply image deep learning algorithms such as GoogleNet, AlexNet, and ResNet for fusion to the original image and optimized algorithm PSO and HHO to be used for optimizing the fused image. Finally, the images are classified as good quality images and poor-quality images. The ResNet with hybrid HHO and PSO method has high efficiency in image fusion, according to numerical data the proposed model with good quality attains accuracy, sensitivity, specificity, precision, and F1-measure are 96.32, 94.25, 95.73, 98.34, and 95.55. in addition, the poor-quality images attain accuracy, sensitivity, specificity, precision, and F1-measure are 96.22, 94.15, 95.63, 98.24, and 95.45 This method optimizes the exposure of the dark areas, increases contrast, and preserves and enhances the edges. Our image trial findings show that this technique substantially improves the underwater image quality.

Keywords: Real-Time Underwater Images, Dark Channel Prior Model (DCP), Tetrolet Transform, GoogleNet, AlexNet, and ResNet

Introduction

Image fusion is an important task to retrieve an image that provides information about the same image at the same time it also helps to reduce the storage capacity of a single image. It is also used to obtain a more complete and accurate description of the same object, which provides easy access for image-guided. The underwater optical imaging quality is of great significance to the exploration and utilization of the Deep Ocean (Gao *et al*., 2021). However, raw underwater images seldom fulfill the requirements concerning low level and high-level vision tasks because of the serious underwater degradation model. The underwater images are rapidly degraded by two major factors. Due to the depth, light conditions, water type, and different light wavelengths, the color tones of underwater images are often distorted. To achieve a more informative image while working with partly blurred images, image fusion is required. Image fusion algorithms scan partly blurred images for information and combine them into a single image. There is less information in the partly blurred images than in the composite image (Yang *et al*., 2021). The fusion of multiple images is used in surveillance systems, image restoration from hazy (Dai *et al*., 2018) and foggy (Tao *et al*., 2020) images, image identification, pan sharpening, multi-focus image fusion, and many other applications. Many academics have recently developed image fusion algorithms to identify medical diagnoses in patients (Li and Zhuang, 2021). Researchers have devised numerous approaches to enhance underwater images (Luo *et al*., 2021). Underwater image processing has recently received a lot of attention as a way to automate and enhance image processing. In the literature, many image fusion approaches for integrating underwater and other types of images have been suggested (Kanagavel *et al*., 2021). The fusion of underwater images is a significant

and common challenge. Many fusion approaches for combining the characteristics of two images into a single image have recently been suggested (Zhu *et al*., 2021). These cutting-edge techniques have a variety of applications, including image pre-processing, target detection, and image categorization. Combining crucial attributes from source images into the fused image is the most challenging step in image fusion (Li and Wang, 2010). To extract picture data, image fusion researchers use Discrete Wavelet Transforms (DWT), contourlet transforms (Lin *et al*., 2020), shift-invariant shear let transforms (Chang, 2020), and Quaternion wavelet TRANSFORMS (Tao *et al*., 2020). Each technique has its own set of constraints and restrictions (Linfeng *et al*., 2020). Image classification and identification applications, including underwater image recognition, image processing, machine learning, deep learning, and Convolutional Neural Networks (CNN) have proven to be good. With the introduction of deep learning, new strategies for reconstructing the merged image are being employed, such as deep properties of the original images, often known as saliency features.

Lewis *et al*. (2007) devised a difficult wavelet-based pixel and region-based fusion method. The image is broken down into various attributes, as well as an area map, using a Dual-tree Complex Wavelet Transform (DCWT). The characteristics of each area are estimated in the DCWT domain and a region-based technique for fusing the images using a region-by-region approach was provided (Lewis *et al*., 2004). Deep Learning (DL) can boost the fusion rate, according to Liu *et al*. (2017). The images were efficiently combined by using deep learning-based fusion. By addressing sparse representation classification in the Tetrolet area, (Shahdoosti and Mehrabi, 2018) created a multimodal image fusion. The input image goes through the tetrolet transform first. To determine significant features from tetrolet sub-band masks, many activity metrics were used. Sparse Unmixing is achieved by combining variable Splitting with the Augmented Lagrangian (SUnSAL) classifier. The classifier chose the coefficients that will appear in the fused image. Cycle-spinning (Shahdoosti and Ghassemian, 2016) was used to prevent artifacts.

Zhang *et al*. (2018) developed a model for multimodal image fusion that allows for Joint Sparsity to be adjusted (JSM). With JSM, distributed compressed sensing is taken into account and a single dictionary learned using K-Singular Value Decomposition (SVD) demonstrates high coherence (Kaur and Singh, 2020). For multi-focus image fusion, Tang and coworkers used neighborhood data to predict focused and defocused pixels in input images using pixel-wise Convolutional Neural Networks. To eliminate fusion artifacts, pCNN can accurately estimate each pixel's focus level in input images (Tang *et al*., 2018).

A spatial domain image fusion method was developed by Zhang *et al*. (2016) the input images are first used to construct the Point Spread Functions (PSF). Following that, input images are blurred using the computed PSFs and a convolution operator. The sharpest pixels in the input images are found using blurred images. They are fused to form a fused image (Zhang *et al*., 2016). In contrast, color reversal artifacts are common (Ell and Sangwine, 2006; Amarsaikhan *et al*., 2012). To get more efficient fusion results, (Sreeja and Hariharan, 2019) employed Texture and edge enhanced images (TEF) (Wang *et al*., 2019). Low contrast images were successfully integrated using the recommended method (Sreeja and Hariharan, 2019). A multi-feature fusion strategy was employed by Kaur and Singh (2020). In the HSV color space, color characteristics are retrieved using moments and histograms. Zernike moments were utilized to extract form features and a grey-level co-occurrence matrix was employed to extract texture. Finally, the computed fused image is fused with the obtained features (Ma *et al*., 2019). A comparative study of different models is shown in Table 1. Researchers have been working on various components of a fusion system, but a good-enough solution for high-accuracy classification and recognition is yet to be developed. To overcome this, here we proposed different deep learning techniques such as GoogleNet, ResNet, and AlexNet, which accurately fuse the two input images into a single image.

Table 1: Comparative study of different models

Ref.	Author	Model	Accuracy (%)
Lewis *et al*. (2007)	Lewis *et al*.	wavelet-based pixel and region-based fusion	78.0
Lewis (2004)	Lewis	DCWT	80.0
Liu *et al*. (2017)	Liu *et al*.	DL	91.5
Shahdoosti and Ghassemian (2016)	Shahdoosti and Ghassemian	SUnSAL	85.0
Zhang *et al*. (2018)	Zhang *et al*.	JSM	79.4
Kaur and Singh (2020)	Kaur and Singh	SVD	73.0
Tang *et al*. (2018)	Tang *et al*.	pCNN	90.5
Zhang *et al*. (2016)	Zhang *et al*.	PSF	92.0

Materials and Methods

Two types of input images from various sensors are selected for further processing in this suggested study. Dehashing color correction procedures are used for both sorts of input images. The dehashing algorithm comprised 3 models, white balance, dark channel prior model, and bilateral filtering. By using this model the images are enhanced. Image decomposition algorithm, that is, modified tetrolet transform is applied to decompose the underwater images. Following, the decomposed images are reconstructed by using a filter bank algorithm. Finally using the deep learning-based technique, the images are fused. The suggested approach is depicted in Fig. 1 as a block diagram.

Dataset

Almost all of the existing underwater image and video datasets generally have both or some of the associated limitations: Repeated deepwater image scenes or lack of data size. Since the images captured from robotic grasping underwater have been very unique in relation to those collected in general marine scenes, the examination of high-level version tasks is tough to give strong support, for example, automated identification and recognition. Setting up a large-scale, differentiated, particular task-driven database in this way has an incredible impulse for model training and evaluation. Furthermore, an appropriate database is also an important reason for the feedback training on the robotic grasping scheme underwater and relevant theoretical research.

The real-world underwater images from 10 movies shot at the primary underwater robot-picking contest, Zhangzidao, Dalian, China, are included in this newly suggested dataset. As for the video sampling environment, the water depth is approximately 3-15 m and all targets in the recognition territories were haphazardly set. Since these videos were recorded at different periods of the day, there were huge contrasts in water quality between shooting fields, the illumination, field depth, fluffy degree, and shading cast in the images are very special. The three sub-datasets developed, as well as their specific settings and capabilities, are listed below.

Underwater Image Quality Set (UIQS)

This sub-dataset will be used to evaluate various image quality improvement strategies. Underwater Color Image Quality Assessment (UCIQE) is a linear combination of chromaticity, saturation, and contrast of underwater images. According to the Underwater Color Image Quality Assessment (UCIQE) metric, images were analyzed and positioned according to their relevant UCIQE scores. To make performance testing of various computations in various underwater circumstances easier, we grouped them into five subgroups named [A, B, C, D, E] and graded them from good to low quality.

