Optimal Model Architecture of Convolutional Neural Networks with Genetic Algorithms

Manel Roumaissa Benabid[1*]

[1*]Department of Computer Science and Mathematics, Lebanese American University, Byblos, Lebanon.

Corresponding author(s). E-mail(s): manelroumaissa.benabid@lau.edu;

Abstract

Convolutional Neural Networks have been widely used in many fields and precisely in computer vision for tasks such as image classification, object detection, and event detection. In the present day, there are several state-of-the-art CNN models that were shown to perform very well at image classification on standard very famous datasets such as ImageNet, CIFAR10, CIFAR100, and MNIST. Through the power of Transfer Learning, researchers were able to leverage the power of these state-of-the-art networks to do other image classification tasks. This paper proposes a new approach that aims at finding the best ensemble model architecture of Convolutional Neural Networks using the power of Transfer Learning from several CNNs to classify images. The search for the optimal model was optimized using Genetic Algorithms.

Keywords: Convolutional Neural Networks, Transfer Learning, Image classification, Genetic Algorithms

1 Introduction

Convolutional neural networks (CNNs) are a type of deep neural network that is particularly well-suited for image classification tasks. CNNs consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers. Convolutional layers apply filters to an input image, extracting features that are important for classification. Pooling layers reduce the spatial dimensionality of the feature maps produced by the convolutional layers, allowing for more efficient processing. Fully connected layers take the extracted features and perform the final classification.

CNNs have been widely used for image classification tasks, achieving state-of-the-art performance on datasets such as ImageNet. The ability of CNNs to automatically learn useful representations of image data has led to their use in many medical imaging applications. For example, CNNs have been used for segmentation tasks, such as segmenting brain tumors from MRI scans.

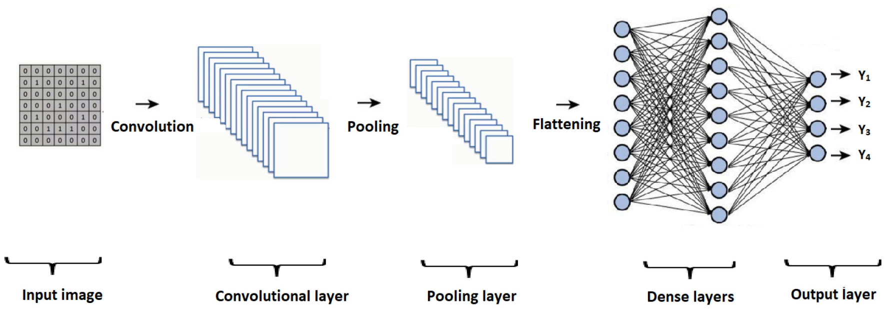

Fig. 1 Basic CNN architecture [1]

Integration of CNNs in medical imaging segmentation has been shown to improve accuracy and efficiency compared to traditional segmentation methods. For example, a study by Kamnitsas et al. (2017) demonstrated that a 3D U-Net architecture achieved state-of-the-art performance for brain tumor segmentation from MRI scans.

Transfer learning is a machine learning technique in which a model trained on one task is used to improve the performance of another related task. The idea is to use the knowledge gained from training on a large dataset to improve the performance of a model on a smaller dataset, which may not have enough labeled examples to train a good model from scratch.

Transfer learning works by taking a pre-trained model, usually on a large dataset, and fine-tuning it on a smaller dataset for the task at hand. The pre-trained model's learned features can be used as a starting point for the new model, which can then be fine-tuned on the smaller dataset.

Transfer learning has been used for various tasks, including image classification, object detection, natural language processing, and speech recognition. In image classification, transfer learning has been used to improve the accuracy of models on smaller datasets, especially in the medical domain, where datasets may be small and labeled examples are scarce.

One example of a paper that used transfer learning for medical image classification is the work of Shin et al. (2016), which used a pre-trained CNN model to classify skin lesions as benign or malignant. The pre-trained CNN model was fine-tuned on a small dataset of skin lesion images, and the resulting model achieved state-of-the-art performance on the task.

Genetic algorithms (GA) are a type of optimization algorithm inspired by the principles of natural selection and genetics. GA was first proposed by John Holland in the 1960s as a method to solve complex optimization problems. The basic idea behind

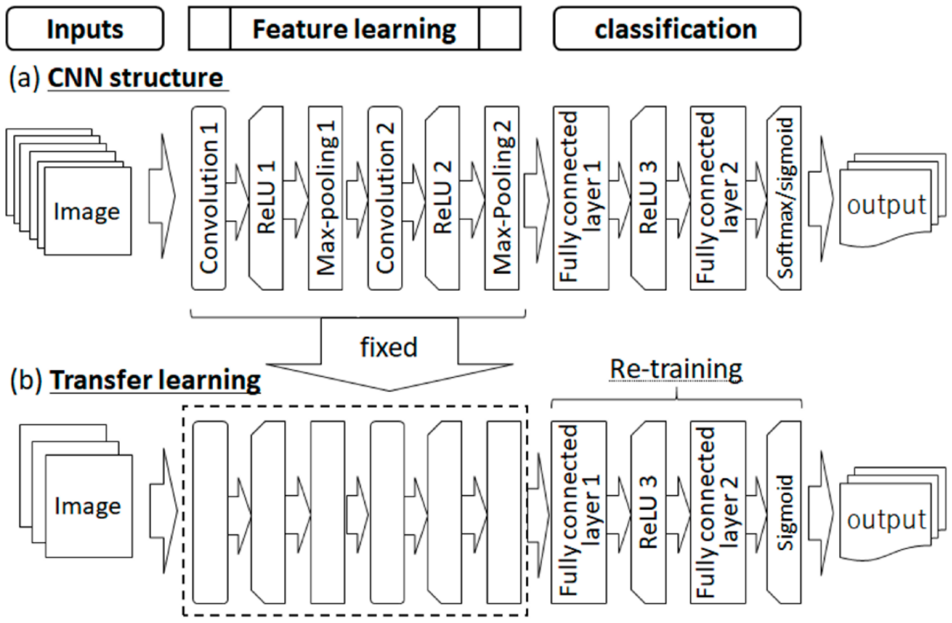

Fig. 2 Transfer Learning [2]

GA is to create a population of potential solutions to a problem, and then evolve this population over generations by selecting the fittest individuals and recombining their genetic material to create new solutions.

The process of GA starts with the creation of an initial population of potential solutions. These solutions are represented as strings of genetic material, which are typically binary strings, although other encodings can also be used. The fitness of each solution is evaluated using an objective fitness function, which quantifies how well a particular solution solves the problem at hand.

In the next step, the fittest individuals in the population are selected to reproduce. This selection is usually performed using a fitness-based selection method, such as rank selection or roulette wheel selection. Once the fittest individuals are selected, their genetic material is combined using a crossover operator to create new solutions. This process of selection and recombination is repeated for several generations until a satisfactory solution is found.

