Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 5, 2021, Pages. 1403 - 1419
Received 15 April 2021; Accepted 05 May 2021.

Detection and Diagnosis of Pancreatic Tumor Using Convolutional Neural Network and Hybrid Particle Swarm Optimization Algorithm (CNN-HPSO) For Image Classification

S. Arulmozhi[1], Dr.R. Shankar[2], Dr.S. Duraisamy[3]

[1]*Research Scholar, PG & Research Department of Computer Science, Chikkanna Government Arts College, Tirupur-641 602.*

[1]*Assistant Professor, Department of Information Technology, Hindusthan College of Arts and Science, Coimbatore-641028*

[2&3]*Assistant Professors, PG & Research Department of Computer Science, Chikkanna Government Arts College, Tirupur-641 602.*

Abstract:

Deep Neural Networks (DNNs) offer better performance in contrast to conventional Machine Learning (ML) techniquesin dealing with real-world applications.However, operational DNNs are based on the knowledge gained. More amounts of time and computational resources are consumed in this technique. In this paper, a method based on Particle Swarm Optimization (PSO) is propounded. This method is proficient in offering speedy convergence when compared to deep Convolutional Neural Networks (CNNs) models for classifying images. A unique encoding strategy along with a velocity operator is developed by incorporating the concepts of PSO with CNN. Experimental results have proved that PSO-CNN outperforms other existing methodologiesbased on Accuracy, Precision, Recall and F-measure.

Keywords: Particle Swarm Optimization (PSO), Convolutional Neural Networks (CNN), Image classification, Pooling layer

1Introduction

Currently, Deep Neural Networks (DNNs) are considered to be morepopular as they arewidelyused in the domain of computer vision. Precisely, Deep Neural Networks (DNNs)[23]offerfinest outcomes in image classification, thusoutperforming the classification proficiencies of manual experts.

The fundamental goal of this experiment is to use CNN [8] architecture inimage classification with appropriateorganizationbetween the searching speed and classification precision. In this way, anintegrated Particle Swarm Optimization (PSO)[2][7][19]implementation with Convolutional Neural Networks (CNN) is presented to address such an issue called as PSO-CNN.

2Related Work

A Neural Network (NN)[3][4]demonstrates that more profound networks with convolutional filters might accomplish preferable outcomes. Also,they demonstrate the idea of network-in-network, in which every single layer of the NN is an additional network that might be utilized to construct networks, while being computationally effective. ResNets use easy route associations, which are denoted as bounce or enduring connections to interface the inputs of layers to their associated output. Rather than finding aninput-output mapping, they perform residual mapping that makes training much simpler. DenseNets associate the output of a layer to the ensuing layer, thus approximating the connection of fully-connected NNs.

The connectivity in ResNet and DenseNet dodges the issue of fading gradient permitting the deepernetworks[10]to include thousands of layers. Nevertheless, numerous enhancements in CNN structures are available in the literature. Hence, better CNNs are reliant on the heuristic cycle that demands experiences gained from the particular problem domain, andno concrete means for planning them is conceived.

The arrangement of individuals' brains was developed billions and millions of years ago through the strategy of natural selection. Artificial Neural Networks (ANNs)[4][5] based on the fact is known as Neuro-Evolution. This methodology permits advancingeither ANNs' designs or loads. In its initial originin 1990s, Neuro-Evolution was utilized explicitly to assessthe weights in a stable ANN architecture by evading various weaknesses created by backpropagation. In those days, specialists didn't focus on enormous amount of information, thus assisting ANNs with backpropagation.

Making use of evolutionary[7] algorithms in training ANNs hasone significant disadvantage. It consumes more time to locate a better collection of weights when compared to backpropagation [11]. Hence, not long after the presentation of developmental calculations for preparing fixed ANNs, these calculations can be used to advance loads and structures simultaneously. Such calculations are termed as Topology and Weight Evolving Artificial Neural Networks (TWEANNs). Neuro-Evolution of Augmenting Topologies (NEAT) [22] is popular TWEANN. NEAT beginning with an essential one-layered neural network is developed into unpredictable organizations. NEAT is likewise fit for making repetitive neural organizations and dodges untimely combination by having a different number of NNs.

The Evolutionary Acquisition of Neural Topologies (EANT) made asan alternativeillustration of TWEANN. Like NEAT, the concept of EANT alsocommences its operation with a straightforward ANN architecture that developsextra intricate inevery cycle of the algorithm.It comes with two internal loops: one focuses on the structural exploration that experimentsnovel models through transformation, and structural exploitation that deals with upgrading loads by utilizing advancement techniques. Because of the dimensional nature of NNs, Neuro-Evolution, in any case, has simply been implementedfor shallow networks used usually in reinforcement learning.

Encoding of ANN by NEAT are inappropriate for networks as computation rapidly turns unmanageable. Hypercube-Based Neuro-Evolution of Augmenting Topologies (HyperNEAT) that utilizes connective Compositional Pattern Producing Networks (connective CPPNs) along with NEAT is adopted to address this issue.

In any case, NNs created by HyperNEATare not capable of coordinating best CNN architectures. Therefore, it is regularly viewed to be healthier to utilize HyperNEAT as a term extractor to further AI algorithms [15]. In 2017, analysts from Google built up the Large-Scale Evolution of Images Classifiers (LSEIC)[9] calculation that had the option to beat the impediments of conventionalNeuroEvolution techniques, and the cutting edge brings about numerous benchmark datasets utilized by DNNs [16].

To advance DNNs offering better class outcomes [17], Auto-Keras[24]is additionally equipped for looking for CNN models with improved class results.

Lately, built up the Evolving Deep Convolutional Neural Network (evoCNN), and Evolving Unsupervised Deep Neural Network (EUDNN)[12]is considered to be more efficient to address the constraints of HyperNEAT. The concept of GA is adopted to scrutinize the CNN architectures with clear crossover and mutation operators. In EUDNN, a collection of vectors is used to encryptweights and associations competently. Nonetheless, all algorithms needmore computational resources that numerous analysts of different fields might not focus on.

