

Deep Learning based Approach for Bone Diagnosis Classification in Ultrasonic Computed Tomographic Images

Marwa Fradi¹, Mouna Afif², Mohsen Machhout³

Monastir University, Physic Department of Faculty of sciences of Monastir, Tunisia

Abstract—Artificial intelligence (AI) in the area of medical imaging has shown a developed technology to have automatically the true diagnosis especially in ultrasonic imaging area. At this light, two types of neural networks algorithms have been developed to automatically classify the Ultrasonic Computed Tomographic (USCT) images into three categories, such as healthy, fractured and osteoporosis bone USCT images. In this work, at first step, a Convolutional Neural Network including two types of CNN models such (Inception-V3 and MobileNet) are proposed as a classifier system. At second step, an evolutionary neural network is proposed with the AmeobaNet model for USCT image classification. Results achieve 100% for train accuracy and 96%, 91.7% and 87.5% using Amoebanet, Inception-V3 and MobileNet respectively for the test accuracy. Results outperforms the state of the art and prove the robustness of the proposed classifier system with a short time process by its implementation on GPU.

Keywords—USCT; Inception-V3; MobileNet; Ameobanet-V2; classification; accuracy; transfer deep learning

I. INTRODUCTION

Nowadays, medical image classification presents the key technique for Computer Aide Diagnosis using deep learning approaches such as CNN, ANN and Evolutionary Neural works. In previous times, several types of research had been done to automatically classify medical images to have the true class diagnosis with the same way as a specialist would, especially in the area of ultrasound images classification [1]. Recently, artificial intelligence algorithms based computer Aided Diagnosis (CAD) have known a big revolution. Many researches have tried to create the similarity between human brain and computer machine by employing neural network algorithms as well as deep learning techniques. Nowadays, neural networks present an important tool for extracting information from medical images by classifying them automatically with a short time process. Moreover, Deep Learning has made a very huge development in medical image analysis and then in ultrasonic medical image classification to identify automatically a complete piece of information [2].

II. STATE OF THE ART

Given the difficulty to classify medical images automatically researchers have carried on the process of image classification based on deep learning programs as well as transfer deep learning [3]. In Fact, Transfer learning is the concept of training a pre-trained neural network model on our small dataset. Related works have used machine learning in

the beginning, then transfer deep learning for medical image classification, segmentation and object detection moving from a pre-trained model, trained on big data to a small data. In recent works [4], a classifier based on fractional Fourier transform combined with SVM has been proposed to classify IRM brain images into pathological and healthy brain images. However, performance results are good, It was shown that the proposed architecture was adapted only for a small number of dataset [4, 5]. However, CNN is used in the classification of numbers and recognition of numbers with Leucin [6]. The approach of transfer deep learning was based on employing a pre-trained network, which was trained on a large number of samples for a similar task, for a new task with little annotated image data. In 2012, in [7]. Krizhevsky has published the first deep learning model with an error rate of 15.3 however Google Net has achieved 6.7% error rate in 2014. Indeed various CNN optimized algorithms have been used such as LeNet, AlexNet, ZF Net, Google Net, VGGNet and ResNet in medical image analysis, proving their efficiency. Therefore deep learning algorithms especially the convolutional neural network are rapidly emerging as an efficient method for medical image classification and then the fast diagnosis detection [8]. MobileNet has been used in [9] for skin lesion classification giving promising results with high accuracy, specificity and F-1 score, thus these results were improved by the big data augmentation and up-sampling process. Moreover, in [10] authors have used collected X-rays images, to be classified for covid-19 detection, using deep learning algorithms, thus they achieved the best accuracy, sensitivity, and specificity obtained is 96.78%, 98.66%, and 96.46% respectively. Also, MobileNet has been used in [9] for skin lesion classification giving promising results with high accuracy, specificity and F-1 score, thus these results were improved by the big data augmentation and up-sampling process. In [11], authors have employed a deep model and statistic feature fusion for feature extraction with a multilayer perceptron for medical image classification giving high classification results. However, in [12], a transfer learning has been implemented on two convolutional neural network models such as VGG16 and InceptionV3, for Pneumonia Detection. Accordingly, deep neural network-based methods [13] provides high performance in classifying the images according to the extracted features solving the issue of handcrafted feature extraction. In other hand, in previous works methods based on artificial neural networks (ANNs) have attracted more attention especially, in the area of ultrasonic images classification [14]. The Artificial Neural networks shows its

huge role in medical image diagnosis class detection. Accordingly, it has been implemented in [15] for breast cancer class detection, showing a high accuracy results which proves the interest of the ANN use. To summarize and taking in consideration the achieved results in the state of the art, CNN models have proved to get good results in terms of classification accuracies and can be implemented on embedded systems like MobileNet, in contrast to the Evolutionary neural networks, which gives excellent results higher than the results achieved by a CNN, but it cannot be as a real time application given its huge size and the complexity of its architecture.