Underwater Colour Cast Set (UCCS)

This sub-dataset is used to assess the ability to fix color casts. A dataset of 300 images from UIQS was compiled to examine color shift correction capacity using a typical estimate of the b channel (red-green bias) from the CIE lab. It consists of three 100-image sub-datasets: Bluish, yellowish, and blue-green. Example images from the three sets of RUIE are shown in Fig. 2.

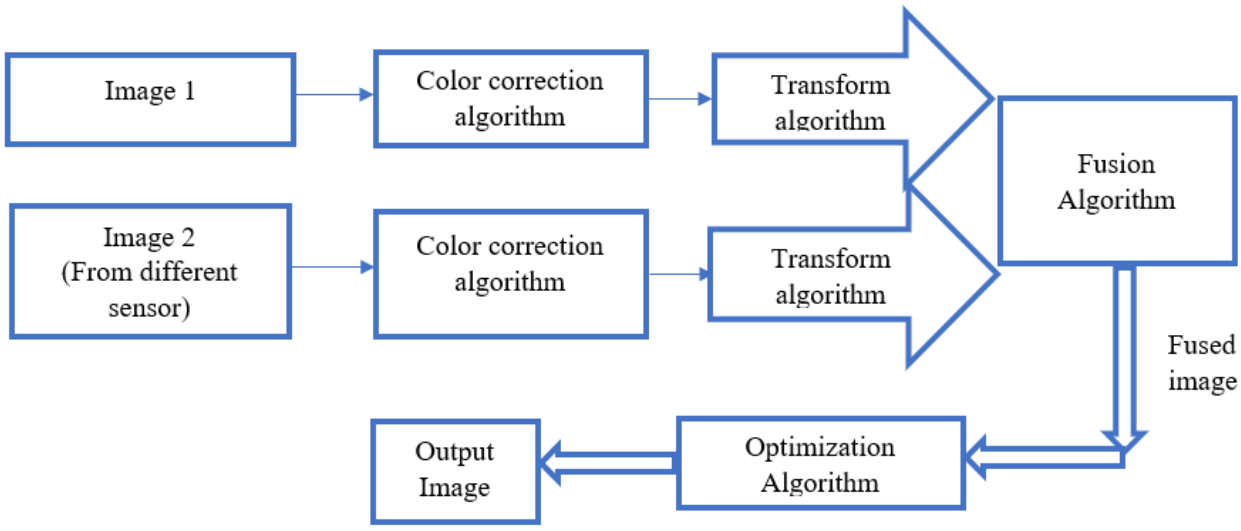


Fig. 1: Block diagram of the proposed methodology

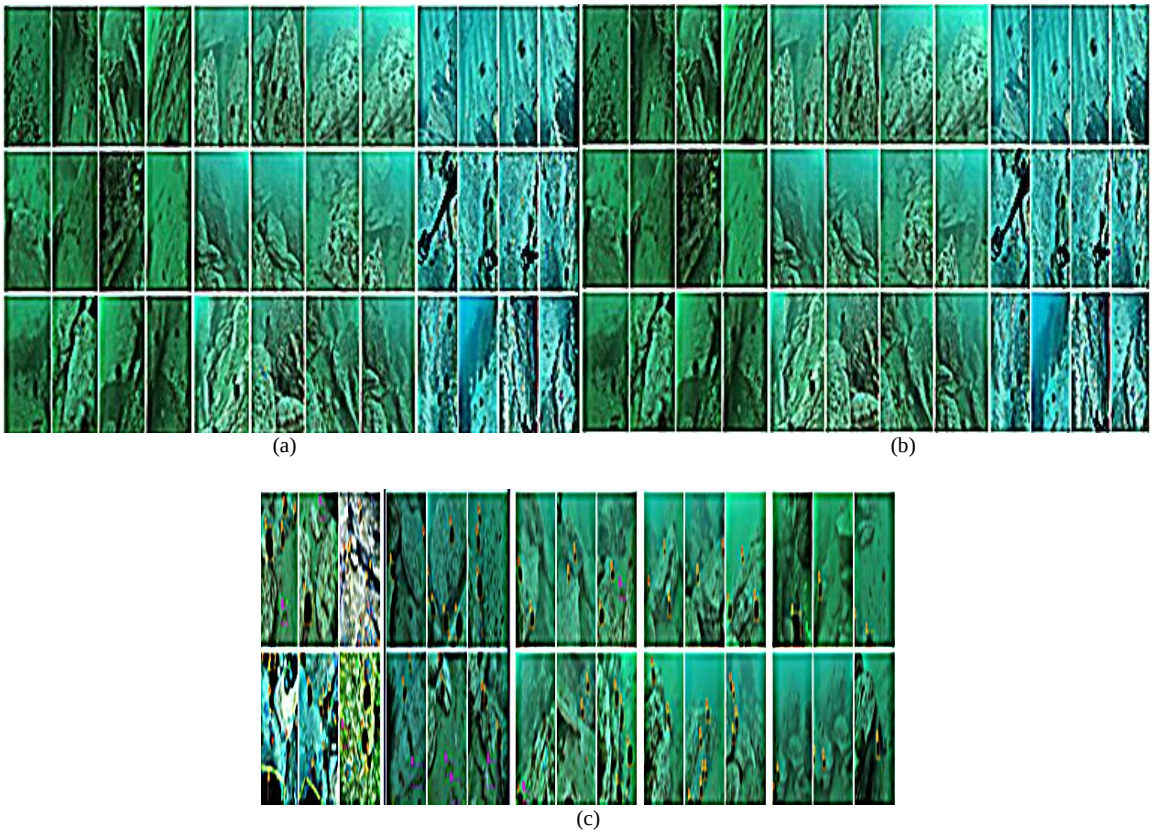


Fig. 2: Example images from the three sets of RUIE. (a) UIQS. (b) UCCS. (c) UHTS

Underwater Task-Driven Test Set (UTTS)

To adapt to high-level visual functions such as recognition and detection, UTTS delivers 300 images, each of which comprises some recognized marine species. The focus was on abalones, sea cucumbers, and sea urchins for the time being, as these are the most popular targets in robotic grabbing operations. In order to investigate the effects of image quality on accuracy findings, UTTS is subdivided into five sub-datasets based on UCIQE scores.

Color Correction Algorithm

The purpose of this section is to examine and evaluate a color-correcting algorithm for underwater image enhancement. Because of the varying lighting conditions, underwater images must first be white-balanced. White balance is an important technique for improving the image's look by eliminating undesirable shade casts when the water depth is more than 30 feet. The Dark Channel Prior (DCP) technique is frequently used to get a haze free image. On the other hand, this method is utilized to improve underwater images (Bhagyasri *et al*., 2019).

When it comes to white balance, when different wavelengths of white light reach the ocean, they are absorbed in proportion to their wavelength, causing an imbalance in the underwater image's red, green, and blue color channels. When the depth of these channels is increased, removing color distortion becomes much more difficult. A white balancing algorithm is used in the suggested technique to remove the color distortion. There are two phases in all automated white-balancing systems. Assessing ambient light or scene lighting is the first and most demanding phase, followed by restoring the image's color balance. Once the ambient light has been measured, it is a very important technique for fixing the colors in the image. White balancing underwater images eliminate color aberrations and remove the artificial color cast. The linear transformation model employed in the correction step is utilized to determine the White Balance (WB) correction parameters. Equation 1 depicts the white balance correcting procedure:

$$F\_WB(x,y) = k \times F(x,y) + \tau \quad (1)$$

Gray level values for *R*, *G*, and *B* channels for every pixel position in the image, represented by $[\![F]\!]\_WB(x,y)$ and $F\_WB(x,y)$ produce white-balanced output. The gain factor and offset value utilized in the white balance correction procedure are *k* Kappa and τ respectively.

Haze is generated by moist visibility-reducing aerosols, which results in reduced scene brightness and increased airtight. The fuzzy image model is commonly referred to as:

$$I(x) = j(x)^{*} t(x) + A\left[1 - t(x)\right] \quad (2)$$

where, I stands for measurement intensity, J for scene radiance, t for medium transmission map, which indicates how much light isn't dispersed and reaches the camera, and A for ambient light. Since the purpose of haze removal is to utilize t and A to get J from I, it's tough to remove haze from a single image.