PSO is an added nature-propelled algorithm which can be used in searching for optimal NNs model[18].Essentially to GAs, PSO can be used to improve loads and models of NNs. Awork using PSO for ANN wasdesigned. ANNs produce betterresults with PSO in contrast to conventional backpropagation. One is for searching and finding improved architectures andanother to train ANNs.They indicated that PSOmight likewise be utilized to develop ANN architectures delivering serious outcomes in contrast to different techniques. Nonetheless, the PSO algorithms were improvisedwithoptimal architectures consisting of whollyrelatedNNsthatdo not fit forimage classification[14].

In order to beat this impediment, built up a PSO-based scheme to consequently build Convolutional Auto Encoders (CAEs) and obtained cutting edge execution in multiple image classification datasets. It is important to build analgorithm to straightforwardly develop CNN architecturesbased on PSO. In their analysis, particle encoding is propelled by computer networks, in which each layer is assigned an IP addressbased on standard PSO.

3 Preprocessing

Noise is an irregular variation of image intensity and noticeable as pieces of grains in the image. Noise may occur during image capturing or transmission. If the pixels in the picture show distinctive intensity values rather than genuine pixel values that are obtained from an image, then it is termed as noise. Noise removal

Received 15 April 2021; Accepted 05 May 2021.

algorithm is the way toward eliminating or decreasing noise from an image. The noise removal algorithms lessen or eliminate the noise by smoothing the whole image leaving territories close to differentiate limits.

Digital images are regularly tainted by impulse noise because of errors produced in noisy sensor, errors that happen during the procedure of changing signs from analog-to-digital and also errors that are created in the communication channels. These errors may definitely adjust a portion of pixels while some of the pixels stay unaltered.

To eradicate impulse noise andupgrade image quality, a median filter is used and a technique based on improved median filtering algorithm is developed. This technique eliminates or successfully stifles impulse noise in the image whiles saving the boundary data and upgrading the picture quality.

The median strategy is a spatial area methodthat uses the covering window to screen the signal based on the selection of an actual median per window. The methodology picked in this work depends at the functional level window that helps in easy selection of ordinary median as the quantity of elements in the window is odd.

Inverse filtering is highly influencedby additive noise. The method of reducing degradationleads to designing of a restoration algorithm for different types of degradation and integrate them. Wiener filter aids in implementing an optimal trade-off between inverse filtering and noise flattening. It removes the increased noise and upturns blurring concurrently. Inverse filtering is a rebuilding strategy for deconvolution, i.e., when an image is obscured by a low pass filter, it is conceivable to recuperate the image by using inverse filtering or comprehensive inverse filtering.

To progressively decompose an image,Discrete Wavelet Transform (DWT) is utilized as it is a numerical tool for image decomposition. It is valuable for handling non- stationary signals processed by this technique. The change comprisesof little waves called wavelets that have varying frequency. Spatial data and frequencies are additionally given. Discrete wavelet is a fixed function that is used to produce wavelets subtleties and their interpretation [1].

DWT is used for image depiction in multi-resolution form. Decoding is done in succession from less to advancedresolution. It separates signals into maximum and minimum recurrence segments. Data identified with edge parts exist in high frequency segment, and the low frequency segment is chosen again for parting into maximum and minimum recurrence segments [5]. The parts having low recurrence are generally chosen for watermarking. In two dimensional deterioration, the main degree of disintegration come in four sub-groups specifically LL1, LH1, HL1 and HH1. For additional degree of decay, the LL sub-band of the DWT1 level is used for 2DWT disintegration as information which splits the LL1 band into the four sub-groups [4]. Similarly, segmentation of third level is applied on LL2 with the assistance of DWT is done and four sub-groups LL3, LH3, HL3 and HH3 are obtained. LH1, HL1 and HH1 are viewed as most noteworthy recurrence groups in image segmentation, while LL3 is considered as being the least recurrence band.

Received 15 April 2021; Accepted 05 May 2021.

3.1 Image Quality Enhancement using Median and Weiner Filters

The Wiener filtering usesMean Square Error (MSE) and is capableof binding the general MSE during filtering and commotion smoothing process.

An original image can be directly assessed using the Wiener filtering mechanism. The methodology depends on a stochastic structure. From the symmetry rule, it is inferred that the Wiener channel in Fourier space is as follows:

$$w(r_1, r_2) = \frac{H(r_1,r_2)S_x(r_1,r_2)}{|H(r_1,r_2)|^2 S_x(r_1,r_2)+H_n(r_1,r_2)}(1)$$

where,

$S_x(r_1, r_2)$ - Power spectra of the actual image

$H_n(r_1, r_2)$- Power spectra of additive noise, H(r1,r2) - Blurring filter

It is informal to observe that the filter includes two separatesections. One deals with inverse filtering, while the other focusses on noise smoothing. In addition to the process of deconvolution by inverse filtering (highpass filtering) it is capable of removing noise using compression (lowpass filtering).

3.2 Discrete Wavelet Transforms (DWT)

Discrete Wavelet Transform (DWT) is used in numericaland functional analysis, wherein wavelets are separately sampled. Temporal resolution is the primary benefit of Fourier as it considers both frequency and location.It differs from Continuous Wavelet Transform (CWT), wherein the signal is decomposed into equally orthogonal set of wavelets. Hence, it is also termed as Discrete-Time Continuous Wavelet Transform (DT-CWT).Scaling function with scaling properties is used to build this wavelet. It is essential that these functions are orthogonal to discrete translations.

$$\phi(b) = \sum_{r=-\infty}^{\infty} a_r \, \phi \, (C_b - r)(2)$$

where,

C- Scaling factor (can be 2)

The regionsamong the functions should be standardized and the scaling function should be orthogonal to the integer translates.

$$\varphi(b) = \sum_{r=-\infty}^{\infty} (-1)^r \, a_{N-1-r} \varphi(2b - r) \quad (3)$$

where,

N - Even integer

The wavelets that are used to decompose signal form an orthonormal base.