Our approach in this paper consists of a data augmentation of the number of USCT images to achieve a big data augmentation by pre-processing algorithms. Thus we solve the issue of USCT data unavailability [16]. Then, a transfer deep learning models have been done such as Convolutional Neural Networks models (Inception V3, MobileNet-v2 and Evolutionary Neural Network model such as AmoebaNet are applied on our dataset. Thus, the aim of this work is automatically classify USCT images with the same way a clinician would, and enables them to get the true diagnosis in short time process.

III. METHOD

The proposed system classifier for USCT images class diagnosis detection is depicted and detailed in Fig. 1.

A. USCT Pre-Processing

USCT image pre-processing is with huge interest given the ambiguity of the obtained USCT images, similarly to computed tomography (CT) images, where noise removal shows to be a primordial step to begin with [17,18] using Fuzzy approaches. This process leads to get a computed tomography image augmentation.

Big data augmentation is an approach to overcome the challenges posed by a limited amount of annotated training data. Augmentation is performed artificially generating more annotated training data, typically by mirroring and rotating the original images given that the difficulty to have a big number of ultrasonic computed tomographic images, image pre-processing to augment the number of images was a crucial step that should be done. For this fact, we have augmented the image number from 30 images to 250 images with a size of 256*256 by morphologic algorithms such as dilatation and erosion, then by simple rotation and finally by image processing algorithms such as Haar Wavelet transform, which is in more details in [19], K-Means and Ostu method.

1) *K-Means algorithm*: It is based on grouping similar data points into clusters. There is no prediction involved. Its algorithm is illustrated by these steps [20].

- Fix the k number of cluster values.
- Identify the k cluster centers.
- Determine the cluster center.
- Determine the pixel distance for each cluster center.

- If the distance is close to the center value, budge to that cluster.
- Otherwise, move to the next cluster and Re-identify the center.

The pre-processing step has an interesting effect on USCT dataset augmentation as depicted in Fig. 2.

2) *Haar wavelet*: A low-pass filter application remains to get an L image which is compressed and the application of a high-pass filter leads to obtain an H image which introduces image details. This process is as depicted in Fig. 3. It is done by equations (1) and (2) as follows:

$$YH[k] = \sum_n x[n] G[2k - n] \tag{1}$$

$$YL[k] = \sum_n x[n] H[2k - n] \tag{2}$$

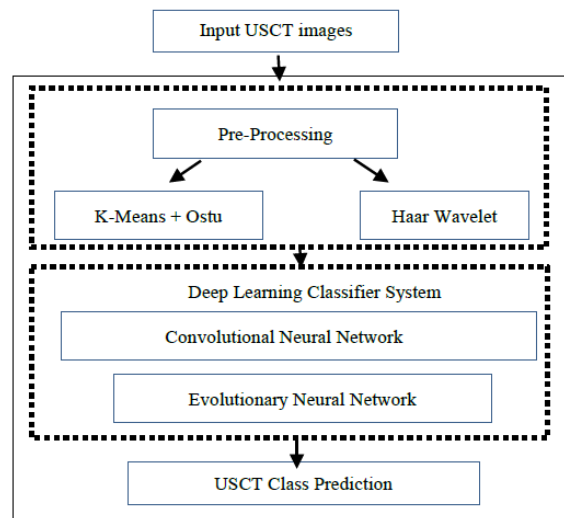


Fig 1. Synoptic Flow of the Proposed Method.

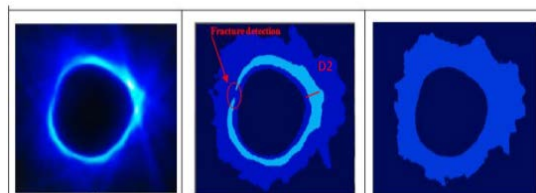


Fig 2. Augmented USCT Dataset by k-Means Algorithm.