Outdoor picture data was used to create the dark channel previously. At least one color channel in the majority of non-sky patches comprises pixels with very low and near-zero brightness. The input image's dark channel $〚 J〛$ _dark (x) is supplied as:

$$J\_dark(x) = \min\nolimits_{\top}(y \in \Omega(x))[\min\nolimits_{\top}(c \in [R,G,B])c \quad (3)$$

J_c is *J's* color channel and $\Omega(x)$ is a local patch centered at x. The result of two minimal operators is a black channel: $\min_{\top}(y \in \Omega(x))$ (.) on each pixel and $\min_{\top}(c \in [R, G, B])$ (.) on each channel. Given the preceding notion, the intensity of the black channel approaches zero if J is an outside haze-free image without sky regions:

$$J\_dark(x) \to 0 \quad (4)$$

A bilateral filter decreases noise by being a non-linear, edge-preserving image-smoothing filter. A weighted average intensity value from surrounding pixels is used to change the amount of intensity at each pixel in an image. Gaussian weight distribution can be used. Weights are based on both Euclidean pixel distance and radiometric changes, which is important (e.g., differences in range, such as intensity of color, depth distance, etc.). Crisp edges are maintained by actively looking through each pixel and changing weights appropriately to the neighboring pixels.

Image Decomposition Using Modified Tetrolet Transforms

During image decomposition, the new tetrolet transform adds a block overlap technique to the previous tetrolet transform's coefficients. Implementing the most effective filter bank method is simple. In contrast to the traditional two-dimensional wavelet transformation, stationary tetrolet transformation is a novel multi-scale geometrical tool based on tetrominoes that allow anisotropic image geometric structures to capture information effectively through multi-direction selection. The suggested technique enhances image decomposition redundancy and introduces a novel image-processing algorithm (Shahdoosti and Mehrabi, 2018).

Image Fusion

Across a wide range of image processing domains, deep learning has become a popular research approach in recent years. Deep learning, as we all know, achieves cutting-edge results in a variety of image processing applications, including image categorization. Deep learning may also be beneficial for extracting visual characteristics that include diverse information at various layers. In the last two years, many deep-learning applications have gotten a lot of press (Liu *et al*., 2018). As a result, we believe deep learning can be used to solve the image fusion problem. We propose an approach for synthesizing a single image that incorporates all of the attributes from underwater images in this research using a deep learning framework such as GoogleNet (Tang *et al*., 2017), ResNet (Li *et al*., 2019), and AlexNet (Yuan and Zhang, 2016). For image fusion, deep learning models are utilized. They categorize fused images to improve image quality.

Deep Learning Technique Using GoogleNet

In a study titled "going deeper with convolutions," Google researchers recommended GoogleNet (or inception V1) in 2014 (with the help of several institutions). This design won the ILSVRC 2014 image categorization competition. As well as AlexNet ILSVRC 2012 and ZF-Net ILSVRC 2013, VGG ILSVRC 2014 runner-up have significantly decreased their error rates. This architecture employs 1 × 1 convolutions in the design's center as well as global average pooling.

The 22 layers (27 levels including pooling layers) of the GoogleNet architecture contain 9 inception modules (Figs. 3-4). To offer a 1 × 1 input height and width, the average pooling layer averages out the feature maps formed by the prior inception module. A dropout layer (40%) is utilized immediately before the linear layer. A regularization strategy for preventing network overfitting during training is the dropout layer. Each of the 1000 classes in the ImageNet dataset is represented by a hidden unit in the linear layer. Using the SoftMax function, the probability distribution of an input vector is calculated in the final layer. The chance of a class or event occurring is described by the SoftMax activation function's values. The values of the vector equal one.

A number of aspects of GoogleNet differ from AlexNet and ZF-Net, which were previously considered state-of-the-art systems. As a result, a complex architecture is created using the 1 × 1 convolutional

algorithm and the global average pooling algorithm. Some of these approaches will be applied in architecture:

- 1 × 1 convolution**:** 1 × 1 convolution is used in the design of the conceptual architecture. The number of architectural parameters was reduced due to these convolutions (weights and biases). By reducing the parameters, we may deepen the architecture
- Global average pooling: In older topologies, such as AlexNet, the network's conclusion was the linked layers. In many designs, the majority of parameters are located in these entirely linked layers, resulting in a higher computation cost. Near the end of GoogleNet's architecture, global average pooling is used. The value of this layer is averaged down from 7 × 7 to 1 × 1. There was also a reduction in the number of trainable parameters and an increase in top-1 accuracy of 0.6%
- Inception module: Inception is incompatible with previous designs, such as AlexNet and ZF-Net. The convolution size of each layer in this architecture is fixed. The Inception module pools 1 × 1, 3 × 3, 5 × 5, and 3 × 3 maximums in parallel at the input, stacking the outputs to produce the final result. According to the idea, different-sized convolution filters will handle objects of different sizes better
- Auxiliary classifier for training: There were several intermediate classifier branches in Inception's design that were only used during training. Each branch contains five layers: A pooling layer with 55, a convolution layer with 1024 outputs, a classification layer with 1000 outputs, and a fully connected layer with 1024 outputs. The resulting loss of these layers was added to the total loss using the weight of 0.3. The gradient is kept steady by these layers, which prevents it from fading

The auxiliary classifier shown in Fig. 5 is composed of an average pool layer, a convolutional layer, two fully connected layers, a 70% dropout layer, and a linear layer with a SoftMax activation function. The activations from the previous inception modules are passed on to each of the auxiliary classifiers. An auxiliary classifier is made up of two completely linked layers, an average pool layer, and a conv layer.

Characteristics and Features of Google Net Configuration Table

- An image with a resolution of 224 × 224 pixels is accepted by the GoogleNet architecture's input layer
- Type**:** This is the current tier in the architecture for the component
- Patch size: This is the sweeping window's width, which is applied to both the conv and pooling layers. Swinging windows are the same in height and breadth on both sides
- Stride: The amount by which the filter/sliding window modifies the supplied image is set
- Output size: The current architectural component's output dimensions (height, width, and the number of feature maps) are determined after the layer processes the input
- Depth: An architectural component's own levels/layers are referred to by the number of levels/layers inside the component
- # 1 × 1 # 3 × 3 # 5 × 5: This is the name given to the inception module's many convolutional filters
- # 3 × 3 reduce # 5 × 5 reduce: This relates to how many 1 × 1 filter were used before the convolutions
- Pool proj: After pooling, this is an inception module's total number of 1 × 1 filter
- Params: The weights in the current architectural component are represented by this number
- Ops: This figure represents the component's total number of mathematical operations