4Segmentation

Active contour is a sort of segmentation procedure. It is represented asa dynamic model for segmentation. Contours are limits intended for the Region of Interest (RoI) needed in animage. Contour is an assortment of focus that goes through interpolation measure. The insertion steps can be direct, splines, polynomial that depictsthe curvature in image [2]. Various representations of active contours are involved in segmentation ofimages. The main aim of using active contours is to characterize even shape and form a closed contour of the region. Active contour models include snake, slope vector stream snake, expand and mathematical or geodesic models.

In order to separate 3-D images from various clinical imaging modalitiescontours can be adopted. 2-D cuts of images are used for separating thecertain object from 3-D images. These 2-D slices along the divided objective locale are exposed to 3-D remaking to isolate the pixels Lattice model of the 3-D picture as planned prior to applying dynamic shape model. The lattice aids in arranging deformable forms of the objective item in the directional 2-D cuts of 3-D pictures.

5FeatureExtraction and Classification

Existing method usesConvolution Neural Network (CNN) for classification.

5.1 Convolution Neural Network (CNN)

Deep Neural Networks (DNNs)[13] can be organizedinFeed-Forward NNs (FFNN) for example, Fully-Connected NNs (FCNNs) and deep CNNs Recurrent Neural Networks (RNNs) such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs)[25]. FFNs arechieflyused for analyzinginformation in every layer, and offer acceptable classification and object identification, while RNNs have feedback loops among layers permitting them to be used in case of time-based jobs, for example, natural language understanding.

The output of a particular layer is given as input to the subsequent layer [21]. The output of every layer is a component of data sources and inner weights. CNN can be characterized as shown in Eq. (4),

where,

I - Input data

$f_i(\bullet)$ - Activation function of the i^{th} layer

w_i $(\bullet)$ - Weight activity of the i^{th} layer

z_i - Output of the weight activity of the i^{th} layer before applying the activation function,

W_i - Weight of the i^{th} layer

O_i - Output of the i^{th} layer

Received 15 April 2021; Accepted 05 May 2021.

$$\begin{cases} O_i = X & i = 1 \\ O_i = f_i(z_i) & i > 1 (4) \\ z_i = g_i(O_i - 1, w_i) & \end{cases}$$

$$\begin{cases} z_i = w_i(.)O_i - 1, \text{ If } i^{th} \text{ layer is conv} \\ z_i = \# \, r, cO_i - 1, \text{ If } i^{th} \text{ layer is pool } (5) \\ z_i = w_i \, . \, O_i - 1, \text{If } i^{th} \text{ layer is FC} \end{cases}$$

Various categories of layers areConvolution (conv), pooling (pool) and Fully Connected (FC) layers as shown in Eq. (5).

While NNs are trained, the mainaim is to decrease the error amongtraining and network expected outputs. In CNNs, it is entirely expected to limit the cross-entropy loss. Thereduction is done usinggradient descent and backpropagation. BAS even CNNs demand more number of boundaries, training is feasible onlyby utilizing Graphic Processing Units (GPUs). Basedon the CNN architecture, trainingmay involve days or weeks making it tedious to experiment CNN architectures. It is essential to create algorithmswhich are proficient inrepeatedlyproducing and assessing CNN architectures in limited time.

6Optimized PSO-CNN based Classification

A type of meta-heuristic algorithm is PSO [6] which is mostly applicable in discrete and combinatorial improvement issues [20]. In PSO, a solution is a particle, and the assortment of whole set of solutions is known as a swarm. The fundamental thought in PSO is that the particles just know about the present speed, individualfinest arrangement accomplished before (pBest), and the molecule that is the existing best in the multitude (gBest). In every cycle, the particles change their speed such that theirnovel positions will be nearer to 'gBest' and 'pBest'concurrently. The speed of every particle, v, is refreshed by the following condition:

$$a_{i,j}(m + 1) = y \times a_{i,j}(m) + d_p \times e_p \times \left(pBest_{i,j} - b_{i,j}(m)\right) + d_g \times e_g \times (gBest_j - b_{i,j}(m))$$

(6)

where,

$a_{i,j}$- Velocity of the i^{th} particle in j^{th} dimension

x - Presentspot of a particle

w - Constant

d_p, d_g- Constants

e_p and e_g- Arbitrary numbers in the range [0, 1]

Received 15 April 2021; Accepted 05 May 2021.

The constant variable is used to adjust the velocity variation in the previous and current steps. In addition to this, the algorithm is also capable for exploration and exploitation whichinvolves modification of constants.The location of the i^{th} particle in j^{th} dimension is reorganized as shown below.

$$b_{i,j}(m+1) = b_{i,j}(m) + a_{i,j}(m+1) \qquad (7)$$

7Proposed System

The proposed algorithm takes the training data and identifies boundaries using the CNN models (Figure 1). In the propounded algorithm, the 'gBest' particle is chosen depending on the finest blocks identified from the swarm using PSO computation.

In the proposed scheme, parameter optimization is undone. Biggerblocks are given to the cutting edge as the 'gBest' particle. Though particle assessment should be restarted at each step,the algorithm guarantees that better blocks are kept.

The proposed calculation includes a PSO structurethat is made of sixstrategies. It allows it to look for ideal CNN models: a productive CNN illustration, initialization of particle swarm, fitness assessment ofeveryparticle, anamountof contrastbetween particles, velocity calculation and particle update. Figure 1 shows the architecture of the proposed system.