GA has been successfully applied to a wide range of optimization problems in various fields, including engineering, finance, and computer science.

2 Literature Review

Image classification plays a crucial role in various domains, including computer vision, healthcare, and autonomous systems. Over the years, Convolutional Neural Networks (CNNs) have emerged as powerful tools for image classification tasks. In recent

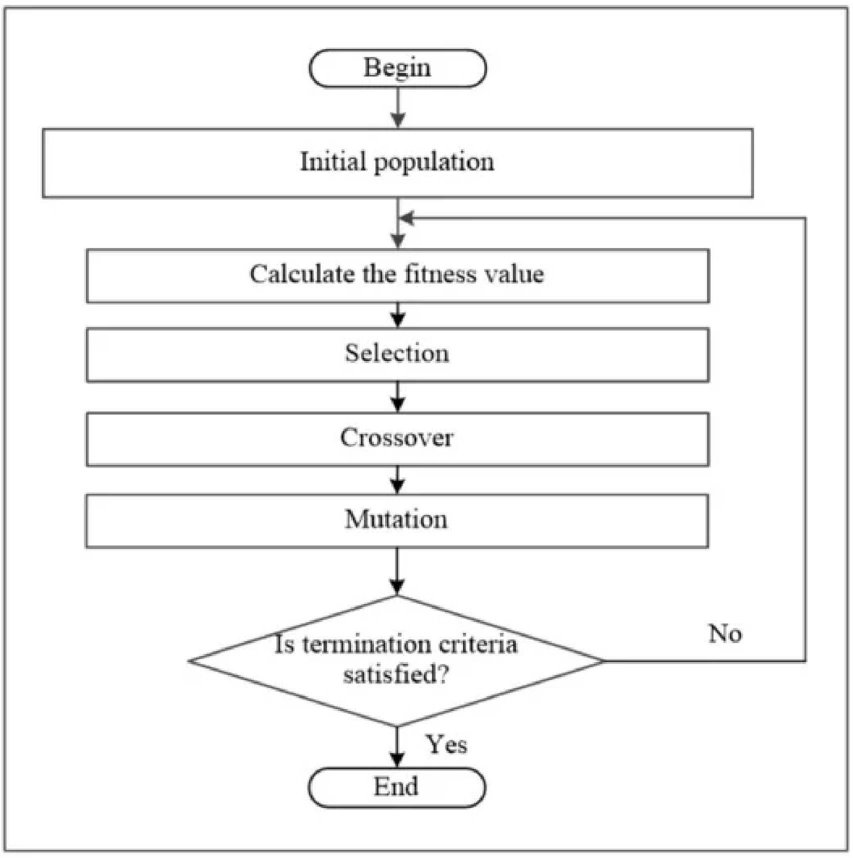
Begin
Initial population
Calculate the fitness value
Selection
Crossover
Mutation
Is termination criteria satisfied?
No
Yes
End

Fig. 3 GA architecture [3]

years, researchers have explored different methodologies to improve the performance of CNNs, with two prominent approaches being the utilization of CNNs directly and the application of transfer learning techniques. This literature review aims to summarize key papers that have employed CNNs and transfer learning for image classification tasks.

CNNs have been extensively employed in image classification due to their ability to automatically learn hierarchical features from raw pixel data. Krizhevsky et al. [4] introduced AlexNet, a groundbreaking CNN architecture, which achieved top performance in the ImageNet [5] Large Scale Visual Recognition Challenge. Their work demonstrated the potential of CNNs in image classification tasks, paving the way for subsequent advancements. Following this, Simonyan and Zisserman [6] proposed the VGG network architecture, which incorporated deeper networks and achieved improved accuracy in image classification benchmarks.

More recently, He et al. [7] introduced the ResNet architecture, which utilized residual connections to enable the training of extremely deep CNNs. ResNet achieved superior performance on various image classification datasets, addressing the issue of vanishing gradients encountered in deeper networks. Similarly, Huang et al. [8] proposed DenseNet, a CNN architecture that connected each layer to every other layer in a feed-forward fashion, facilitating feature reuse and further enhancing classification accuracy.

Transfer learning has gained significant attention as a technique to leverage pre-trained CNN models and adapt them to new image classification tasks. Yosinski et al. [9] explored the effectiveness of transfer learning by fine-tuning pre-trained CNNs on new datasets. They demonstrated that pre-trained CNNs, initially trained on large-scale datasets such as ImageNet, can significantly improve performance even when applied to different image classification tasks.

Moreover, Donahue et al. [10] introduced the concept of "DeCAF" features, which are high-level representations learned from CNNs. By extracting and utilizing these features, they achieved state-of-the-art results on multiple image classification tasks. This work further established the potential of transfer learning in image classification using CNNs.

CNNs have been used in many fields such as image classification, object detection, and text recognition. Lu [11] used Convolutional Neural Networks with data augmentation to classify 5822 RGB images in 10 types of food classes from ImgeNet [5], they compare their method against BoF or bag-of-feature and Support vector machines, their model obtained an accuracy of 74% without data augmentation and 90% with data augmentation while BoF and SVM gave an accuracy of 74%. Wang and Chen [12] designed an 8-layer CNN using ReLU activation function with dropout layers. They used a fruit dataset with 3600 images of 18 types. They obtained an accuracy of 95%. Another paper by Jia et al. [13] used Haar wavelet entropy with multiplayer perceptron as a classifier and standard genetic algorithms to train the classifier to detect and identify fruit categories. They obtained an accuracy of 81% which were better than the results of backpropagation gradient descent with an accuracy of 74% and simulated annealing of 78%. Convolutional Neural Networks have been widely used for medical image classification due to their robustness. A paper by Yadav and Jadhav [14] proposed a deep CNN based medical image classification for disease diagnosis using transfer learning.

Genetic Algorithms (GAs) have emerged as powerful optimization techniques for improving the performance of neural networks. By combining principles from evolutionary biology and optimization, GAs offer an effective approach to finding optimal or near-optimal solutions in complex search spaces.

Optimizing the architecture of neural networks is crucial for achieving optimal performance. GAs have been employed to automatically search for optimal network architectures. Stanley and Miikkulainen [15] introduced NeuroEvolution of Augmenting Topologies (NEAT), which combines the power of GAs with the ability to evolve neural network structures. NEAT demonstrated the capability to discover network architectures that outperformed hand-designed architectures in various tasks.

In addition to optimizing neural network architectures and weights, GAs have been utilized for hyperparameter tuning. Hyperparameters significantly impact the performance of neural networks, and manually tuning them can be time-consuming and challenging. Ayan [16] applied GAs to optimize hyperparameters, such as freezing ratio, dropout rate, optimizer, and network size. The results demonstrated that GAs could effectively search the hyperparameter space and discover optimal configurations, improving the performance of neural networks.