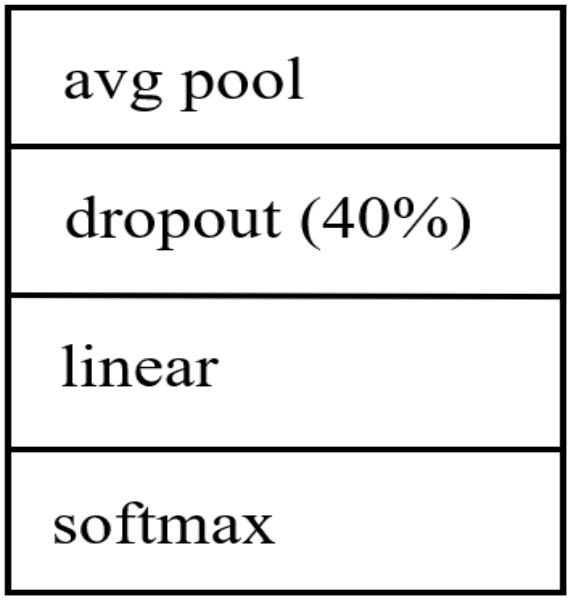


Fig. 3: Partition of googleNet

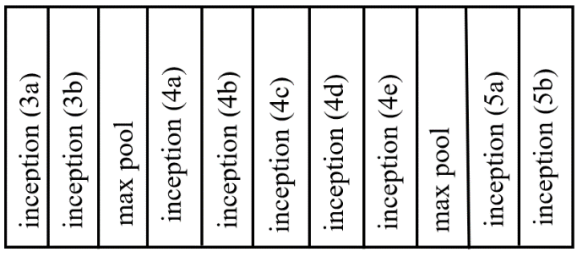


Fig. 4: Inception module of GoogleNet

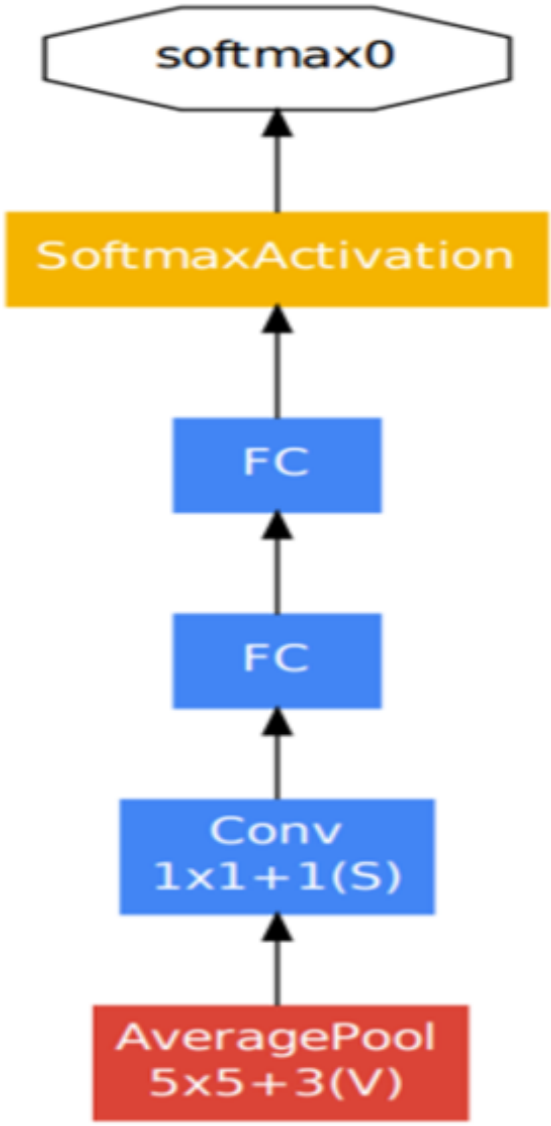


Fig. 5: Auxiliary classifier

Deep Learning Technique Using Alex Net

ImageNet's large-scale visual recognition competition in 2012 was won by AlexNet shown in Fig. 6. Alex Krizhevsky and his colleagues were the first to present the concept in 2012. Each layer of AlexNet has its own set of learning parameters. Except for the output layer, which is made up of a mixture of max pooling and three fully connected layers, the model is made up of five layers, each of which uses ReLu activation. They discovered that employing the ReLu as an activation function cut training time in half. To avoid overfitting in their model, dropout layers were also included. The model is also trained using the ImageNet dataset. There are around 14 million images in the ImageNet database, which are organized into a thousand categories. AlexNet is a system with a lot of depth. The images with dimensions of $227 \times 227 \times 3$ are used as input to this model:

1. Five convolutional layers, three max-pooling layers, two normalization levels, two fully connected layers, and one SoftMax layer make up AlexNet's architecture
2. Convolutional filters and nonlinear activation functions are used for each convolutional layer
3. To achieve maximal pooling, the pooling layers are necessary
4. The input size is fixed because the layers are all connected
5. Normally, the input size is $224 \times 224 \times 3$, but due to padding, it is $227 \times 227 \times 3$
6. There are 60 million parameters in AlexNet

This design consists of three fully linked layers and five convolutional layers. But it's not the only difference between AlexNet and other convolutional neural networks: Here are a handful of AlexNet's distinguishing features:

- ReLU nonlinearity. Instead of using the tan h function, which was popular at the time, Alex Net employs Rectified Linear Units (ReLU). On the CIFAR-10 dataset, a ReLU-trained CNN outperformed a tan h-trained CNN by six times
- Multiple GPUs. Previously, Graphics Processing Units (GPUs) were restricted to three gigabytes of RAM (nowadays those kinds of memory would be rookie numbers). The 1.2 million images in the training batch added to the problem. Two GPUs are used to split half of AlexNet's neurons for multi-GPU training. This not only enables the teaching of larger models but also reduces the amount of time required
- Overlapping pooling. Neighboring clusters of neurons' outputs are "pooled" in CNNs. Adding overlap, on the other hand, reduced error by 0.5%, implying that overlapping pooling models are more difficult to overfit

Overfitting is a problem that has to be addressed. Alex Net has 60 million parameters, which creates a serious overfitting issue. To avoid overfitting, two tactics were used:

- Data Augmentation. The authors used a label-preserving modification to improve the diversity of their data. The training was increased 2048-fold as a result of vertical reflections and visual modifications. They also lowered the top-1 error rate by 1% by changing the brightness of RGB channels using RGB pixel values and Principal Component Analysis (PCA)
- Dropout. This method is quite likely to cause neurons to shut down (e.g., 50%). Each cycle samples a different set of model parameters, allowing each neuron to gain more stable characteristics that may later be applied to other randomly generated neurons. Dropping out, on the other hand, extends the learning curve and causes the model to lag

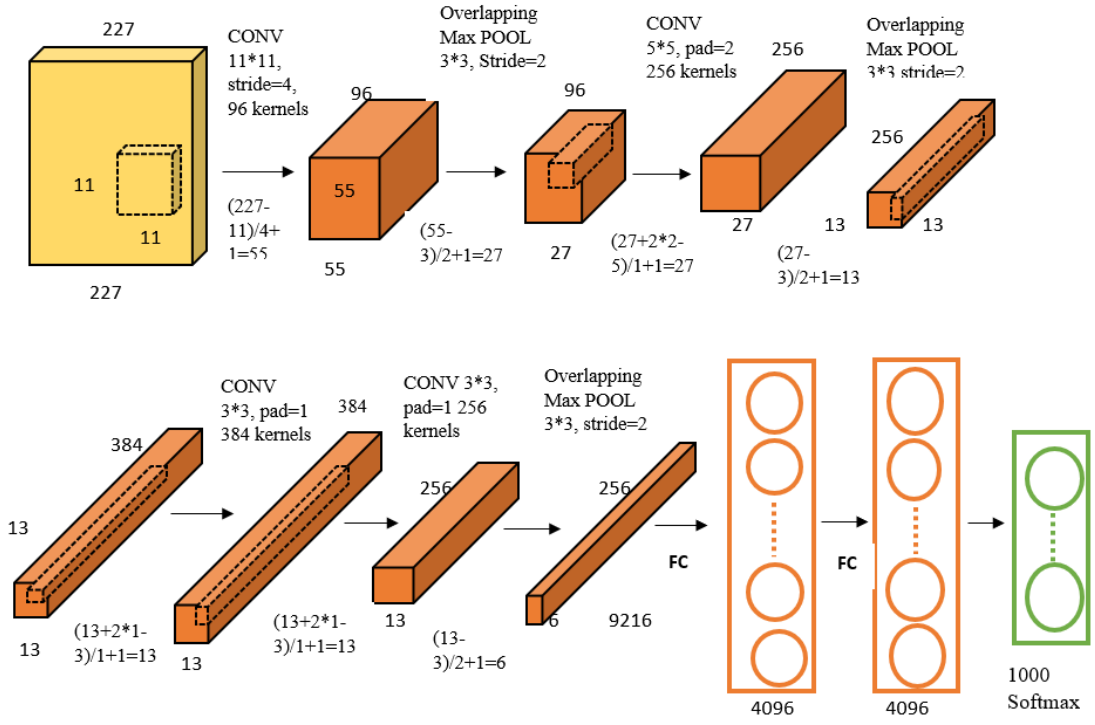


Fig. 6: Architecture of AlexNet

AlexNet is a sophisticated model capable of handling even the most difficult datasets. AlexNet's performance drops when any of the convolutional layers are removed. In the field of artificial intelligence in computer vision, AlexNet is a well-known framework for object recognition. AlexNet, rather than CNNs, may be employed for image processing in the future.

Deep Learning Technique Using ResNet

In 2015, the Microsoft Research Asia team submitted ResNet as a CNN architecture into the ImageNet competition (MSRA). In addition to winning the competition, the MSRA design established new categorization, detection, and placement criteria. The ResNet architecture relies on residual blocks, unlike AlexNet or VGG, both of which are traditional sequential network designs (Fig. 7). These blocks are employed to address the degradation problem, which happens as the network depth develops and is unrelated to overfitting.

In the network's construction, the residual block is used. A macro-architecture is made up of convolutions and pooling layers. This block's description, as well as the whole architecture, was made public by He *et al.* (2016). They showed that residual blocks may be used to train incredibly deep networks using a typical Stochastic Gradient Descent (SGD) optimizer. SGD is much faster but the convergence path of SGD is noisier than that of the original gradient descent. This is because in each step it is not calculating the actual gradient but an approximation. So, there are a lot of fluctuations in the cost. But still, it is a much better choice, and also SGD approximates the gradient using only one data point. So, evaluating the gradient saves a lot of time compared to the Adam optimizer. The design of ResNet50 is divided into four phases, as shown in the diagram (Fig. 8). A 32-bit height and width picture with a 3-bit channel width could be accepted by the network. For simplicity, we'll suppose that the input size is $224 \times 224 \times 3$. Initial convolution and max-pooling are done in each ResNet architecture with 7×7 and 3×3 kernel sizes. Following that, the network's Stage 1 continues with three three-layer residual blocks. To complete the convolution process, each of the three levels of stage 1's block requires kernels with sizes of 64, 64, and 128 bits. Curved arrows depict the relationship identification. Stride 2 is used for convolution in the residual block. As we progress from one stage to the next, the channel width doubles, and the input size is cut in half. The bottleneck design is used by advanced networks like ResNet50, ResNet152, and others. Each residual function F has three levels that are placed on top of each other. The 1×1, 3×3, and 1×1 layers all make use of convolutions. The shrinking and expanding of dimensions are handled by the 1×1 convolution layer. When the input/output dimensions are lowered, the 3×3 layer becomes a bottleneck. A 1000-neuron fully linked layer is the network's final layer, followed by an average pooling layer (ImageNet class output).