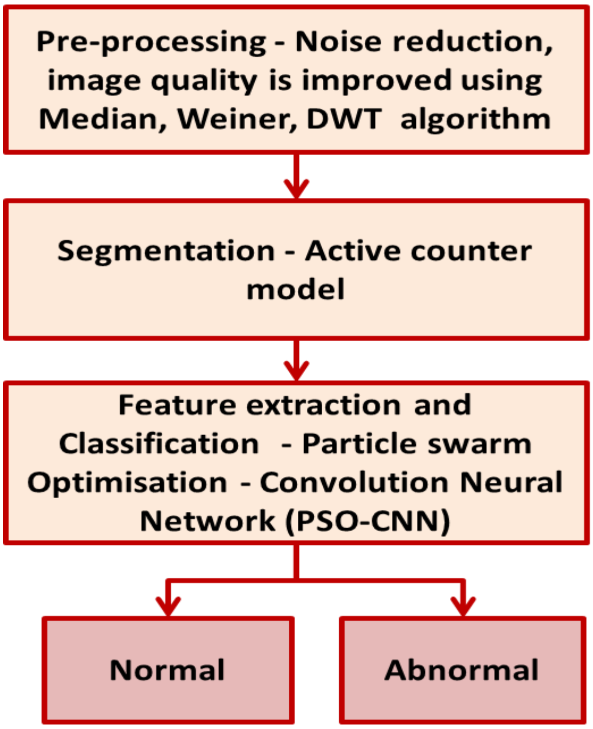

Pre-processing - Noise reduction, image quality is improved using Median, Weiner, DWT algorithm
Segmentation - Active counter model
Feature extraction and Classification - Particle swarm Optimisation - Convolution Neural Network (PSO-CNN)
Normal
Abnormal

Figure 1: Architecture

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 5, 2021, Pages. 1403 - 1419
Received 15 April 2021; Accepted 05 May 2021.

7.1 CNN Representation

An immediate encoding plan for CNN structures is proposed.CNN is utilized for preparing, testing or assessment.

The propounded algorithm just looks for CNN designs. It is important to mention about four layers: Conv, Max Pool, normal pool and FC. The layers contain information with respect to their type and hyper boundaries. For instance, in case of conv layer, the number of output features is mentioned.In case of FC layer, the amount of neurons is denoted, whereas the size of the kernelisrepresented when the layer type is convolutional or pooling.

Figure2 delineates the depictionused in PSO-CNN calculation, where C denotes Convolutional, ‘P’ stands for Pooling (max or normal) and ‘FC’ denotes FC, separately.

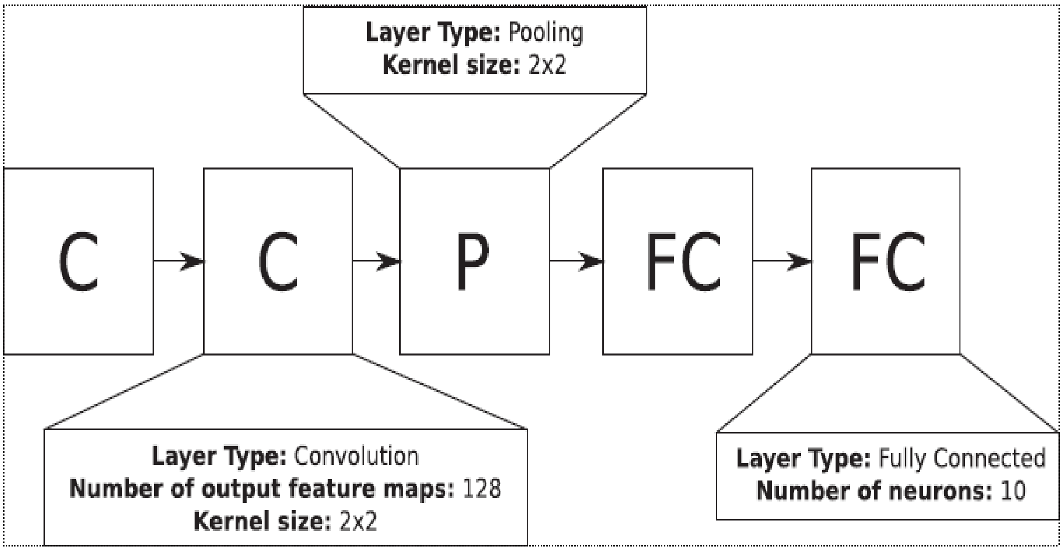


Figure 2: Single particle representation in the proposed PSO-CNN

One significant aspect of the proposed scheme is that it is not essential to update the particles of PSOover mathematical properties.

In the proposed illustration, each particle is a sequence of functional blocks that uses custom blocks. At present, the blocks in the proposed PSO-CNN may belong to any of the four layers types as mentioned earlier.

7.2 SwarmInitialization

In the proposed PSO-CNN, swarm initialization is the primary step. This can be done using a function InitializeSwarm().This functionmakes‘N’ particles with self-assertive CNN structures. The steps are detailed in the algorithm below.

Each particle has a self-assertive amount of layers, among three up to'l_{max}'. To create achievable CNN models, the first layer of particle could be a convolution layer and the last layer is a FC layer.