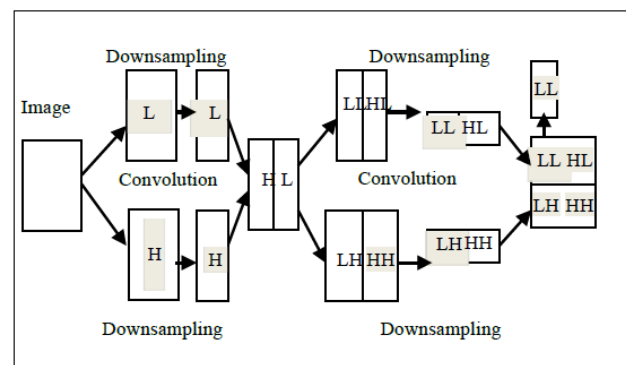


Fig 3. Principle of Haar Wavelet Application.

3) *Otsu-method*: The Otsu method of the threshold is the most powerful and global threshold method. It performs image binarization based on the histogram image shape. It assumes that the image for binarization contains only foreground and background pixels Using the simple formula in the Otsu algorithm, we get:

$$\sigma^2 = \Psi A (u A - u)^2 + \Psi B (u B - u)^2 \quad (3)$$

A.4. USCT augmented data

The proposed pre-processing, applied on USCT images, shows a big data augmentation and offer a free USCT database for USCT researchers given the challenge of USCT dataset as shown in Fig. 4.

B. Deep Learning Classifier System

1) Convolutional Neural Network

a) Data Training

- **Frameworks:** We have used 250 images divided into three classes, a Linux operating system, NVIDIA Graphic Interface named Titan. With GPU computing deep learning is 10 to 30 times faster than on CPU.
- **Libraries:** For the data training we have needed many libraries that should be installed such as TensorFlow, keras, Numpy, Matplotlib and OpenCv.
- **Training Parameters:** We use the TensorFlow library and train the networks with stochastic gradient descent, with learning rate 10⁻³, momentum 0.9, no weight decay and batch size of 10 images per step. There is no need for jittering as instead of data augmentation we can simply generate more synthetic training data. Input images are resized to 256 × 256. Deep Learning Neural Networks Models have been applied with a modification in the last layer by introducing our fully connected neural network layer. We have three classes USCT images with a size of 256*256, the first was healthy images, the second osteoporosis images and the last one was the fractured images.

b) *Transfer Learning*: The transfer learning is essentially based on the use of pre-trained NetWork Model to try to work around the perceived requirement of a large dataset. The training parameters are in details in Table I.

The training process of our implemented deep learning algorithm is as depicted in the Fig. 5.

2) Classical Convolutional Neural Network

a) *Convolutional Layer*: It is a based step in Deep learning, by the application of a filter kernel with its weights value to the input image. It has two principal functions. At first step, a kernel multiplication has been done to each pixel value covered by this kernel then all image pixel values have been summed to have the feature maps then the output of the first map will be the input of the next map, repeating the same process but with a big numbers of interconnected neurons.

b) *Max Pooling Layer*: It is an operation based on downsampling the output convolutional images. It outputs the

max value in a local neighbourhood of each feature pas it is illustrated by Fig. 6 [6].

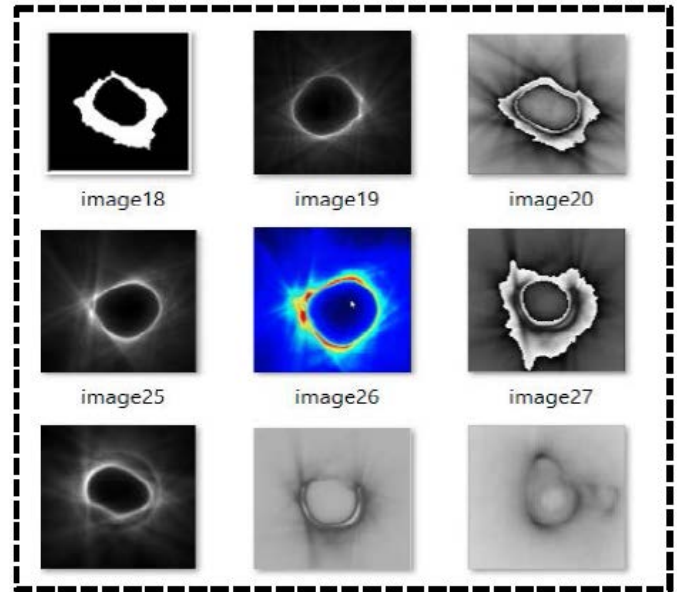


Fig 4. Augmented USCT Dataset.