With the objective of forecasting illness that can affect grapes and tomato leaves at an early stage, Paymode and Malode [17] designed a model of CNN methods to detect Multi-Crops Leaf Disease (MCLD). They used VGG model to improve performance measure. Their designed model achieved an accuracy of 98.4% on grapes and 95.71% on tomatoes. In another paper by Chakraborty et al. [18] COVID-19, Pneumonia, and healthy cases were classified from X-ray images using a transfer learning approach with VGG19 model on a dataset of 3797 images. They achieved an accuracy of 97.11% and a precision and recall of 97%. Kumar, Prasad, and Metan [19] designed a hybrid Deep-CNN with Nature-inspired ResNet 152 Transfer Learning to classify brain images as normal. benign tumors, and malignant tumors. The weight parameters were optimized using Covid-19 optimization algorithm (CoV-19 OA). Their model obtained an accuracy of 99.57%. In a similar context, Showkat and Qureshi [20] aim at assessing the performance of ResNet models in classifying pneumonia from CXR images and to build a customized ResNEt model to evaluate its contribution to the performance improvement. All the models achieved accuracies higher than 90%, the customized model achieved 95% accuracy with 95.65% precision.

In their paper, Kumar and Janet [21] tackle the challenge of distinguishing malware programs with the power of transfer learning. They leveraged the power of Deep Transfer Learning for Malware Image Classification. They achieved an accuracy of 98.92% on the MalImg dataset and 93.19% for Microsoft datasets, their method outperformed VGG16, VGG19, ResNet50, and Google's inceptionV3. For the purpose of identifying the freshness of fruits in the food industry, Kazi and Panda [22] utilized an image classification approach using CNNs and Transfer Learning. The purpose was to identify three types of fruits and their freshness. In this comparative approach of various architectures of classic CNN and a residual CNN (ResNet50), ResNet50 outperformed AlexNet and VGG16 on the fruits dataset with a test accuracy of 99%.

Ayana et al. [23] designed a model in which three pre-trained models—EfficientNetB2, InceptionV3, and ResNet50—along with three optimizers—Adam, Adagrad, and stochastic gradient descent (SGD)—were used to execute the proposed multistage transfer learning (MSTL) technique. They utilized datasets with 20,400 images of cancer cells, 200 images of ultrasound from Mendeley, and 400 images of ultrasound from the MT-Small-Dataset. They achieved an accuracy of 99% with ResNet50 + Adagrad on the Mendeley dataset and an accuracy of 98.7% on the MT-Small-Dataset.

Taneja et al. [24] used genetic algorithms with CNN to implement a model to classify malware files into families. They achieved an accuracy of 98.11% on the MalImg dataset. Casillas and Osuna-Ensico [25] leverage the power of Genetic Algorithms at finding optimal solutions to find the optimal CNN architecture, in one of the chapters of the book they utilized micro genetic algorithms to find the CNN architectures in the fastest way. They tested the proposed model on three datasets: MNIST, MNIST-RB, and MNIST-fashion, and compared the results against two other frameworks: proCNN and simple GA. The results showed better performance by the architectures found using their framework in terms of both accuracy and processing time. The gaCNN achieved an accuracy of 94.6% on MNIST, 91% on FASHION, and 97.3% on MNIST-RB compared to psoCNN with 91.9%, 92.8% and 95.2% respectively. micro-gaCNN outperformed both models with 96.4% accuracy on MNIST, 96.6% on FASHIOn, and 97.99% on MNIST-RB.

3 Motivation/ Problem Statement

Machine learning techniques, particularly convolutional neural networks (CNNs), have demonstrated significant advantages over traditional methods in classifying medical images. Traditional methods often rely on manual feature extraction and selection, which can be time-consuming and prone to errors. In contrast, CNNs automatically learn hierarchical representations of data from raw input, eliminating the need for manual feature engineering.

Several studies have shown the superior performance of CNNs in medical image classification. For instance, a study by Liu et al. (2021) applied CNNs to classify mammograms and achieved an accuracy of 95.2%, outperforming traditional methods. Similarly, Wang et al. (2020) used CNNs for COVID-19 diagnosis from CT scans and achieved an accuracy of 90.8

In addition to higher accuracy, CNNs can also save time and cost compared to traditional methods. For example, a study by Akkus et al. (2017) demonstrated that CNNs can significantly reduce the time required for brain tumor segmentation in MRI scans, compared to manual segmentation by radiologists.

Furthermore, CNNs are more effective than conventional approaches in handling the enormous volume and complexity of medical picture data. For instance, a research by Litjens et al. (2017) examined CNN applications in medical image processing and came to the conclusion that CNNs can analyze high-dimensional data well and outperform conventional approaches in a variety of medical imaging tasks.

In conclusion, machine learning methods—particularly CNNs—offer a number of advantages over conventional approaches in the categorization of medical images, including greater accuracy, efficiency, and improved management of vast and complicated data sets. CNNs are increasingly being used in clinical practice and research for medical imaging as a result of these benefits.

This article presents a novel approach for accurately classifying images into their appropriate classes. The proposed method leverages the power of transfer learning by employing a pre-trained convolutional neural network ensemble. To identify the most effective ensemble, we utilize a genetic algorithm that searches for the optimal combination of networks. By combining these techniques, we aim to improve the accuracy of image classification.

4 Materials and Methods

The proposed method aims at searching for the best architecture of an ensemble of state-of-the-art pre-trained convolutional neural networks. the power of Transfer learning is utilized along with Genetic Algorithms (GA) as the optimization search algorithm to find the best ensemble of CNN with the best classifier.

4.1 Model Architecture

The model consists of two parts:

- The ensemble of convolutional neural networks
- The fully connected (FC) layers

The ensemble consists of 7 state-of-the-art convolutional neural network models:

- VGG16
- VGG19
- Xception
- NasNetLarge
- EfficientNet
- ResNet50
- MobileNet

The FC part consists of 3 different FC layers architectures

- Shallow network with two hidden layers:
  - layer1: 4096 nodes with ReLU activation function and dropout (0.2)
  - layer2: 1000 nodes with ReLU activation function and dropout (0.2)
  - a softmax output layer
- Medium network with three hidden layers:
  - layer1: 1024 nodes with ReLU activation function and dropout (0.2)
  - layer2: 512 nodes with ReLU activation function and dropout (0.2)
  - layer3: 256 nodes with ReLU activation function and dropout (0.2)
  - a softmax output layer
- deep Neural Network with five hidden layers:
  - layer1: 4096 nodes with ReLU activation function and dropout (0.2)
  - layer2: 4096 nodes with ReLU activation function and dropout (0.2)
  - layer3: 1000 nodes with ReLU activation function and dropout (0.2)
  - layer4: 512 nodes with ReLU activation function and dropout (0.2)
  - layer5: 256 nodes with ReLU activation function and dropout (0.2)
  - a softmax output layer

In order to find the best ensemble of CNN and the best classifier, we define a Genetic Algorithm that takes in the trained models on the training set and computes the validation accuracy on the validation set. The best solution is defined as the ensemble that gives the best validation accuracy.