Moreover, FC layers cannot be put between convolution or pooling layers, whereas it can be placed only at the end of the model. During the declaration stage, it is more important to ensure that once the FC layer is included in the architecture,each succeeding layer can be possibly a FC layer.FC layers used at the last phase of a CNN are a traditional way embraced by different scientists. In order to segregate the features mined by the convolutional and pool layers, FC layers are used. By making use of FClayers in convolutional and pooling layers, the boundaries of the overall CNN can be built. The algorithm is shown below.

```
N - Size of  Swarm
$Max_{Layers}$ - Maximum  quantity of layers
$Max_{Maps}$ - Maximum  quantity of feature maps
$Max_K$ - Maximum Convolution kernel size
$Max_n$ - Maximum Amount of Neurons in FC layer
$n_{out}$ - Amountof Output
$P_i$- $i^{th}$ Particle
$d_i$- depth of $i^{th}$ Particle

Output:
Set of 'n' particles - {$P_1$, $P_2$ .... $P_n$}
for i=1 to n do
    $d_i$=rand(3,d);
    for j=1 to $d_i$ do
        if j==1 then
            layer[j]=addConv($K_{max}$, $maps_{max}$);
        elseif j=$d_i$ then
            layers[j]=addFC($n_{out}$);
        elseif layers[j-1].type== 'FC' then
            layers[j]=addFC($n_{max}$);
        else
            type=rand(1,3);
            if type==1 then
                layers[j]=addConv($K_{max}$, $maps_{max}$);
            else if type==2 then
                layers[j]=addPool();
            else
                layers[j]=addFC($n_{max}$);
            end
        end
    end
    $layers_i$= layers;
end
return S={$P_1$, $P_2$ .... $P_n$};
```

In the Algorithm, the addConv()functionhelps to include theconv layer to the architecture with arbitrary number of output feature maps from 1 up to ‘$\text{maps}_{\max}$’, its part size is picked indiscriminately from 3 × 3 up to × $k_{\max}$, where ‘$k_{\max}$’ signifies the greatest convolution portion size, and its kernel strides is 1 × 1.

The addPool() function arbitrarily adds a maxpool or average pool layer to a particle architecture. The size of the window is set to 3 × 3 in steps of 2 × 2. The actuation capacity of all layers is consistently a Rectified Linear Unit (ReLU). The quantity of pool layers is compelled by the size of the information. For a contribution of size 28 × 28, the computation can just amount to 3 pool layers.

7.3 Fitness

The ComputeLoss() is used for fitness evaluation. The particle architecture is arranged into an undeniable CNN and preparing it for an aggregate of trained epochs. The assessment is finished by associating the loss function of every particle, the crossentropy loss. Accordingly, the purpose of the technique is to discover particle architecture with the minimum loss irrespective of the quantity of parameters or other measures.

Training is done using Adam [37] and weights are adjusted with Xavier [38]. Moreover, it is additionally conceivable to add dropout and batch standardization among layers evading the overfitting issue [39]. This is the principalbottleneck of the propounded algorithmas the particles should be trained in the whole dataset.

8Results and Discussion

In this experiment, segmented CT scans of three various pancreatic tumor classes from the Memorial Sloan Kettering Cancer Center (MSKCC). The three classes are pancreatic ductal adenocarcinoma (PDAC), Intraductal Papillary Mucosal Neoplasms (IPMNs), and Pancreatic Neuroendocrine Tumors (PNETs). There are no benign scans. The datasetconsists of 103 patients with IPMN, 57 with PNET, and 260 with PDAC, for a total of 420 patients. The images are classified into normal and abnormal images.

Figure 3 shows the image taken as input. Figures4to 6show the image after applying Median, Weiner and DWT Filters respectively.