TABLE I. TRAINING DATA PARAMETERS

Training Parameters	Number
Iterations	4000
Hidden Layers	10
Train-images	190
Validation-images	30
Test -images	30
Error-rate	0.001
Batch	10 Images/Step

```

0% (N=10)
INFO:tensorflow:2019-03-20 10:52:36.974300: Step 3950: Train accuracy = 100.0%
INFO:tensorflow:2019-03-20 10:52:36.974713: Step 3950: Cross entropy = 0.000000
INFO:tensorflow:2019-03-20 10:52:36.984929: Step 3950: Validation accuracy = 100.0% (N=10)
INFO:tensorflow:2019-03-20 10:52:37.093775: Step 3960: Train accuracy = 90.0%
INFO:tensorflow:2019-03-20 10:52:37.094046: Step 3960: Cross entropy = 0.605381
INFO:tensorflow:2019-03-20 10:52:37.101454: Step 3960: Validation accuracy = 100.0% (N=10)
INFO:tensorflow:2019-03-20 10:52:37.212316: Step 3970: Train accuracy = 100.0%
INFO:tensorflow:2019-03-20 10:52:37.212540: Step 3970: Cross entropy = 0.000001
INFO:tensorflow:2019-03-20 10:52:37.220535: Step 3970: Validation accuracy = 100.0% (N=10)
INFO:tensorflow:2019-03-20 10:52:37.339074: Step 3980: Train accuracy = 100.0%
INFO:tensorflow:2019-03-20 10:52:37.339393: Step 3980: Cross entropy = 0.000000
INFO:tensorflow:2019-03-20 10:52:37.351399: Step 3980: Validation accuracy = 100.0% (N=10)
INFO:tensorflow:2019-03-20 10:52:37.543588: Step 3990: Train accuracy = 90.0%
INFO:tensorflow:2019-03-20 10:52:37.544066: Step 3990: Cross entropy = 0.232043
INFO:tensorflow:2019-03-20 10:52:37.571098: Step 3990: Validation accuracy = 90.0% (N=10)
INFO:tensorflow:2019-03-20 10:52:37.723482: Step 3999: Train accuracy = 100.0%
INFO:tensorflow:2019-03-20 10:52:37.723997: Step 3999: Cross entropy = 0.000000
INFO:tensorflow:2019-03-20 10:52:37.736003: Step 3999: Validation accuracy = 100.0%
    
```

Fig 5. Train Data Accuracy.

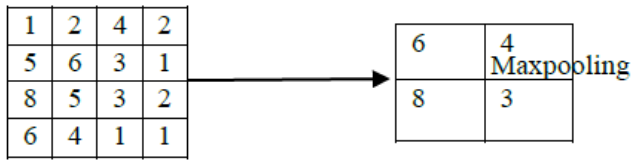


Fig 6. Max-Pooling.

c) *Classifier Layer*: The classifier layer is based on the fully connected layers. It plays a crucial role to classify images based the detected features. A fully connected layer is a layer whose neurons have full connections to all activation in the previous layer.

The classical CNN, consisting of a convolutional layer, a max pooling layer and a classifier layer is a depicted in Fig. 7.

3) *Proposed architecture*: Our proposed architecture was based on a transfer deep learning models such as AmoebaNet [21, 22], Inception-V3[1], Mobile net [23] and NasNet to classify USCT images automatically into three classes. Our approach is to modify the last fully connected layer by our FCNN layer which is composed by three categories: healthy images, osteoporosis images and fractured image. Then with our proposed architecture we have implemented it into Graphic interface processor GPU. All models have roughly the same proposed architecture by a modification of the last layer with our FCNN layer. In the example below in figure Inception –V3 architecture is detailed.

4) *Transfer deep learning inception-v3 model architecture*: In this part of the work, we try to transfer knowledge in order to develop a new USCT image classification system using the third generation of the inception family: “inception v3” [1].

This architecture provides 42 layers, it demonstrates more computational efficiency compared to the pervious inception family architectures. Inception v3 architecture presents a promising network composition with different parameters optimization as depicted in Fig. 8.

In order to ensure more robustness of the neural network, a 5 x 5 convolution is replaced by two 3 x 3 convolutions. By using this technique, the parameters number is decreased from 25 to 18 parameter. This technique reduces considerably the network complexity. Also, this technique contributes for the inception v3 powerful block named “inception module-A”. The inception module-A architecture is provided in Fig. 9.

One convolution of 3 x 3 is replaced by two convolutions of 1 x 3. This modification participates in the “inception module B”. In the following figure the inception module B is presented.

Another module named “inception module C is proposed in inception v3 model as presented in Fig. 11. All these modules come on reducing the number of parameters of the whole network and minimize the risk of overfitting.