The architecture of the GA goes as demonstrated in Fig1:

- Create the initial population
- Compute the fitness value of the population

- Select: chooses two chromosomes from the population
- Crossover: performs crossing over of the two selected chromosomes at a certain probability and returns two children Fig.4
- Mutation: performs mutation on the children with a certain probability Fig5
- Add to the new population

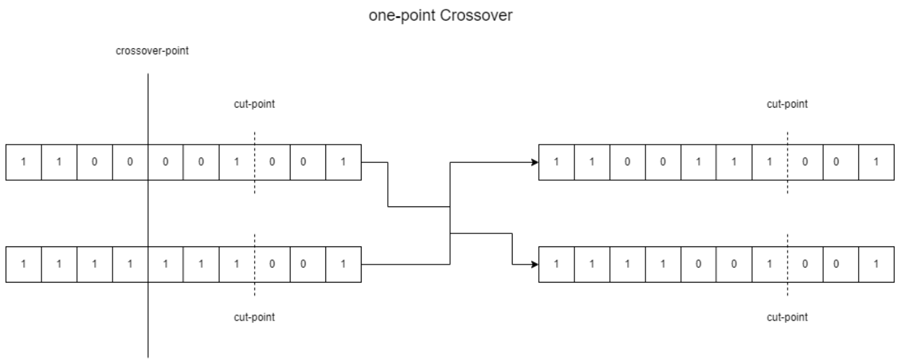

Fig. 4 One-point crossover

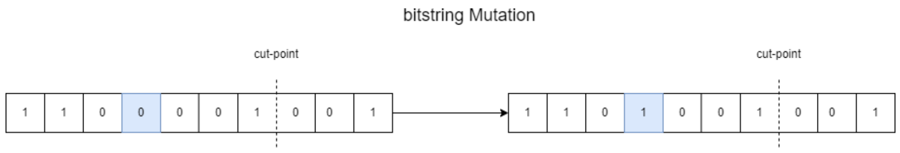

Fig. 5 bitstring mutation

Steps 2 to 5 are repeated until the size of the new population is the same as the old population.

Additional steps were performed:

Elitism: the best chromosomes of the old population are added directly to the new population.

The stopping criterion in our model is the number of iterations the GA has to perform.

4.2 Evaluation Metrics

The metrics used to evaluate the performance of our models are:

- Accuracy
- balanced accuracy
- Precision
- Recall
- F1 score

- Specificity
- G-mean

The geometric mean or the g-mean is the square root of the recall multiplied by the specificity. Fig6 shows the formulas of the evaluation metrics and Fig7 shows the structure of a Confusion Matrix for multi-class classification

$$
\begin{aligned}
precision &= \frac{TP}{TP + FP} \\
recall &= \frac{TP}{TP + FN} \\
F1 &= \frac{2 \times precision \times recall}{precision + recall} \\
accuracy &= \frac{TP + TN}{TP + FN + TN + FP} \\
specificity &= \frac{TN}{TN + FP}
\end{aligned}
$$

Fig. 6 Evaluation Metrics

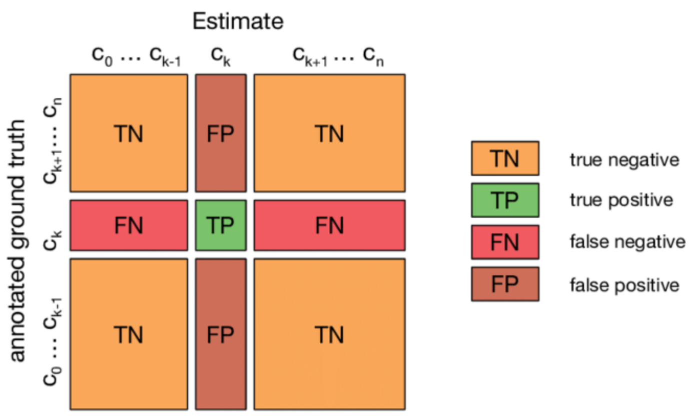

Fig. 7 Confusion Matrix for multi-class classification [26]

5 Experiments

This section discusses the experiments implemented to test our model. We ran our model on three datasets:

- MRI image dataset
- Mice Kidney Fibrosis
- Tiny-MNIST
- Tiny-CIFAR10

The reason for taking a ”tiny” version of the datasets is to accommodate the available computational power and memory of the machines.

The experiments were run on two machines with the same following specs:

- RAM: 32 GB
- Processor: Intel Core i9-9900 CPU @ 3.1GHZ, 3096 Mhz, 8 Core(s), 16 Logical Processor(s)
- GPU: NVIDIA GeForce RTX 2080

The experiments were run as follows:

- Individual models were run on the training data independently
- The validation set along with the models was fed to the GA to find the best ensemble of models with the highest validation accuracy.
- The best solution that the GA found is trained on the training set
- The solution then is used to predict (unseen) testing data
- The individual models are used to predict on unseen data
- Comparison between the individual models' prediction results and those of the ensemble chosen by the GA

Table1. shows the parameters of the experiments.

Table 1 Experiments Setup

Parameter	**Value**
Optimizer	Adam
Learning Rate	0.001
Length of the chromosome	10
Number of Generations	100
crossover probability	0.8
mutation probability	0.2
crossover-type	one-point
mutation type*	bitstring
selection type	ranking selection
population size**	7
number of elites**	1

* The mutation value increases by 0.05 if the evaluation value is constant for five consecutive iterations.

** Except for the first experiment where the GA ran with population size 6 and no elitism and a constant mutation rate of 0.2.

5.1 Experiment 1:

5.1.1 Data

The first experiment was done using an MRI image dataset for brain tumors. The purpose of the experiment is to find the best ensemble model for classifying images as no-tumor (no) or tumor (yes). All the images (training, validation, and testing) were preprocessed using the following techniques:

- Converting the images to grayscale to apply the denoising technique
- Applied BM3D denoising technique to enhance the contrast of the images.
- Converting grayscale images to 3 channels (RGB) images to be taken by the CNNs.
- resizing all images to (224, 224, 3) to be accepted by the CNN models.

BM3D denoising technique was chosen after trial and error and comparison of several denoising techniques such as the Gaussian, Bilateral, Wavelet, Isotropic diffusion, and BM3D denoising filter. The consensus was that BM3D gave way better results with less blurring of the original images and better image enhancement. The majority of images contain repetitive or similar information. The BM3D algorithm, also known as Block Matching 3D Denoising, utilizes this redundant data to perform denoising. It identifies similar blocks within the image, forms a three-dimensional array, and employs a collaborative filtering technique to convert this array into a sparse representation in the transform domain, ultimately restoring the image. [27]

The dataset was split into three samples:

Training: 151 images

Validation: 50 images

Testing: 51 images

Fig.6 illustrated an MRI image with no tumor while Fig.7 illustrates an image with a tumor.