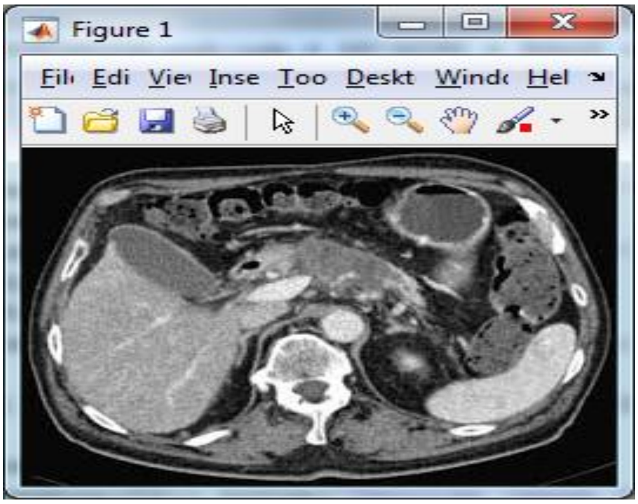


Figure 3: Input Image

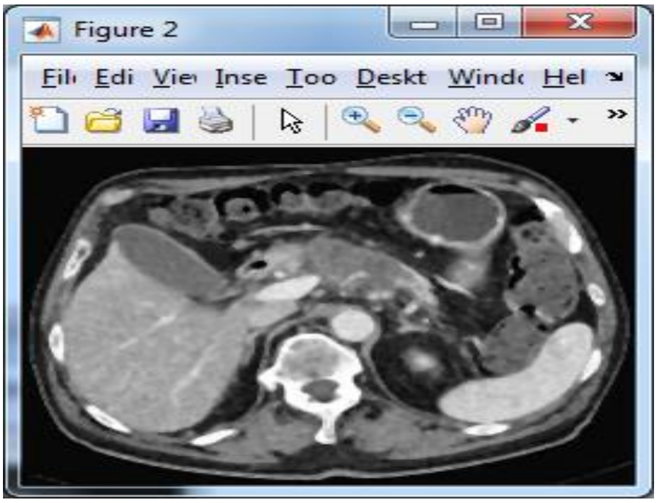


Figure 4: Image after Applying Median Filter

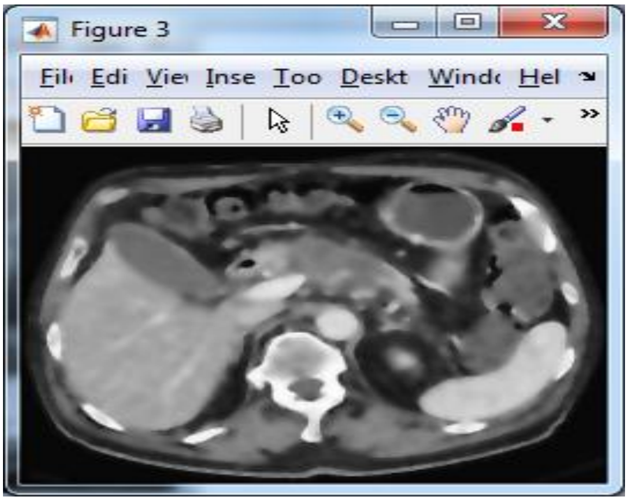


Figure 5: Image after Applying Weiner Filter

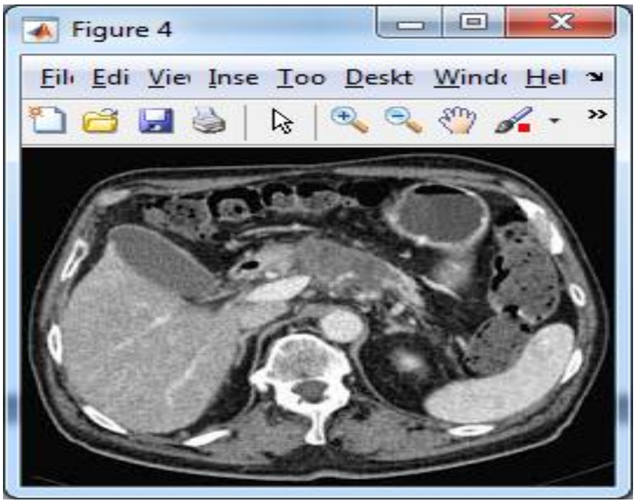


Figure 6: DWT based Denoised Image

Received 15 April 2021; Accepted 05 May 2021.

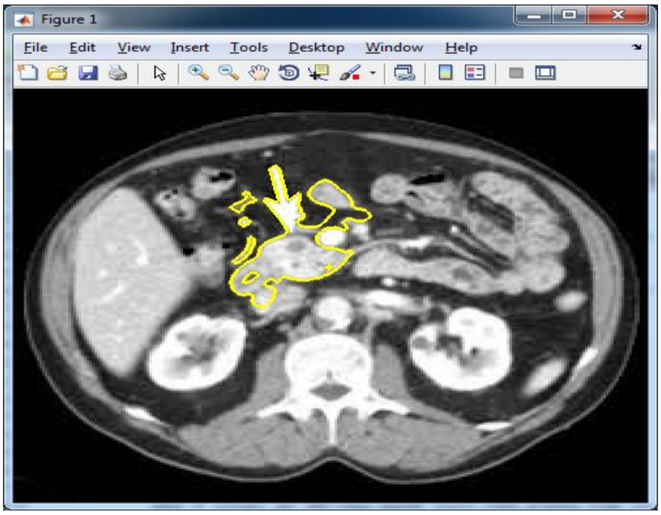
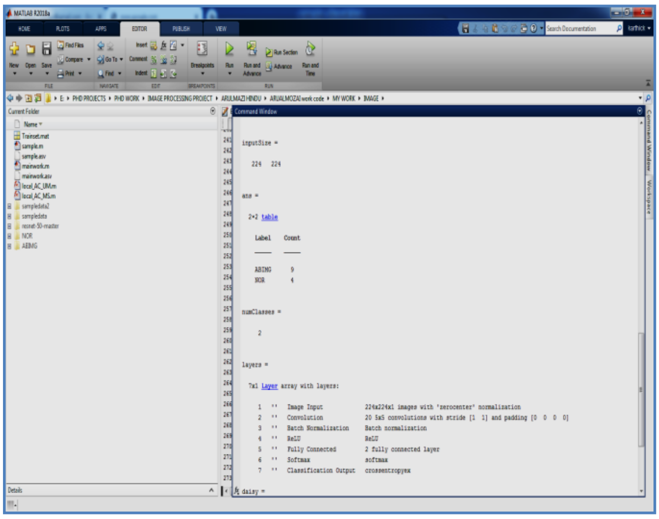


Figure 7: SegmentedImage



```
inputSize =

   224   224


ans =

  2×2 table

    Label    Count
    _____    _____

    ABIMG      9
    NOR        4


numClasses =

     2


layers =

  7x1 Layer array with layers:

     1   ''   Image Input             224x224x1 images with 'zerocenter' normalization
     2   ''   Convolution             20 5x5 convolutions with stride [1  1] and padding [0  0  0  0]
     3   ''   Batch Normalization     Batch normalization
     4   ''   ReLU                    ReLU
     5   ''   Fully Connected         2 fully connected layer
     6   ''   Softmax                 softmax
     7   ''   Classification Output   crossentropyex

fx daisy =
```

Figure 8: CNN Layer Work

Figure 7 shows the segmented image and Figure 8 shows the work done in the CNN pooling layer.

8.1 Results after Denoising

The performance of the system after applying various filters is analyzed. It is seen that DWT offers 82.4% and 55% better Peak Signal-to-Noise Ratio (PSNR) in contrast to Median and Weiner filters respectively (Figure 9).

Similarly, DWT offers 53.7% and 39.5% reduced Mean Squared Error (MSE) in contrast to Median and Weiner filters respectively (Figure 10).