5) *USCT image classification using MobileNet v2*: Deep convolutional neural networks (DCNN) have revolutionized computer vision area in the last few years. In this section, we

aim to develop a new powerful USCT image classification and recognition system based on the lightweight neural network mobileNet v2 architecture.

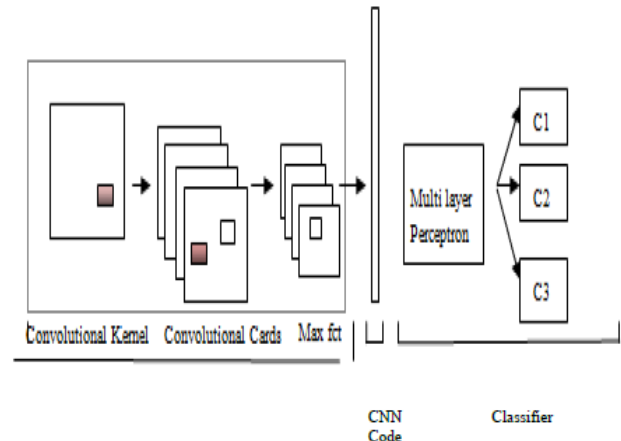


Fig 7. Classical CNN Architecture.

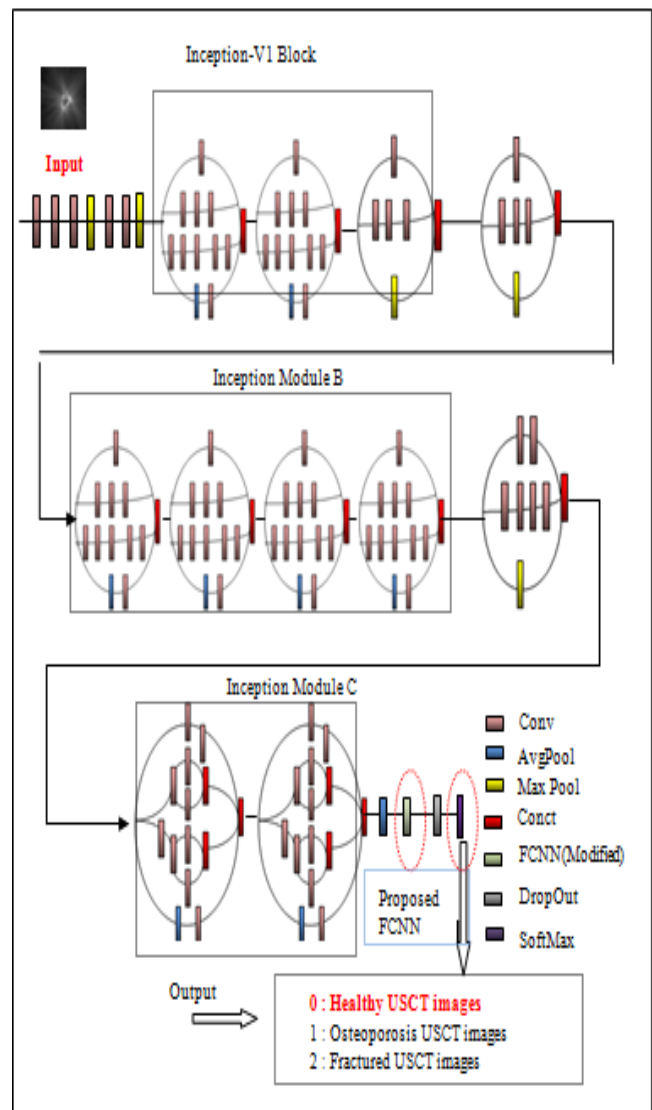


Fig 8. Proposed Inception -V3 Architecture.

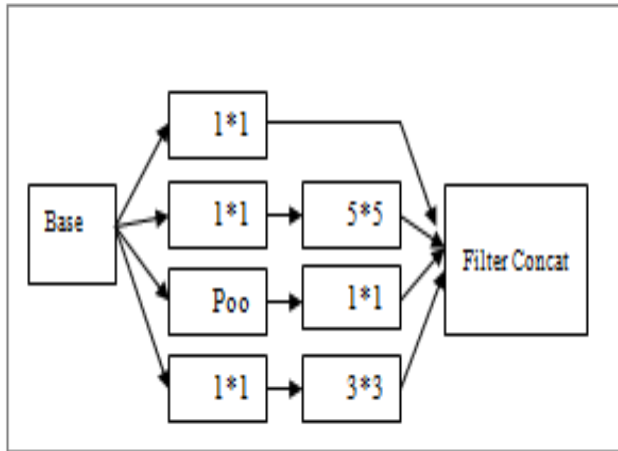


Fig 9. Inception-Module A.