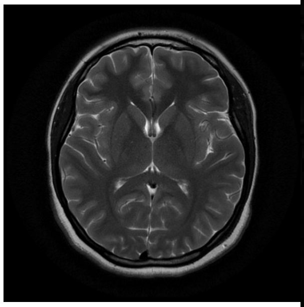

Fig. 8 (no) No tumor

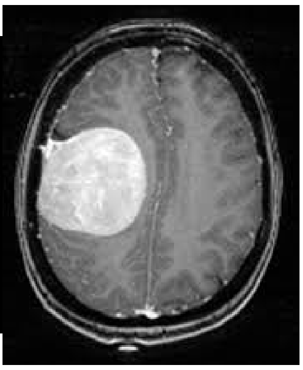

Fig. 9 (yes) tumor present

Note that data augmentation was not applied in this experiment.

5.1.2 Results

Each model took 5min to train on average and the chromosome took 3min to evaluate on average. The fitness evolution is demonstrated in Fig8. We justify the graph structure by the absence of elitism. Hence, the best chromosome may not be present in the new population. But in the end, the best solution was found with 88% validation accuracy at iterations 97 and 98 but it got replaced in 99 and 100 by the new population since elitism is not performed.

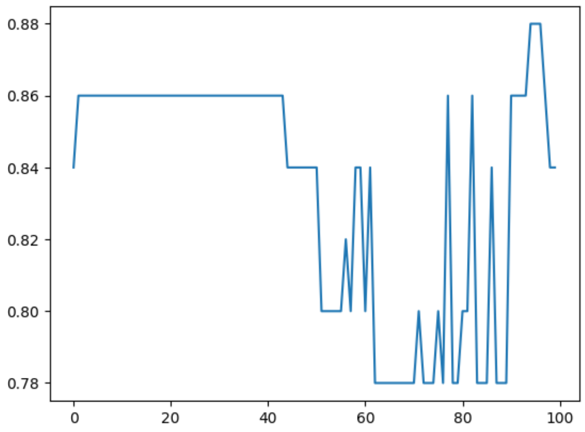

Fig. 10 Fitness evolution of the MRI Images experiment

* VGG19, Xception, MobileNet/ DNN Table2 shows the testing results of individual models on the test data on individual models and the ensemble. Our model composed of VGG19, Xception, MobileNet with DNN had an accuracy of 92.15%, we notice that MobileNet alone with a medium fully connected layers gave better results with 97.6% the same performance accuracy (and similar results in other metrics).

5.2 Experiment 2:

5.2.1 Data

The second experiment was done using ultrasound images of mice kidneys to detect the presence and stage of kidney fibrosis. The purpose of the experiment is to find the best ensemble model for classifying images as 0 (sham), 1 (mild), and 2 (severe) as shown in Figures 9,10, and 11. As we can see the images are very blurry and it is hard to distinguish between the stages of the fibrosis with the naked eye. We hope that our model can find useful information in the images to be able to distinguish between the three stages of Kidney Fibrosis. No method of image pre-processing nor augmentation was performed. The dataset was split into three samples:

Training: 209 images
Validation: 99 images
Testing: 33 images

Metrics	Accuracy	Balanced Accuracy	Precision	Recall	F1 score	Specificity	G-mean
VGG16/shallow	92.2	91.4	91.4	96.5	62.6	92.6	92.6
VGG19/shallow	88.2	88.3	88.4	86.7	87.55	86.77	86.77
Xception/shallow	92.15	91.45	91	92.66	91.5	92.66	92.66
ResNet50/shallow	72.5	71.2	71	71.2	70	71.2	71.2
MobileNet/shallow	92.1	92.6	92.2	90	91	90	90
EfficientNet/shallow	60	60	30	50	30	50	50
NasNetLarge/shallow	90	89.5	89.5	90.1	89.75	90	90
VGG16/medium	86.2	85.8	85.85	85.1	85	85	85
VGG19/medium	90.1	89.5	90	90.1	90	90.2	90.2
Xception/medium	86.3	86.07	87.8	87.9	87.8	87.8	87.8
ResNet50/medium	62.7	61.5	54.3	54.3	54.3	54.2	54.2
MobileNet/medium	90.2	93.05	87.5	87.5	87.5	87.5	87.5
EfficientNet/medium	60.7	60.7	50	50	50	50	50
NasNetLarge/medium	86.27	85.55	86.9	86.9	86.9	86.9	86.9
VGG16/DNN	92.15	91.77	91.77	91.77	91.77	91.77	91.77
VGG19/DNN	88.23	87.73	89.43	89.43	89.43	89.43	89.43
Xception/DNN	86.27	86.07	87.82	87.82	87.82	87.82	87.82
ResNet50/DNN	60.78	60.78	50	50	50	50	50
MobileNet/DNN	**97.6**	**97.6**	**97.6**	**97.6**	**97.6**	**99.7**	**96.7**
EfficientNet/DNN	60.78	60.78	50	50	50	50	50
NasNetLarge/DNN	84.31	83.54	83.54	83.54	83.54	83.54	83.54
ensemble*	92.15	91.77	93.54	90	93.54	93.5	91.75

Table 2 Test Results on the MRI image dataset.

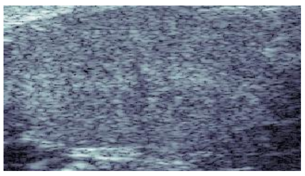

Fig. 11 (0) sham

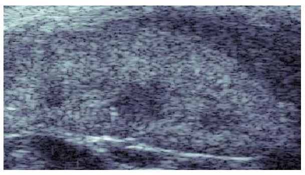

Fig. 12 (1) Mild Fibrosis

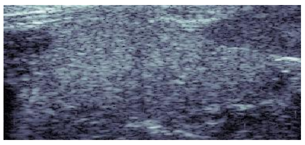

Fig. 13 (2) Severe Fibrosis

5.2.2 Results

The models took around 7min to train on average and each chromosome took 5min on average to evaluate. The fitness evolution is demonstrated in Fig12.