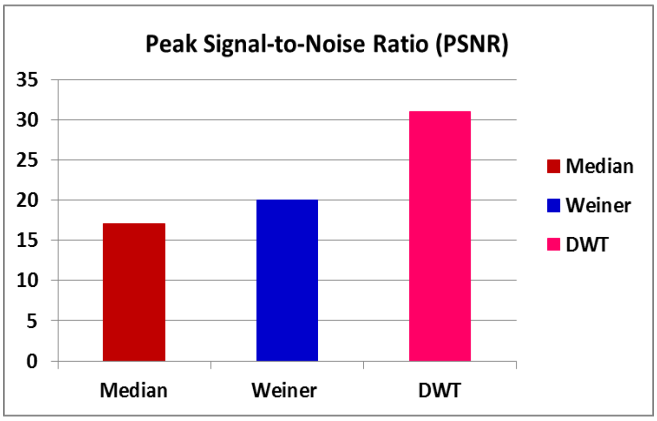


Figure 9: Peak Signal-to-Noise Ratio (PSNR)

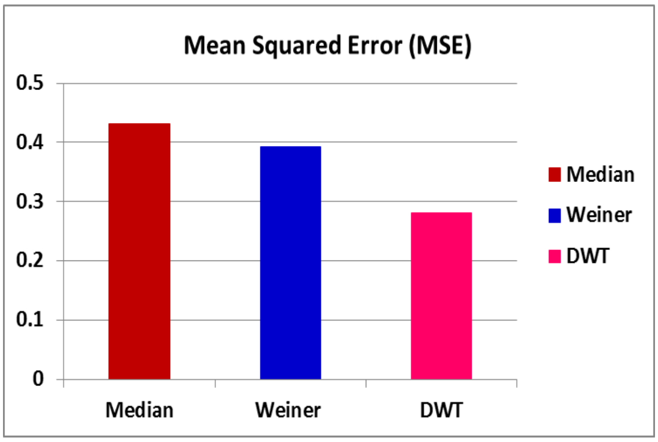


Figure 10:Mean Squared Error (MSE)

Similarly, DWT offers 6.7% and 3.3% better Structural Similarity Index Measure (SSIM) in contrast to Median and Weiner filters respectively (Figure 11).

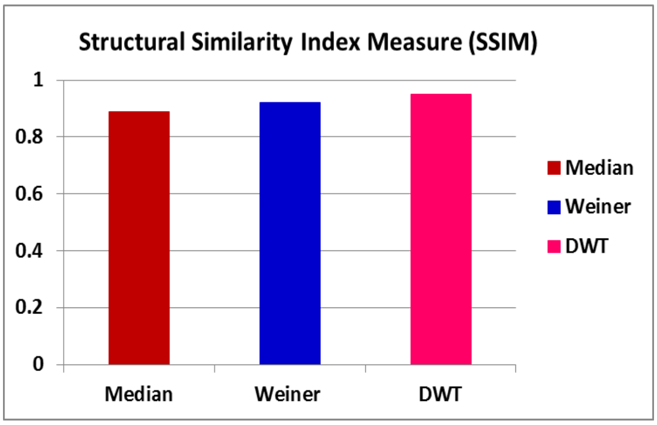


Figure 11:Structural Similarity Index Measure (SSIM)

8.2 Results after Classification

The results of classification are discussed below. It is seen that the proposed PSO-CNN offers 5% better accuracy (Figure 12), 6% better Precision (Figure 13), 5% better Recall (Figure 14) and 4% better F-Measure (Figure 15) in contrast to the existing CNN.