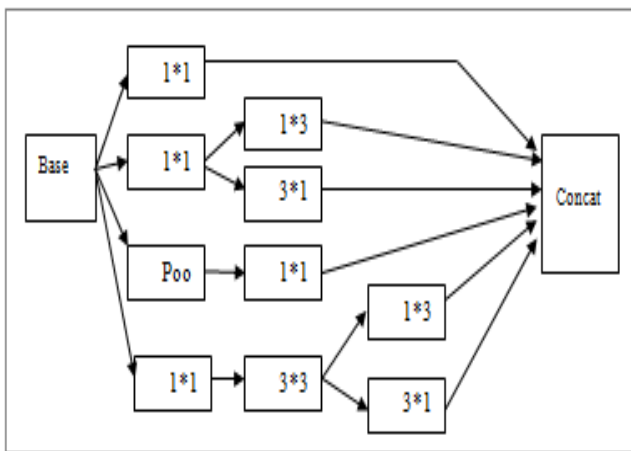


Fig 10. Inception Module B.

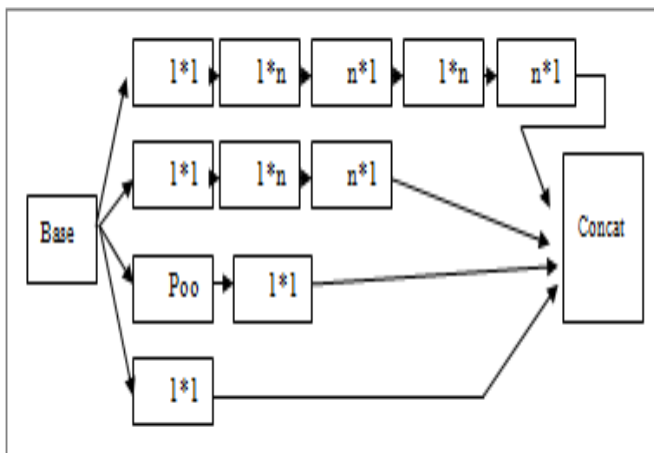


Fig 11. Inception Module C.

MobileNet v2 architecture as presented in Table II introduces a new powerful block named the “inverted residual block with linear bottleneck”. All the features extracted are filtered by using a lightweight depthwise convolution. By using the depthwise convolution, we ensure a considerable reduce of the network parameters.

TABLE II. MOBILENET V2 ARCHITECTURE

Input	Layer	n
224x224x3	Conv 2D	1
112x112x32	Bottleneck	1
112x112x16	Bottleneck	2
56x56x24	Bottleneck	2
28x28x32	Bottleneck	2
14x14x64	Bottleneck	1
14x14x96	Bottleneck	2
7x7x160	Bottleneck	1
7x7x320	Conv 2D 1x1	1
7x7x1280	Average pooling 7x7	-
1x1x1280	Conv 2D 1x1	-

Another powerful fact present in the mobileNet v2 architecture is the use of the pointwise convolution before the use of depthwise separable convolution. This operation is named “bottleneck”. MobileNet v2 [24] architecture present the updated version of MobileNet v1 version [23]. This updated version is promising as it provides three main new components which are the following:

- Linear bottleneck layer
- Shortcut connection
- Inverted residual block

The inverted residual layer enables the feature map to e encoded in low-dimensional sub-space. The bottleneck layer appears very similar as the residual block where each block contains the input representation followed by the explanation layer. As bottleneck layers contains the necessary information and acts as an implementation of non-linear transformation, for this fact it introduces the shortcut connection directly between bottleneck layers.

To summarize, the MobileNet v2 architecture is composed of a regular convolution followed by 11 bottleneck layers and a pointwise convolution and an average pooling layer, another pointwise convolution layer then it ends with a fully connected layer and a softmax layer used to classify the objects categories.

C. USCT Image Classification: Approach Adopted based on Evolutionary Algorithms

In order to improve deep learning models, an effort was devoted to apply new techniques named “evolutionary algorithms” to neural network topologies. State-of-the-art results obtained by this type of algorithms have demonstrated their higher performances compared to human-crafted ones. In this part of the proposed work, we propose to develop new USCT image classification systems using aging evolutionary algorithm Amoebanet-A [21, 22].