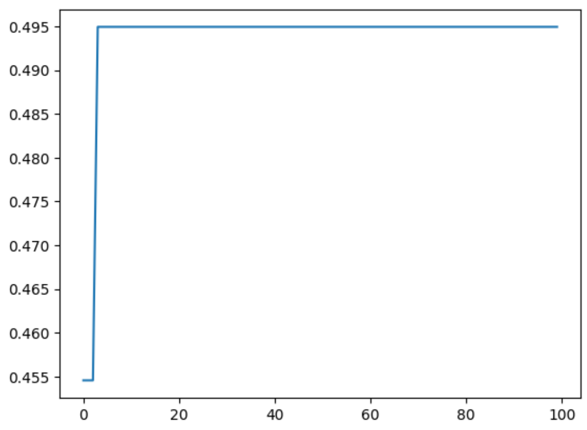

Fig. 14 Fitness evolution of the mice kidney fibrosis experiment

As we can notice the fitness value started at 45.5% validation accuracy and went up to 49.5% in the first iterations and stayed constant throughout the entire process of the GA.

Table 3-7 shows the test results of the individual models and the best ensemble found by the GA (VGG16, ResNet50/CNN) on the unseen data. We notice that the individual models VGG16/medium and MobileNet/DNN performed the best with an accuracy of 57.5% and 54.5% respectively on the unseen data.

5.3 Experiment 3:

5.3.1 Data

The third experiment was done using a small part of the MNIST dataset. The purpose of the experiment is to classify the images into one of the 10 classes that represent handwritten digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9.

A sample of 10,000 samples was taken at random for training.

A sample of 5000 samples was taken at random for validation.

A sample of 5000 samples was taken at random for testing. The data were converted from grayscale to 3 channels (R,G,B) and resized to (224, 224, 3) to be taken by the CNN models.

5.3.2 Results

The GA was unfortunately run for only 10 generations due to time restrictions, the last 3 iterations all gave the same ensemble which consist of VGG16, EfficientNet,

Metrics	Accuracy	Balanced Accuracy	G-mean
VGG16/shallow	57.5	45.3	67.3
VGG19/shallow	39.3	67.7	52.4
Xception/shallow	60	63.7	70
ResNet50/shallow	33.3	33.3	47.1
MobileNet/shallow	36.3	42.7	49.8
EfficientNet/shallow	33.3	33.3	47.1
NasNetLarge/shallow	18.2	26.3	32.7
VGG16/medium	57.5	45.3	67.4
VGG19/medium	36.3	67.2	49.7
Xception/medium	39.4	62.2	52
ResNet50/medium	33.3	33.3	47.1
MobileNet/medium	**63.6**	**67.5**	**72.2**
EfficientNet/medium	33.3	33.3	47.1
NasNetLarge/medium	42.4	47.2	55
VGG16/DNN	27.2	14.5	41.6
VGG19/DNN	33.3	33.3	47.1
Xception/DNN	51.5	68	62.5
ResNet50/DNN	33.3	33.3	47.1
MobileNet/DNN	54.5	68	64.9
EfficientNet/DNN	33.3	33.3	47.1
NasNetLarge/DNN	33.3	30.5	47.1
ensemble*	33.3	33.3	47.1

Table 3 Test Results on the Mice Kidney Fibrosis image dataset. *ensemble: VGG16,ResNet50/DNN

Metrics	Precision class 0	Precision class 1	Precision class 2
VGG16/shallow	90.9	0	45
VGG19/shallow	100	NaN	35.4
Xception/shallow	50	83.3	57.9
ResNet50/shallow	NaN	Nan	33.3
MobileNet/shallow	36.3	66.6	25
EfficientNet/shallow	33.3	NaN	NaN
NasNetLarge/shallow	10	50	19
VGG16/medium	90.9	0	45
VGG19/medium	100	nan	34.3
Xception/medium	100	50	36.6
ResNet50/medium	nan	nan	33.3
MobileNet/medium	77.7	75	50
EfficientNet/medium	nan	nan	33.3
NasNetLarge/medium	30	66.6	45
VGG16/DNN	nan	0	29
VGG19/DNN	nan	nan	33.3
Xception/DNN	60	100	44
ResNet50/DNN	nan	nan	33.3
MobileNet/DNN	83.3	80	40.9
EfficientNet/DNN	nan	nan	33.3
NasNetLarge/DNN	20	46.6	25
ensemble*	nan	0	31.25

Table 4 Precision test Results on the Mice Kidney Fibrosis image dataset. *ensemble: VGG16,ResNet50/DNN

Metrics	sensitivity class 0	Sensitivity class 1	Sensitivity class 2
VGG16/shallow	90.9	0	81.8
VGG19/shallow	18.2	0	100
Xception/shallow	36.36	45.5	100
ResNet50/shallow	0	0	100
MobileNet/shallow	72.7	18.2	18.2
EfficientNet/shallow	100	0	0
NasNetLarge/shallow	9	9	36.36
VGG16/medium	90.9	0	81
VGG19/medium	9	0	100
Xception/medium	9	9	100
ResNet50/medium	0	0	100
MobileNet/medium	63.3	54.5	72.2
EfficientNet/medium	0	0	100
NasNetLarge/medium	27.2	18.2	81.8
VGG16/DNN	0	0	81.8
VGG19/DNN	0	0	100
Xception/DNN	27.3	27.3	100
ResNet50/DNN	0	0	100
MobileNet/DNN	45.5	36.36	81.8
EfficientNet/DNN	0	0	100
NasNetLarge/DNN	18.2	63.6	18.2
ensemble*	0	0	90

Table 5 Sensiticity test Results on the Mice Kidney Fibrosis image dataset. *ensemble: VGG16,ResNet50/DNN

Metrics	Specificity class 0	Specificity class 1	Specificity class 2
VGG16/shallow	95.5	90.9	50
VGG19/shallow	100	100	9.1
Xception/shallow	81.8	65.5	36.63
ResNet50/shallow	100	100	0
MobileNet/shallow	36.3	95.4	72.7
EfficientNet/shallow	0	100	100
NasNetLarge/shallow	59	95.4	22.7
VGG16/medium	95.5	90.9	50
VGG19/medium	100	100	4.5
Xception/medium	100	95.5	13.6
ResNet50/medium	100	100	0
MobileNet/medium	90.9	90.9	63.63
EfficientNet/medium	100	100	0
NasNetLarge/medium	68.2	65.5	50
VGG16/DNN	100	90.9	0
VGG19/DNN	100	100	0
Xception/DNN	90.9	100	36.3
ResNet50/DNN	100	100	0
MobileNet/DNN	95.5	95.5	40.9
EfficientNet/DNN	100	100	0
NasNetLarge/DNN	63.3	63.3	72.7
ensemble*	100	95.4	0

Table 6 Specificity test Results on the Mice Kidney Fibrosis image dataset. *ensemble: VGG16,ResNet50/DNN

Metrics	f1 class 0	f1 class 1	f1 class 2
VGG16/shallow	70.5	0	50
VGG19/shallow	56.5	NaN	37.3
Xception/shallow	857.8	70	59.2
ResNet50/shallow	nan	nan	33.3
MobileNet/shallow	36.3	47	29.2
EfficientNet/shallow	33.3	nan	nan
NasNetLarge/shallow	12.9	26.6	18.6
VGG16/medium	70	0	50
VGG19/medium	53.3	nan	35.3
Xception/medium	56.5	44	37.9
ResNet50/medium	nan	nan	33.3
MobileNet/medium	70	68.8	56
EfficientNet/medium	nan	nan	33.3
NasNetLarge/medium	35.1	51.8	43.6
VGG16/DNN	nan	0	28
VGG19/DNN	nan	nan	33.3
Xception/DNN	55.4	68	47.4
ResNet50/DNN	nan	nan	33.3
MobileNet/DNN	65.9	64.8	46.7
EfficientNet/DNN	nan	nan	33.3
NasNetLarge/DNN	25	38.8	28.5
ensemble*	nan	0	30.7

Table 7 f1-score test Results on the Mice Kidney Fibrosis image dataset. *ensemble: VGG16,ResNet50/DNN

ResNet50, and NasNetLarge with medium size fully connected layers. The models took about 1min to load and an average of 5min to evaluate each chromosome.