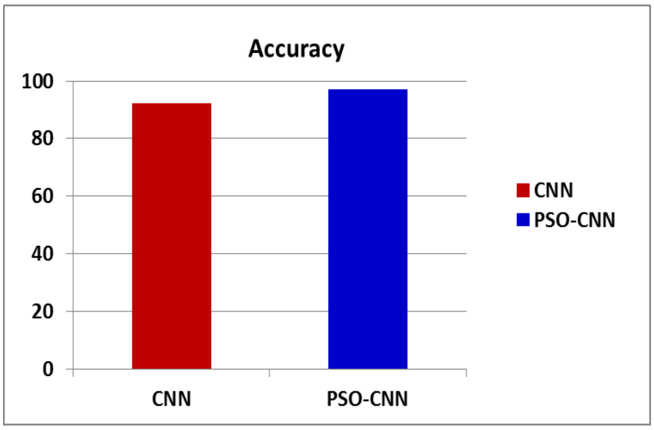


Figure 12: Accuracy

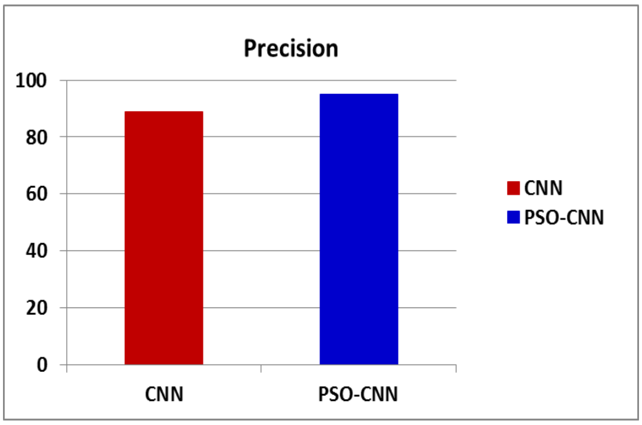


Figure 13: Precision

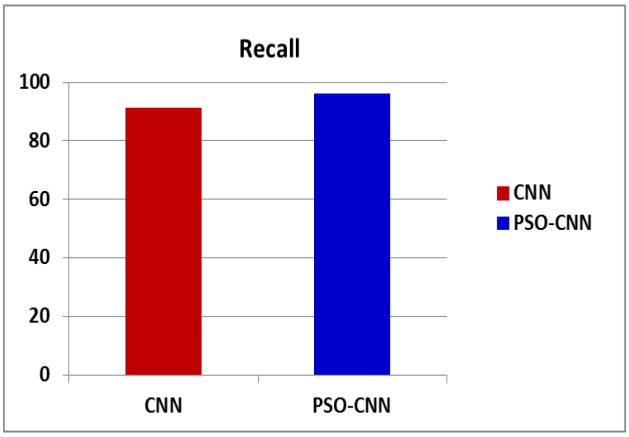


Figure 14: Recall

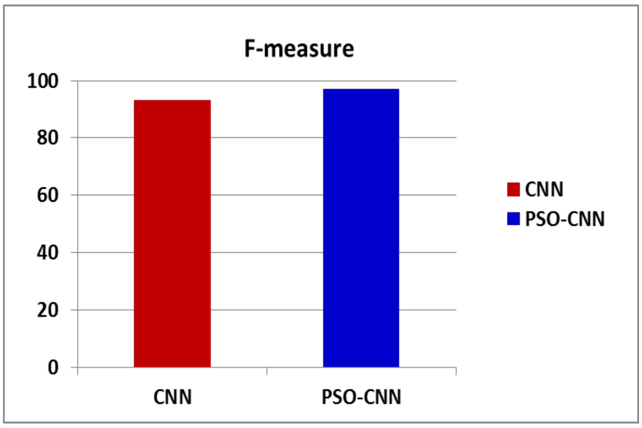


Figure 15: F-measure

9Conclusion

An innovative algorithm to search for deep Convolutional Neural Networks (CNNs) model based on PSO (PSO-CNN) is propounded in this paper. In addition to this, a novel encoding approach is also propounded. Two blocks of CNN architecture is considered wherein, one block holds just conv and pool layers whereas the other layer is the FC layer. This encoding system considers adjustable length CNN designs to be likened and joined utilizing a customary PSO algorithm. From the results it is evident that PSO-CNN can rapidly discover an optimized CNN architecture for a given dataset. With just 30 particles and 20 cycles, the algorithm discovers models fit for accomplishing test errors similar to those plans using more unpredictable and convoluted structures. From the exploratory outcomes, it is inferred that PSO-CNN would have the option to discover better designs if more computational force is accessible. In the future, the proposed PSO-CNN can be utilized for comparingthe objectives in the search process.

References

[1]Wei, Z., Li, G., & Qi, L. (2006). New nonlinear conjugate gradient formulas for large-scale unconstrained optimization problems. Applied Mathematics and Computation, 179(2), 407-430.

[2]Carvalho, M., &Ludermir, T. B., Particle swarm optimization of neural network architectures andweights. In 7th IEEE International Conference on Hybrid Intelligent Systems (HIS 2007), pp. 336-339, 2007.

[3]Siebel, N. T., & Sommer, G., Evolutionary reinforcement learning of artificial neural networks. International Journal of Hybrid Intelligent Systems, 4(3), 171-183, 2007.

[4]Stanley, K. O.,D'Ambrosio, D. B., & Gauci, J., A hypercube-based encoding for evolving large-scale neural networks. Artificial life, 15(2), 185-212, 2009.

[5]Glorot, X., &Bengio, Y., Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp. 249-256, 2010.

[6]Sahu, A., Panigrahi, S. K., &Pattnaik, S., Fast convergence particle swarm optimization for functions optimization. Procedia Technology, 4, 319-324, 2012.

[7]Hu, W., & Yen, G. G., Adaptive multi-objective particle swarm optimization based on parallel cell coordinate system. IEEE Transactions on Evolutionary Computation, 19(1), 1-18, 2013.

[8] Simonyan, K., & Zisserman, A., Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[9] He, K., Zhang, X., Ren, S., & Sun, J., Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision, pp. 1026-1034, 2014.

[10] Ioffe, S., &Szegedy, C., Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

[11] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., &Rabinovich, A., Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1-9, 2015.

[12] Ioffe, S., &Szegedy, C., Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

[13] Liang, M., & Hu, X., Recurrent convolutional neural network for object recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3367-3375, 2015.

[14] Chan, T. H., Jia, K., Gao, S., Lu, J., Zeng, Z., & Ma, Y., PCANet: A simple deep learning baseline for image classification?. IEEE transactions on image processing, 24(12), 5017-5032, 2015.

[15] He, K., Zhang, X., Ren, S., & Sun, J., Deep residual learning for image recognition, arXiv preprint arXiv:1512.03385, 2016.

[16] He, K., Zhang, X., Ren, S., & Sun, J., Proceedings of the IEEE conference on computer vision and pattern recognition. Convolutional Pose Machines, 4724-4732, 2016.

[17] Real, E., Moore, S., Selle, A., Saxena, S., Suematsu, Y. L., Tan, J., Le Q.V., &Kurakin, A., Large-scale evolution of image classifiers. arXiv preprint arXiv:1703.01041, 2017.

[18] Liu, H., Simonyan, K., Vinyals, O., Fernando, C., &Kavukcuoglu, K., Hierarchical representations for efficient architecture search. arXiv preprint arXiv:1711.00436, 2017.

[19] Qolomany, B., Maabreh, M., Al-Fuqaha, A., Gupta, A., &Benhaddou, D., Parameters optimization of deep learning models using Particle swarm optimization. In 13th IEEE International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 1285-1290, 2017.

[20] Kenny, A., & Li, X., A study on pre-training deep neural networks using particle swarm optimisation. In Asia-Pacific Conference on Simulated Evolution and Learning, Springer, Cham, pp. 361-372,2017.

[21] Sun, Y., Xue, B., Zhang, M., & Yen, G. G., A particle swarm optimization-based flexible convolutional autoencoder for image classification. IEEE transactions on neural networks and learning systems, 30(8), 2295-2309, 2018.

[22] Wang, B., Sun, Y., Xue, B., & Zhang, M., Evolving deep convolutional neural networks by variable-length particle swarm optimization for image classification. In IEEE Congress on Evolutionary Computation (CEC), pp. 1-8, 2018.

[23] Sun, Y., Yen, G. G., & Yi, Z., Evolving unsupervised deep neural networks for learning meaningful representations. IEEE Transactions on Evolutionary Computation, 23(1), 89-103, 2018.

[24] Jin, H., Song, Q., & Hu, X., Auto-keras: An efficient neural architecture search system. In Proceedings of the 25thACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1946-1956, 2019.

[25] Sun, Y., Xue, B., Zhang, M., & Yen, G. G., Evolving deep convolutional neural networks for image classification. IEEE Transactions on Evolutionary Computation, 24(2), 394-407, 2019.