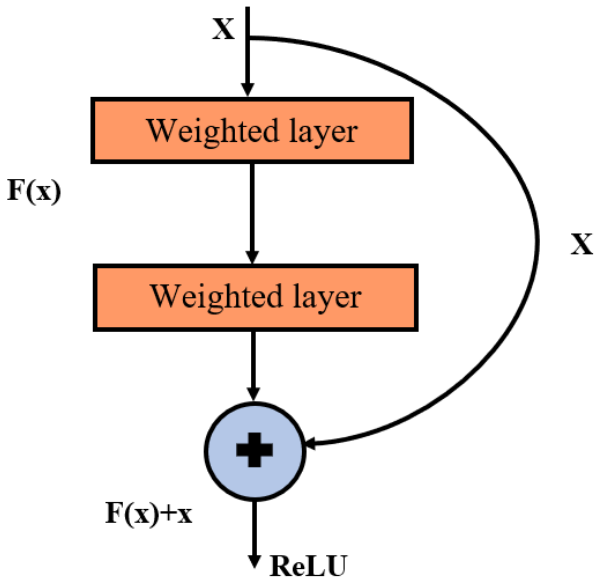


Fig. 7: Illustration of a Residual block proposed by He *et al*. (2016), where x is the input and $F(x) + x$ is the output of the residual block, before the Rectified Linear Unit (ReLU)

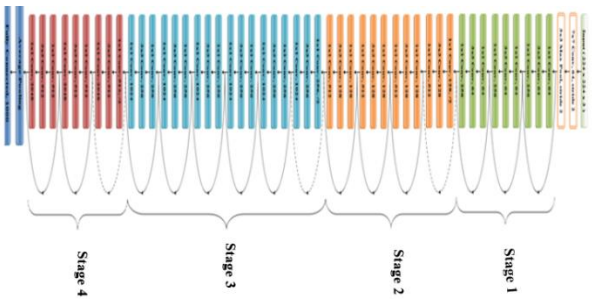


Fig. 8: Architecture of ResNet-50

Each of the five phases in the ResNet-50 model has a residual block. Each residual block is made up of three layers, each having 1*1 and 3*3 convolutions. Consider unused bricks as an example of a simple idea. Each layer feeds the next in classic neural networks. In a network with unused blocks, each tier feeds directly into the next, resulting in identity links that are 2-3 hops apart. The fused images are optimized using Harris Hawks Optimization (HHO) (Jia *et al*., 2020) and Particle Swarm Optimisation (PSO) (Shen and Niu, 2007).

Particle Swarm Optimization

Particle swarm optimization is a bionic intelligent optimization technique based on animal swarm behavior. Kennedy, an American psychologist, and Eberhart, an electrical engineer, created it in 1995. PSO is a swarm intelligence-based stochastic optimization approach that finds the optimum solution by utilizing the information-sharing mechanism of individual birds. Individual collaboration allows the PSO algorithm to recall personal best knowledge as well as global best information. Every particle in the population has the power to solve the situation at hand (Liu *et al*., 2017).

The basic premise of the typical PSO method and the procedure of obtaining the particle's best solution is then introduced.

- The initial speed and location of each particle are set by initializing a collection of particles. Fitness values are calculated by estimating the virtues and defects of the particle's current location using the fitness function
- In the next step, virtues are compared and the defect degree between the particle's current position and its ideal position is calculated. The article is moved to the particle's ideal position if the present position is better than its best
- Following, assess if the population's global ideal location is the best of all possible positions for all particles. If it isn't, change the population's global ideal location
- Finally, determine if the end condition is met by updating the speed and location of each particle. The algorithm is finished when all of the conditions are met. If you're not pleased, repeat the iteration process

The particle swarm optimization approach works like this: m particles form a group in a D-dimensional search space. The $X_i = (x_{i1}, x_{i2},\ldots, x_{id})$, i = 1,2, …represents the position of the i^{th} particle, the $V_i = (v_{i1}, v_{i2},\ldots, v_{id})$ represents the speed of the i^{th} particle, the best $P_{best} = (p_{i1}, p_{i2},\ldots, p_{id})$ indicates the ideal location for the i^{th} particle and the $g_{best} = (g_1, g_2,\ldots, g_d)$. The particle is updated using the formulas (5-6) to follow these two optimal values and the particle's speed and location are modified to fulfill the criteria of the iteration's conclusion:

$$v_{id}^{k+1} = \omega^k v_{id}^k + c_1 r_1 \left(p_{id}^k - x_{id}^k\right) + c_2 r_2 \left(g_d^k - x_{id}^k\right) \quad (5)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^k \quad (6)$$

where, $c_1, c_2 \rightarrow 1$ learning factors, $r_1, r_2 \rightarrow$ random values in the interval (0,1) and $\omega \rightarrow$ inertia weight are all variables.

When the method is used to determine high dimensional difficult optimization problems, the link between the algorithm implementation process and parameter values is greater. The PSO algorithm's convergence speed is faster during the prophase of iteration. The local search speed slows down during the anaphase of iteration when the algorithm converges to a local minimum. This problem can be efficiently solved using the accelerated particle swarm optimization technique. Following that, we provide an enhanced

Particle Swarm Optimization approach (APSO). Since individual best is employed to improve the variety in the quality solutions, the accelerated particle swarm optimization employs just the global best position to update the velocity v_{id}^{k} *v*. As a result, the i^{th} particle's velocity is updated as follows:

$$v_{id}^{k+1} = v_{id}^{k} + c_1\epsilon + c_2\left(g_d^k - x_{id}^k\right) \tag{7}$$

where, $\epsilon\rightarrow$ is a random number between 0 and 1. The i^{th} particle's location is updated using the following equation:

$$x_{id}^{k+1} = \left(1-c_2\right)x_{id}^{k} + c_1\epsilon + c_2 g_d^k \tag{8}$$

The third parameter variable is introduced, which is based on the APSO algorithm. This variable can be combined with the other two variables to create a new c_1 and c_2 solution. The search space exploration of the particles is governed by the $c_1\theta$ function. The mathematical expression for $c_1\theta$ is as follows:

$$c_1\theta = c_{1\max} - \frac{c_{1\max} - c_{1\min}}{\theta_{\max}}\theta \tag{9}$$

$\theta_{\max}$ is the maximum number of Iterations

Furthermore, the function $c_2\theta$ is regarded as an increasing function, with the following expression:

$$c_2\theta = c_{2\max} - \left(c_{2\max} - c_{2\min}\right)\tan\left(\frac{\pi}{2}\times\frac{\theta}{\theta_{\max}}\right) \tag{10}$$

where, $c_{2\min}$ and $c_{2\max}$ are the smallest and maximum values of $c_2\theta$ at the first and last iterations, respectively. During the search process, the function $c_2\theta$ strikes a balance between global and local explorations.

We may update the particle's location equation and velocity equation using formulas 9-10 as follows:

$$v_{id}^{k+1} = v_{id}^{k} + c_1\left(\theta\right)\epsilon + c_2\left(\theta\right)\left(g_d^k - x_{id}^k\right) \tag{11}$$

$$x_{id}^{k+1} = \left(1-c_2\left(\theta\right)\right)x_{id}^{k} + c_1\left(\theta\right)\epsilon + c_2\left(\theta\right)g_d^k \tag{12}$$

Harris Hawks Optimization

The chasing behavior of Harris hawks inspired HHO, a population-based, gradient-free optimization approach. Ma *et al*. (2019) just released an essay on HHO. The algorithm is built on preaching, predation, and surprise pounces, which are all-natural predatory behaviors of Harris hawks. Exploration and exploitation are the two parts of HHO, like other meta-heuristic algorithms. HHO, on the other hand, has two exploration stages and four exploitation stages, as indicated below.