* VGG16,EfficientNet,ResNet50,NasNetLarge/medium

Table 8 displays the testing results of the individual models and the ensemble model over the test data. We notice that our ensemble model performed very well with a 98.44% accuracy on the unseen data. However, ResNet50 with DNN fully connected layers performed the best with an accuracy of 98.8%.

5.4 Experiment 4:

5.4.1 Data

The final experiment was executed using a small part of the CIFAR10 dataset. The purpose of the experiment was to find the best ensemble models for classifying the ten classes:

- airplanes
- cars
- birds
- cats
- deer
- dogs
- frogs
- horses
- ships

Metrics	Accuracy	Balanced Accuracy	Precision	Recall	F1 score	Specificity	G-mean
VGG16/shallow	96.82	96.82	96.8	96.8	96.7	99.6	98.2
VGG19/shallow	97.74	97.81	97.6	97.6	97.5	99.7	98.6
Xception/shallow	95.94	96.06	95.6	95.8	95	99.54	97.66
ResNet50/shallow	98.16	98.12	98.1	98.14	98.11	98	99
MobileNet/shallow	97.5	97.5	97.2	97.4	97.3	99	98.6
EfficientNet/shallow	98.32	98.34	98.26	98.2	98.2	99.8	99
NasNetLarge/shallow	95.64	95.68	95.6	95.6	95.6	99.5	97.5
VGG16/medium	98.72	98.72	**98.7**	**98.7**	98.7	99.8	99.2
VGG19/medium	98.3	98.3	98.3	98.3	98.3	99.8	99
Xception/medium	95.6	95.7	95.6	95.6	95.6	99.5	97.5
ResNet50/medium	98.6	98.6	98.5	98.5	98.5	99.8	99.2
MobileNet/medium	97.5	97.5	97.4	97.4	97.4	99.7	98.6
EfficientNet/medium	98.1	98.1	98	98	98	99.7	98.9
NasNetLarge/medium	96.6	96.6	96.5	96.5	96.5	99.6	98.0
VGG16/DNN	98	98.1	97.9	97.9	97.9	99.7	98.8
VGG19/DNN	98	98.1	98	98	98	99.7	98.9
Xception/DNN	95.4	95.4	95.3	95.3	95.3	99.4	97.4
ResNet50/DNN	**98.8**	**98.8**	**98.7**	**98.7**	98.7	99.8	**99.3**
MobileNet/DNN	97.6	97.6	97.6	97.6	97.6	99.7	96.7
EfficientNet/DNN	98.1	99.1	98	98	98	99.7	98.9
NasNetLarge/DNN	96.1	96.1	96	96	96	99.5	97.8
ensemble*	98.44	98.45	98.4	98.4	98.4	99.8	99.1

Table 8 Test Results on the MNIST image dataset. * VGG16,EfficientNet,ResNet50,NasNetLarge/medium

- trucks

A sample of 10,000 samples was taken at random for training.

A sample of 5000 samples was taken at random for validation.

A sample of 5000 samples was taken at random for testing.

The images were resized to (224, 224, 3) before they were fed to the models, no other method of pre-processing methods were performed.

5.4.2 Results

The GA was unfortunately run for only 10 generations due to time restrictions, the last 3 iterations all gave the same ensemble which consist of EfficientNet and ResNet50 with DNN fully connected layers. The models took about 1min to load and an average of 5min to evaluate each chromosome.

Table 9 shows the test results of the individual models and the ensemble on the test data. We notice from the table that EfficientNet alone outperformed the ensemble with an accuracy of 92.5%. Nevertheless, the ensemble outperformed all other models with an accuracy of 88.9%.

6 Discussion

From the previous results, we can conclude that the ensemble model chosen by the GA may outperform most of the individual models, but it might still be outperformed by individual models. In addition, we note that in the case where an individual model outperforms the ensemble if the individual model is not in the initial generation, it

Metrics	Accuracy	Balanced Accuracy	Precision	Recall	F1 score	Specificity	G-mean
VGG16/shallow	80.1	80	79.7	79.9	79.2	97.7	88.4
VGG19/shallow	81.1	80	80.3	81	80.4	97.9	89.1
Xception/shallow	10	4	0.9	9	0.9	89	29
ResNet50/shallow	82.1	82	81.7	82	82	98	89
MobileNet/shallow	54.5	54	50	55	94.9	54	71.7
EfficientNet/shallow	91.6	91.5	92	91.5	92	99	95
NasNetLrge/shallow	33.1	34	30	33.4	33	92.5	55.6
VGG16/medium	81.7	81.5	80.9	81.6	82	97.9	89.4
VGG19/medium	81.1	80.2	80.9	80.7	82	97.9	89
Xception/medium	10	4	0.9	9	0.9	89	29
ResNet50/medium	83.9	84.2	86.9	83.7	86.4	98	90.7
MobileNet/medium	51.7	82.6	51	51.7	53.2	94.6	69.9
EfficientNet/medium	92.4	92.3	96.1	92.4	92.3	99.1	95.7
NasNetLrge/medium	32.4	30.1	33.3	33.3	34.2	92.4	55.5
VGG16/DNN	79.4	80.3	76.3	79.2	79.1	97.7	88.0
VGG19/DNN	80.5	80.4	80.9	79.9	82	97.8	88.4
Xception/DNN	24.7	33.1	23	25.3	16.6	91.7	48.1
ResNet50/DNN	84.8	84.6	85	84.6	84	98.3	91.2
MobileNet/DNN	52.4	52.8	45.8	52.1	50.8	94.7	70.2
EfficientNet/DNN	**92.5**	**92.6**	**92.2**	**92.4**	**91.9**	**99.1**	**95**
NasNetLarge/DNN	42.5	43.4	42.3	42.1	44	93	62.7
ensemble*	88.9	89.2	88.9	88.5	85	98.7	96.5

Table 9 Test Results on the CIFAR10 image dataset. * ResNet50, Efficientnet/ DNN

is less likely that it will be discovered by the GA due to crossover and mutation. One way to overcome this is to start with an initial generation that contains all the individual models in addition to other chromosomes then perform crossover and mutation from there. In addition, one setback of the experiments is that they did not run for a good number of iterations and hence might not have explored the entire search space. In addition, to explore the search space more efficiently, it is suggested to try different GA parameters such as selection method, crossover type, and mutation type to create variability in the generations. Moreover, if computational power is available, additional state-of-the-art neural networks should be used such as AlexNet, DenseNet, EfficientResNet, and others. We could also introduce other types of classifiers such as K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Decision trees, and others in addition to Fully Connected Layers.