The AmoebaNet architecture make two additional variables to the standard evolutionary algorithms: firstly, it proposes a new tourmenant selection [25] process for evolutionary algorithms. In the standard tourmenant selection,

the best neural network architecture or genotype are kept while on the AmoebaNet architecture each generated architecture or genotype is associated with a specific age parameter and bias. In this stage, the tournament selection is charged to keep the younger genotypes. This is the evolutionary algorithm with aging architecture technique. Secondly, AmoebaNet perform a set of mutations in a simpler way in the NasNet Search Space (NAS) [26]. NAS search space presents a space of image classifiers. It uses reinforcement learning technique to search for the best neural topologies. Applying the neural architecture search NAS to a huge dataset require a huge computational resource. To address this problem, it processes on smaller datasets and transfer the learned model on largest datasets. All architectures obtained using the NAS search space are independent and various in terms of input image size as well as the network depth.

Every neural network present in the NAS search space as shown in Fig. 12 is composed of convolution layer or (calculation cell). We note that the NAS search space search for the best calculation cell structure. By searching for the best calculation layer, the NAS search space presents a much faster way of neural architectures search as well as the obtained calculation cells are able to be generalized for other tasks types. The following figure provides the NAS architecture used in ImageNet [7] dataset.

Architectures obtained by the NAS search space are in a feed-forward stocks. Each layer of the proposed neural network receives two inputs one direct input and a skip input from its previous layer. Layers provided are from two different categories: normal and reduction cells. The two types of cells are safe but the only difference between the two cells is that the reduction cell is followed by a stride of 2 in order to minimize the feature map dimensions while the normal cell keep the same input size.

Once the neural network architecture is specified, we have to specify two free parameters that are used during training N , and F while N is the number of normal cells and F is the number of convolution layers and output filters. In the proposed work, we set $N=18$ and $F=448$.

In order to develop our USCT image classification system, we used the aging evolutionary algorithm AmoebaNet-A. Models population will be initialized randomly by architectures from the NAS search space. The model that present the best calculation cells will be taken as a parent. A new architecture will be obtained using the mutation operation to obtain the child architecture. The child architecture is kept to be used to train and test the classification system. Child architectures are obtained by using three types of mutations:

- Hidden state mutation
- Identity mutation
- Op mutation

We note that when training the neural network architecture, one of these three mutations is applied randomly.

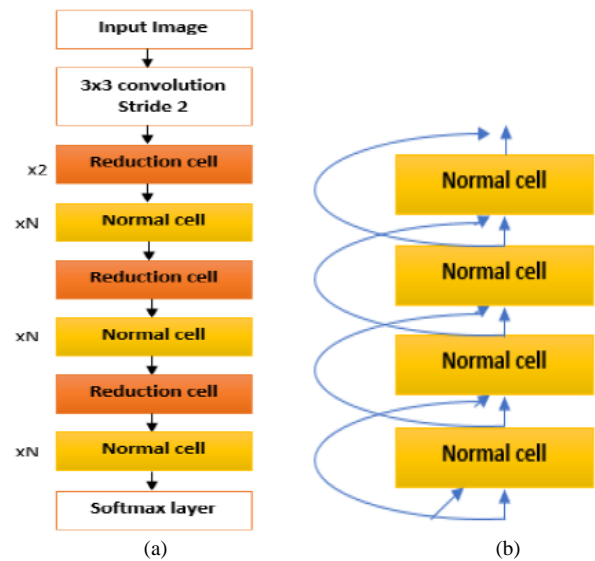


Fig 12. (a) NAS Search Space, (b) Detailed Architecture with Skip Input.

IV. RESULTS

A. Train-test Accuracy Results

Due to transfer deep learning on our dataset, we have achieved 100% of accuracy in train process and 96% in test process classification accuracy as shown in Table III. More we are going deeper more the classification test accuracy is improved. We have achieved the top accuracy USCT image classification with AmoebaNet Model by 96% of test-accuracy where Inception-V3 comes with the second top accuracy with 91.7%.

B. Histogram of Classification Results

Regarding obtained results as illustrated in the Figure.13 by the histogram, AmoebaNet with 32 layers has the top test accuracy against results accuracy given by Inception-V3 with its 42 layers, MobileNet with its 28 layers and NasNet with Deep learning Models. We conclude that when we are going deeper, more results accuracy is enhanced.

C. Time Process Results

Our framework is based on the python language of the Keras package and on an Nvidia Titan GPU using a Linux operating system. Graphics cards (GPU) are characterized by the large number of cores that the processors allow as well as very large memory integrated with these processors. They are very useful for several computer tasks, precisely for software implementations like deep learning algorithms.