The first stage involves defining the goal function and related solution space. The starting population is produced when the parameter values are assigned. When seeking prey, harris hawks are in the exploration phase (the rabbit). Although detecting prey is often challenging, hawks have an extraordinary vision that allows them to discover and track prey. In this example, the hawks are keeping an eye on the situation in the hopes of catching a sight of the prey. In practice, each cycle's candidate solutions are all Harris hawks and their fitness values are calculated based on predicted prey. The Harris hawks may then wait for the prey in various locations, as stated in the equation below:

$$X\left(t=1\right)\begin{cases} X_{rand}\left(t\right) - r_1\left|X_{rand}\left(t\right) - 2r_2 X\left(t\right)\right| q \geq 0.5 \\ \left(X_{rabbit}\left(t\right) - X_m\left(t\right)\right) - r_3\left(LB + r_4\left(UB - LB\right)\right) q \leq 0.5 \end{cases} \tag{13}$$

In this equation, $X(t + 1)$ represents the rabbit's current position vector, $X(t)$ indicates the average position of the current population, $X(t)$ represents the hawks' current position vector and, , , , and q(wait) are random values within [0, 1], while LB and UB represent the problem variables. HHO employs the following equation to compute the average location of hawks $X_m(t)$:

$$X_m\left(t\right) = \frac{1}{N}\sum_{i=1}^{n} X_i\left(t\right) \tag{14}$$

$X_i(t)$ denotes the current location of the hawks in iteration *t*, whereas *N* represents the total number of hawks.

The transition from exploration to exploitation: Exploration and exploitation must be balanced correctly for meta-heuristic algorithms to work well. HHO uses the prey escape energy to transition from exploration to exploitation, as well as between different sorts of Exploitations (E). The rabbit's energy is depleted while evading the hawks, according to HHO, which may be computed as follows:

$$E = E_0\left(1 - \frac{t}{T}\right) \tag{15}$$

T is the maximum number of repeats and E_0 is the starting energy state, whose value varies at random over the span (-1, 1) when the rabbit's escaping energy $|E| \geq 1$, (exploration phase). During the exploitation phase $|E| < 1$, when the hawks' energy is spent, they hunt for a solution in their surroundings.

Exploitation phase: In this phase, Harris hawks assault the prey, using the previous phase's position as a guide.

The rabbit, on the other hand, is continually attempting to flee and the hawks pursue him. As a result, HHO relies on four distinct attacking strategies. The variables r and $|E|$ decide which strategy will be utilized. The rabbit's fleeing energy is represented by $|E|$ and the possibility of escaping is represented by r, with r<0:5 indicating a higher chance of success and r≥0:5 indicating failure.

HHO's worldwide search ability is improved with the EOBL approach. The following is a definition of the opposing point: The elite opposite point $X_i = (x_{i1}, x_{i2},\ldots,x_{id})$ in the current population $X_e = (x_{e1}, x_{e2}, \ldots, x_{ed})$, for the individual $\breve{X}_i = \left(\breve{x}_{i1}, \breve{x}_{i2},..., \breve{x}_{iD}\right)$.

$$\breve{x}_{i,i} = s \times \left(da_j + ab_j - x_{i,j}\right) \tag{16}$$

where, $s \in [a_i, b_i]$, $S \in U[0,1]$ S is a factor that may be used in a wide range of situations. Dynamic boundaries are da_j and db_j, which may be defined as:

$$da_j = \min\left(x_{i,j}\right), db_j = \max\left(x_{i,j}\right) \tag{17}$$

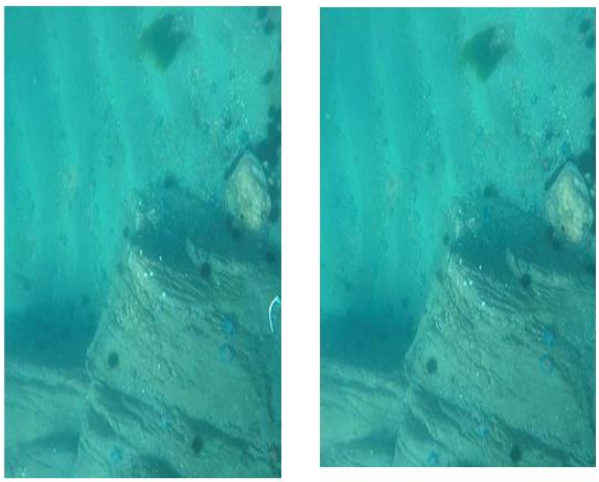
The inverse, on the other hand, can go beyond the search limit $[a_i, b_i]$. To resolve this issue, the changed person is given a random value inside $[a_i, b_i]$ as follows:

$$\breve{x}_{i,j} = rand\left(a_j, b_j\right), if\ \breve{x}_{i,j} < a_j \,||\, \breve{x}_{i,j} > b_j \tag{18}$$

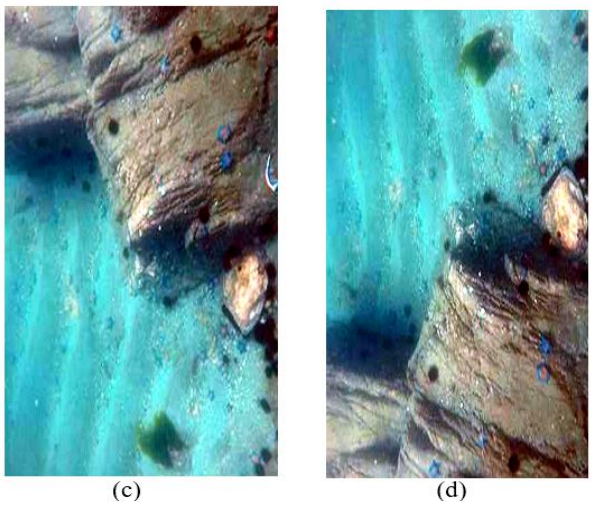
HHO uses rabbit energy $|E|$ to bridge the gap between exploration and exploitation and to define the present exploitation strategy. Rabbit energy is also employed to deter hawks away from the local maxima. Hawks, on the other hand, may become caught in local optima if the rabbit escaping energy quickly changes its convergence to the optimal solution (Zhang *et al*., 2016). This paragraph summarizes the Three Suggested Search strategies (TSS) for enhancing the HHO algorithm's global and local search processes (mutation, Mutation Neighborhood Search (MNS), and rollback method). The benefit of HHO based deep learning is feature concatenation, which allows us to learn features at any step without compressing them, as well as the capacity to manage and manipulate those features. This method allows us to prevent parameter number explosion, which reduces the complexity of the training process and eliminates the problem of overfitting.

Experimental Results and Discussion

The Proposed work is implemented in the environment of MATLAB 2018a. To evaluate all presented methods, we have used underwater image datasets that are mentioned in above section. Here we are sampling a pair of images that undergoes fusion and optimization. The underwater images are utilized to evaluate the suggested fusion method. The experimental results of the original underwater image, color-corrected image, and reconstructed images are shown in Fig. 9(a-b) underwater images. Figure 9(c-d) demonstrates the color correlation findings from images 1 and 2. The low pass findings, as well as coverage distributions for both images, are shown in Fig. 9(e-f). The covering distribution of images 1 and 2 are shown in Fig. 9(g-h) and 10(a-b) shows the reconstructed image (b). Figure 11 displays the fused images with rich characteristics including smooth curves and geometric structures created with the innovative Tetrolet transform. Then the deep learning algorithms are performed in the fusion process and from that best one is selected. Finally, the image is classified as good quality and poor quality. These merged images are superior to any of the originals. The values of performance measures for various acceptable coverings are shown in Table 2.

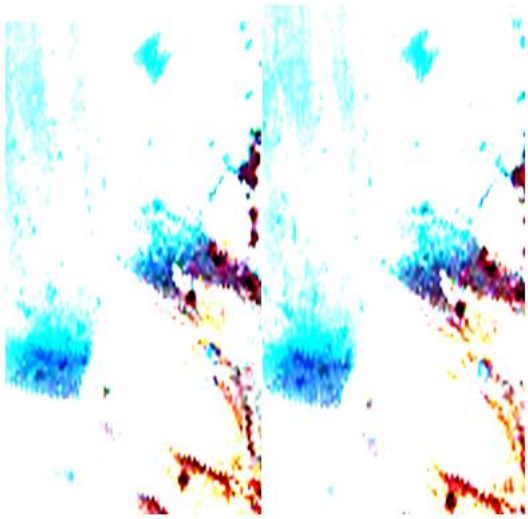


(a) (b)

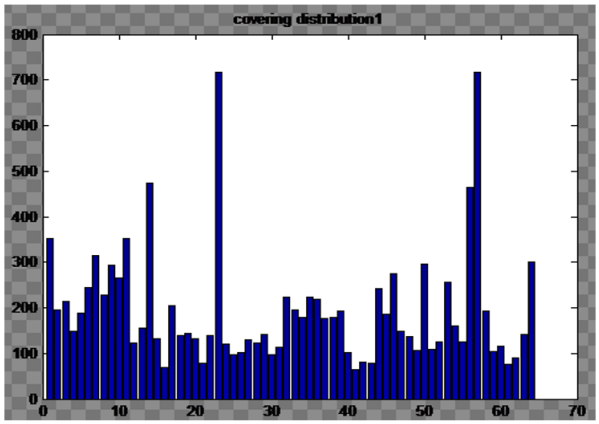


(c) (d)