We can also conclude that there is not one ensemble that is the best at predicting any type of images, each dataset may have a different model that is best at predicting the outcome.

References

[1] Maeda-Gutiérrez, V., Galván-Tejada, C.E., Zanella-Calzada, L.A., Celaya-Padilla, J.M., Galván-Tejada, J.I., Gamboa-Rosales, H., Luna-García, H., Magallanes-Quintanar, R., Guerrero Méndez, C.A., Olvera-Olvera, C.A.: Comparison of convolutional neural network architectures for classification of tomato plant diseases. Applied Sciences **10**(4) (2020) https://doi.org/10.3390/app10041245

[2] Kimura, N., Yoshinaga, I., Sekijima, K., Azechi, I., Baba, D.: Convolutional neural network coupled with a transfer-learning approach for time-series flood predictions. Water **12**(1) (2020)

[3] Albadr, M.A., Tiun, S., Ayob, M., AL-Dhief, F.: Genetic algorithm based on natural selection theory for optimization problems. Symmetry **12**(11) (2020) https://doi.org/10.3390/sym12111758

[4] Alex Krizhevsky, G.E.H. Ilya Sutskever: Imagenet classification with deep convolutional neural networks. (2012)

[5] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Li, F.-F.: Imagenet: a large-scale hierarchical image database. IEEE Conference on Computer Vision and Pattern Recognition, 248–255 (2009) https://doi.org/10.1109/CVPR.2009.5206848

[6] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)

[7] Kaiming He, S.R.J.S. Xiangyu Zhang: Deep residual learning for image recognition. Microsoft Research (2015) https://doi.org/10.48550/arXiv.1512.03385

[8] Gao Huang, L.v.d.M.K.Q.W. Zhuang Liu: Densely connected convolutional networks (2017) https://doi.org/10.48550/arXiv.1608.06993

[9] Jason Yosinski, Y.B.H.L. Jeff Clune: How transferable are features in deep neural networks? Advances in Neural Information Processing Systems **27**, 3320–3328 (2014) https://doi.org/10.48550/arXiv.1411.1792

[10] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition, pp. 647–655 (2013)

[11] Lu, Y.: Food image recognition by using convolutional neural networks (cnns) (2016)

[12] Wang, C.Y. SH.: Fruit category classification via an eight-layer convolutional neural network with parametric rectified linear unit and dropout technique. Multimed Tools Appl **79**, 15117–15133 (2020) https://doi.org/10.1007/s11042-018-6661-6

[13] Jia, W., Snetkov, L., Aok, S.: An effective model based on haar wavelet entropy and genetic algorithm for fruit identification, vol. 1955, p. 040013 (2018). https://doi.org/10.1063/1.5033677

[14] Yadav, J.S.M. S.S.: Deep convolutional neural network based medical image classification for disease diagnosis. J Big Data **6**, 113 (2019) https://doi.org/10.1186/s40537-019-0276-2

[15] Stanley, M.R. K. O.: Evolving neural networks through augmenting topologies. Evolutionary computation **10 (2)**, 99–127 (2002) https://doi.org/10.1162/106365602320169811

[16] Ayan, E.: Genetic algorithm-based hyperparameter optimization for convolutional neural networks in the classification of crop pests. Arab J Sci Eng **10 (2)**, 99–127 (2023) https://doi.org/10.1007/s13369-023-07916-4

[17] Paymode, A.S., Malode, V.B.: Transfer learning for multi-crop leaf disease image classification using convolutional neural network vgg. Artificial Intelligence in Agriculture **6**, 23–33 (2022) https://doi.org/10.1016/j.aiia.2021.12.002

[18] Chakraborty, P.S..H.K.M.A. S.: A transfer learning-based approach with deep cnn for covid-19- and pneumonia-affected chest x-ray image classification. SN COMPUT. SCI. **3**, 14 (2022) https://doi.org/10.1007/s42979-021-00881-5

[19] Ananda Kumar, K.S., Prasad, A.Y., Metan, J.: A hybrid deep cnn-cov-19-res-net transfer learning architype for an enhanced brain tumor detection and classification scheme in medical image processing. Biomedical Signal Processing and Control **76**, 103631 (2022) https://doi.org/10.1016/j.bspc.2022.103631

[20] Showkat, S., Qureshi, S.: Efficacy of transfer learning-based resnet models in chest x-ray image classification for detecting covid-19 pneumonia. Chemometrics and Intelligent Laboratory Systems **224**, 104534 (2022) https://doi.org/10.1016/j.chemolab.2022.104534

[21] Kumar, S., Janet, B.: Dtmic: Deep transfer learning for malware image classification. Journal of Information Security and Applications **64**, 103063 (2022) https://doi.org/10.1016/j.jisa.2021.103063

[22] Kazi, P.S.P. A.: Determining the freshness of fruits in the food industry by image classification using transfer learning. Multimed Tools Appl **81**, 7611–7624 (2022) https://doi.org/10.1007/s11042-022-12150-5

[23] Ayana, P.J.J.J.-W..C.S. G.: A novel multistage transfer learning for ultrasound breast cancer image classification. Diagnostics **12 (1)**, 135 (2022) https://doi.org/10.3390/diagnostics12010135

[24] Taneja, P.S., Gopal, S., Yadav, P., Gupta, R.: Malware family categorization using genetic algorithm-cnn-based image classification technique. In: Choudrie, J., Mahalle, P., Perumal, T., Joshi, A. (eds.) ICT with Intelligent Applications, pp. 199–209. Springer, Singapore (2023). https://doi.org/10.1007/978-981-19-3571-8_21

[25] Marquez Casillas, O.-E.V. E.S.: Architecture optimization of convolutional neural networks by micro genetic algorithms. 10.1007/978-3-030-70542-8$_7$**967**, 135(2021)

Krüger, F.: Activity, context, and plan recognition with computational causal behaviour models (2016)

Li, Y., Zhang, J., Wang, M.: Improved bm3d denoising method. IET Image Processing 11(12), 1197–1204 (2017) 10.1049/iet-ipr.2016.1110 https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-ipr.2016.1110