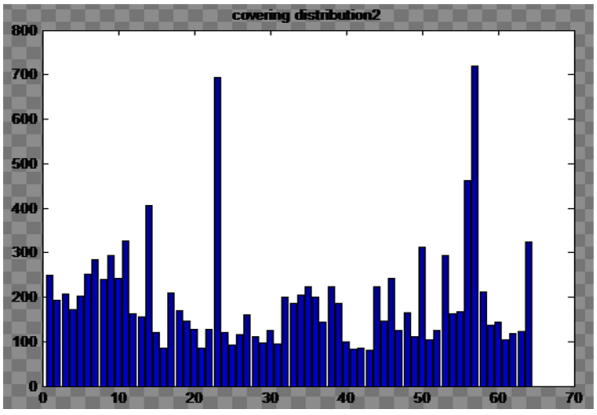
DOI: 10.3844/jcssp.2023.467.482



(e) (f)



(g)



(h)

Fig. 9: Experimental results: (a). Input image 1 (b). Input image 2 (c). Dehazed image of input image1 (d) Dehazed image of input image 2, (e) Low pass result of input image 1, (f) Low pass result of input image 1, (g) and (h). Covering distribution of images 1 and 2

(a)

(b)

Fig. 10: (a-b) reconstructed images of images 1-2

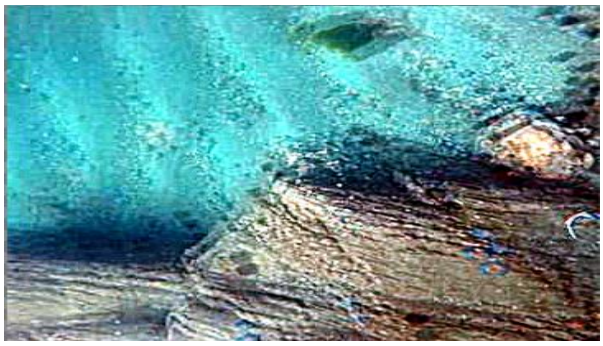


Fig. 11: Fused image

Performance Measures

True Positive (TP): You made a good forecast and it turned out to be correct
True Negative (TN): You expected something negative and it turned out to be true
False Positive (FP): You anticipated something positive to happen, but it didn't
False Negative (FN): You predicted a negative but it didn't come true

Accuracy: This is a measure of how much we predicted the classes accurately:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \times 100\% \quad (19)$$

Specificity: Relates to the true negative rate of the considered class:

$$Specificity = \frac{TN}{TN + FP} \times 100\% \quad (20)$$

Sensitivity: It is a quantitative or completeness measure. Simply put, a strong recall algorithm returns the majority of the important outcomes:

$$Sensitivity = \frac{TP}{TP + FN} \times 100\% \quad (21)$$

Precision: It is a measure of precision or quality. Greater precision implies that an algorithm returns considerably more relevant outcomes than unimportant ones:

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (22)$$

F-measure: It is a measure that allows you to compare recall and precision at the same time:

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (23)$$

Table 2 depicts the overall performance measures for images and Figs. 12-13. Figure 12 depicts the graphs for performance measures for images of good-quality and poor-quality images respectively. From the result, the fused image in Fig. 10 is of good quality based on the classification result by various DL models. This model attains the highest accuracy of 96.32%.

Table 3 the comparison results are evaluated by GoogleNet, AlexNet, and ResNet with hybrid HHO + PSO performance metrics, and the evaluation results are pointed out. The highest evaluation metric result indicates the superiority of the proposed classification. In the ResNet + PSO + HHO metric, this method is better than the comparison method in the GoogleNet and AlexNet metric after integrating the uneven color blur and low contrast of the underwater image. The proposed method has a better performance than the existing works. In the future, the proposed algorithm is also applied to multifocal image fusion such as infrared and visible image fusion and other medical modality image fusion. While evaluating our model, we obtained much greater fusion and classification accuracy performance compared to other models. From the visual effects and objective measures, the contrast experiments show that the proposed algorithm has better performance than the compared algorithms.

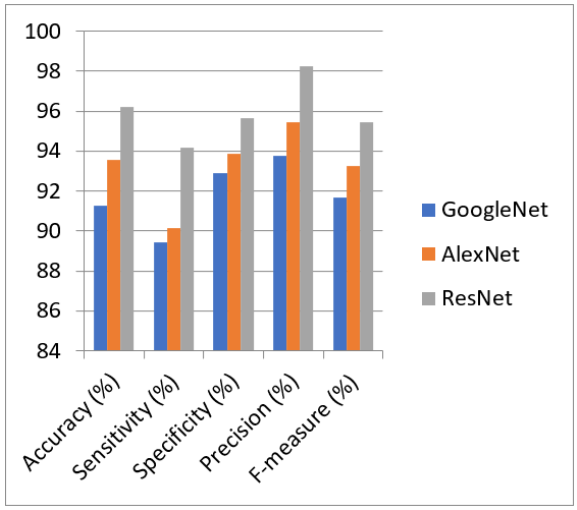


Fig. 12: Performance metrics for good-quality images

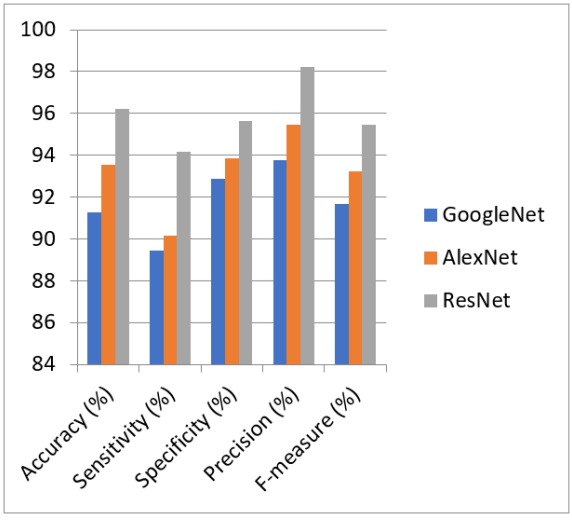


Fig. 13: Performance metrics for poor-quality images

Table 2: Overall performance measures for image 1

Images	Classification methods	Accuracy (%)	Sensitivity (%)	Specificity (%)	Precision (%)	F-measure (%)
Poor quality	Google Net	91.25	89.45	92.88	93.78	91.66
	Alex Net	93.56	90.15	93.86	95.44	93.23
	ResNet + PSO + HHO	96.22	94.15	95.63	98.24	95.45
Good quality	Google Net	91.35	89.55	92.98	93.88	91.76
	Alex Net	93.66	90.25	93.96	95.54	93.33
	ResNet + PSO + HHO	96.32	94.25	95.73	98.34	95.55

Table 3: Comparison of the proposed model with the existing model

References	Author	Results (%)
Tao *et al*. (2020)	Tao	78.94
Shahdoosti and Mehrabi (2018)	Reza	83.00
Liu *et al*. (2018)	Liu	91.50
Li *et al*. (2019)	Li	95.00
Proposed (good quality)	GoogleNet	91.35
	AlexNet	93.66
	ResNet + PSO + HHO	96.32
Proposed (poor quality)	Google Net	91.25
	AlexNet	93.56
	ResNet + PSO + HHO	96.22

Conclusion

Innovations have been incorporated into image processing. This study summarizes the overall analysis of the fusion and classification approach. The underwater images are obtained from the open access dataset, this dataset is fused with each other to enhance the classification accuracy. During this study, UIQS, UCCS, and UTTS datasets were used for research, color correction was done by the dehazing color correlation algorithm and haze was removed by a procedure relying on DCP. Then the image decomposition filter bank algorithm has been employed on the images. The fused images were classified by DL methods namely GoogleNet, ResNet, and AlexNet in which they classify the fused images to get better images compared to others. ResNet50 responds to classification very well than others. Image fusion using ResNet gave better results when compared to GoogleNet and AlexNet. All the performance parameters such as accuracy, sensitivity, specificity, precision, and F-measure were analyzed. These proposed model performances are compared with the existing systems with the standard parameters. The results show that the proposed system provides high accuracy which is much higher than the existing fusion methods.

Acknowledgment

The research was done as part of my (first author) Ph. D. work, with second author as my Internal Supervisors.All the experiments are conducted in *Noorul Islam Centre for Higher Education Kumaracoil*, Tamilnadu, India.

Funding Information

This research received no specific grant from any funding agency in the public, commercial, or not for profit sectors.

Author's Contributions

Devika Sarath: Investigation, problem formulation, methodology, formal analyzed, software implementation, data analyzed, original drafted paper preparation.

M. Sucharitha: Supervision, designed research plan, research administration, problem formulation, methodology, resources, written reviewed, drafted paper correction and edited.

Ethics

This article is an original research work. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

Amarsaikhan, D., Saandar, M., Ganzorig, M., Blotevogel, H. H., Egshiglen, E., Gantuyal, R., ... & Enkhjargal, D. (2012). Comparison of multisource image fusion methods and land cover classification. *International Journal of Remote Sensing*, *33*(8), 2532-2550. https://doi.org/10.1080/01431161.2011.616552

Bhagyasri, G., Prasannakumar, G., & Murthy, P. S. N. (2019, May). Underwater image enhancement using SWT based image fusion and colour correction. In 2019 International Conference on Intelligent Computing and Control Systems (ICCS) (pp. 749-754). IEEE. https://ieeexplore.ieee.org/abstract/document/9065524

Chang, H. H. (2020). Single underwater image restoration based on adaptive transmission fusion. *IEEE Access*, *8*, 38650-38662. https://doi.org/10.1109/ACCESS.2020.2971019

Dai, C., Lin, M., Wang, Z., Zhang, D., & Guan, Z. (2018). Color compensation based on bright channel and fusion for underwater image enhancement. *Acta Optica Sinica*, *38*(11), 1-10. https://doi.org/10.3788/AOS201838.1110003

Ell, T. A., & Sangwine, S. J. (2006). Hypercomplex Fourier transforms of color images. *IEEE Transactions on Image Processing*, *16*(1), 22-35. https://doi.org/10.1109/TIP.2006.884955

Gao, F., Wang, K., Yang, Z., Wang, Y., & Zhang, Q. (2021). Underwater image enhancement based on local contrast correction and multi-scale fusion. *Journal of Marine Science and Engineering*, *9*(2), 225. https://doi.org/10.3390/jmse9020225

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770-778).

Jia, H., Peng, X., Kang, L., Li, Y., Jiang, Z., & Sun, K. (2020). Pulse coupled neural network based on Harris hawks optimization algorithm for image segmentation. *Multimedia Tools and Applications*, *79*, 28369-28392. https://doi.org/10.1007/s11042-020-09228-3

Kanagavel, M., Sivakumar, P., & Bama, S. (2021, December). Image Fusion Based Selective Optimal Restoration of Colors in Single Underwater Images. In *2021 International Symposium on Ocean Technology (SYMPOL)* (pp. 1-6). IEEE. https://doi.org/10.1109/SYMPOL53555.2021.9689288

Kaur, M., & Singh, D. (2020). Fusion of medical images using deep belief networks. *Cluster Computing*, *23*, 1439-1453. https://doi.org/10.1007/s10586-019-02999-x

Lewis, J. J., O'Callaghan, R. J., Nikolov, S. G., Bull, D. R., & Canagarajah, N. (2007). Pixel-and region-based image fusion with complex wavelets. *Information Fusion*, *8*(2), 119-130. https://doi.org/10.1016/j.inffus.2005.09.006

Lewis, J. J., O'callaghan, R. J., Nikolov, S. G., Bull, D. R., & Canagarajah, C. N. (2004, June). Region-based image fusion using complex wavelets. In *Proc. 7th International Conference on Information Fusion* (Vol. *1*, pp. 555-562).

Li, H., & Zhuang, P. (2021). DewaterNet: A fusion adversarial real underwater image enhancement network. *Signal Processing: Image Communication*, *95*, 116248. https://doi.org/10.1016/j.image.2021.116248

Li, H., Wu, X. J., & Durrani, T. S. (2019). Infrared and visible image fusion with ResNet and zero-phase component analysis. *Infrared Physics & Technology*, *102*, 103039. https://doi.org/10.1016/j.infrared.2019.103039

Li, Y., & Wang, S. (2010, October). Underwater object detection technology based on polarization image fusion. In *5th International Symposium on Advanced Optical Manufacturing and Testing Technologies: Optoelectronic Materials and Devices for Detector, Imager, Display and Energy Conversion Technology* (Vol. *7658*, pp. 1229-1233). SPIE.https://doi.org/10.1117/12.867624

Lin, S., Chi, K. C., Li, W. T., & Tang, Y. D. (2020). Underwater optical image enhancement based on dominant feature image fusion. *Acta Photonica Sinica*, *49*(3), 310003. https://doi.org/10.3788/gzxb20204903.0310003

Linfeng, B., Zhang, W., Pan, X., & Zhao, C. (2020). "Underwater image enhancement based on global and local equalization of histogram and dual-image multi-scale fusion." *IEEE Access 8*, 128973-128990. https://doi.org/10.1109/ACCESS.2020.3009161

Liu, D., Wen, B., Liu, X., Wang, Z., & Huang, T. S. (2017). When image denoising meets high-level vision tasks: A deep learning approach. *arXiv preprint arXiv:1706.04284*. https://doi.org/10.24963/ijcai.2018/117

Liu, Y., Chen, X., Wang, Z., Wang, Z. J., Ward, R. K., & Wang, X. (2018). Deep learning for pixel-level image fusion: Recent advances and future prospects. *Information Fusion*, *42*, 158-173. https://doi.org/10.1016/j.inffus.2017.10.007

Luo, W., Duan, S., & Zheng, J. (2021). Underwater image restoration and enhancement based on a fusion algorithm with color balance, contrast optimization and histogram stretching. *IEEE Access*, *9*, 31792-31804. https://doi.org/10.1109/ACCESS.2021.3060947

Ma, J., Yu, W., Liang, P., Li, C., & Jiang, J. (2019). FusionGAN: A generative adversarial network for infrared and visible image fusion. *Information fusion*, *48*, 11-26. https://doi.org/10.1016/j.inffus.2018.09.004

Shahdoosti, H. R., & Ghassemian, H. (2016). Combining the spectral PCA and spatial PCA fusion methods by an optimal filter. *Information Fusion*, *27*, 150-160. https://doi.org/10.1016/j.inffus.2015.06.006

Shahdoosti, H. R., & Mehrabi, A. (2018). Multimodal image fusion using sparse representation classification in tetrolet domain. *Digital Signal Processing*, *79*, 9-22. https://doi.org/10.1016/j.dsp.2018.04.002

Shen, L., & Niu, Y. (2007). Blind color image fusion based on the optimal multi-objective particle swarm optimization. *International Journal of Multimedia and Ubiquitous Engineering*, *2*(2), 51-61. https://doi.org/10.1109/MUE.2007.204

Sreeja, P., & Hariharan, S. (2019). Three-dimensional fusion of clustered and classified features for enhancement of liver and lesions from abdominal radiology images. *IET Image Processing*, *13*(10), 1680-1685. https://doi.org/10.1049/iet-ipr.2018.5158

Tang, H., Xiao, B., Li, W., & Wang, G. (2018). Pixel convolutional neural network for multi-focus image fusion. *Information Sciences*, *433*, 125-141. https://doi.org/10.1016/j.ins.2017.12.043

Tang, P., Wang, H., & Kwong, S. (2017). G-MS2F: GoogLeNet based multi-stage feature fusion of deep CNN for scene recognition. *Neurocomputing*, *225*, 188-197. https://doi.org/10.1016/j.neucom.2016.11.023

Tao, Y., Dong, L., & Xu, W. (2020). A novel two-step strategy based on white-balancing and fusion for underwater image enhancement. *IEEE Access*, *8*, 217651-217670. https://doi.org/10.1109/ACCESS.2020.3040505

Wang, H., Li, S., Song, L., & Cui, L. (2019). A novel convolutional neural network-based fault recognition method via image fusion of multi-vibration-signals. *Computers in Industry*, *105*, 182-190. https://doi.org/10.1016/j.compind.2018.12.013

Yang, H. H., Huang, K. C., & Chen, W. T. (2021, May). Laffnet: A lightweight adaptive feature fusion network for underwater image enhancement. In *2021 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 685-692). IEEE. https://doi.org/10.1109/ICRA48506.2021.9561263

Yuan, Z. W., & Zhang, J. (2016, August). Feature extraction and image retrieval based on AlexNet. In *Eighth International Conference on Digital Image Processing (ICDIP 2016)* (Vol. *10033*, pp. 65-69). SPIE. https://doi.org/10.1117/12.2243849

Zhang, Q., Liu, Y., Blum, R. S., Han, J., & Tao, D. (2018). Sparse representation based multi-sensor image fusion for multi-focus and multi-modality images: A review. *Information Fusion*, *40*, 57-75. https://doi.org/10.1016/j.inffus.2017.05.006

Zhang, Y., Chen, L., Zhao, Z., & Jia, J. (2016). Multi-focus image fusion based on cartoon-texture image decomposition. *Optik*, *127*(3), 1291-1296. https://doi.org/10.1016/j.ijleo.2015.10.098

Zhu, D., Liu, Z., & Zhang, Y. (2021). Underwater image enhancement based on colour correction and fusion. *IET Image Processing*, *15*(11), 2591-2603. https://doi.org/10.1049/ipr2.12247