TABLE III. TRANSFER DEEP LEARNING ACCURACY RESULTS

Algorithm	Image Resolution	Train Accuracy	Test Accuracy
Mobile Net	256*256	100%	87.5%
NasNet	256*256	100%	75%
Ameobanet	256*256	100%	95.8%
SOM [4]	256*256	100%	94%
Inception-V3	256*256	100%	91.7%

Test Accuracy Classification Results

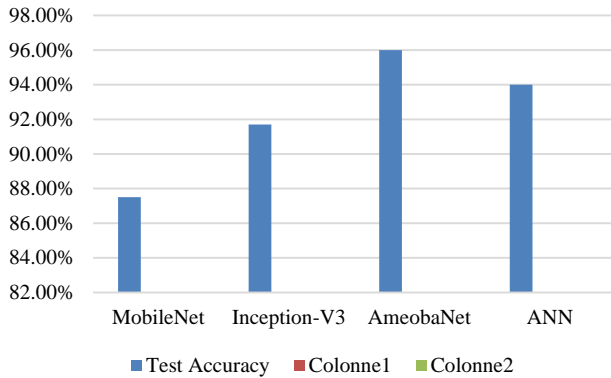


Fig 13. Histogram Results of Test-Accuracy USCT Images Classification.

Due to our implementation of our proposed neural networks models on GPU as shown by Annexes 1, 2, 3 and 4 in Annexes section , we have gained time process speed with 47.8 mn to each model trained on our USCT images against 149 mn with CPU models implementations as detailed by the following table.

TABLE IV. TIME-PROCESS RESULTS ON GPU AND CPU

Algorithm	CPU-Time- Process	GPU-Time-Process
MobileNet	146.9 mn	48mn
Inception-V3	145.52mn	47.6mn
AmeobaNet	148mn	49mn

V. DISCUSSIONS

A comparative study was done with previously works, given results were shown in the Table. 5 below. AmeobaNet as an evolutionary deep neural network has achieved the best accuracy in our work and outperforms recent works [18, 19] with 22% of accuracy. Indeed, it is explained by the mind of the evolutionary algorithms which aims to search for the best architecture that can be used for a desired task. The architecture search is done in the network architecture search space (NAS). However, The Inception-V3 comes by a promising results accuracy with a value of 91.7% surpassing the state of the art with a test accuracy value equal to 11%. The inception v3 module computes multiple different transformations over the same input map in parallel, connecting the results into a single output. For each layer, it does a 5x5 convolution, 3x3 convolution, and max pooling, each carries different information, which of course is computationally costly. Therefore the authors of Inception decided to overcome this problem by introducing the dimension reductions. What is meant by dimension reduction is by using 1x1 convolution before going to the bottlenecks 3x3 and 5x5 convolutions. Therefore it has the compressed version of the spatial information. We have outperforming previously works. In Fact, we overcome [27] with inception -v3 with 13% and with SOM by 14% .With AmeobaNet we have overcome the major related works with neural networks classification models against Alex Net, NasNet, Inception V3 and MobileNet models accuracy classification [4, 5, 7, 24, 27].

TABLE V. COMPARATIVE STUDY WITH THE STATE OF THE ART

Algorithm	Our Work	Related Work
Mobil Net V2	87.5%	89.9% [24]
NasNet	75% %	80.8%
Ameobanet	96%	74,5 [21,22]
SOM [14]	94%	79% [3]
Inception-V3	91.7%	78.8% [27, 4, 5]

VI. ANNEXES

```

ing/saver.py:1266: checkpoint_exists (from tensorflow.python.training.checkpoint
management) is deprecated and will be removed in a future version.
Instructions for updating:
Use standard file APIs to check for files with this prefix.
INFO:tensorflow:Restoring parameters from /tmp/retrain_checkpoint
INFO:tensorflow:Final test accuracy = 91.7% (N=24)
INFO:tensorflow:Save final result to : /tmp/output_graph.pb
2019-03-20 11:10:18.749128: W tensorflow/core/graph/graph_constructor.cc:1272] I
mporting a graph with a lower producer version 26 into an existing graph with pr
oducer version 27. Shape inference will have run different parts of the graph wi
th different producer versions.
INFO:tensorflow:Saver not created because there are no variables in the graph to
restore
2019-03-20 11:10:30.161044: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1
512] Adding visible gpu devices: 0, 1
2019-03-20 11:10:30.161262: I tensorflow/core/common_runtime/gpu/gpu_device.cc:9
84] Device interconnect StreamExecutor with strength 1 edge matrix:
2019-03-20 11:10:30.161290: I tensorflow/core/common_runtime/gpu/gpu_device.cc:9
90] 0 1
2019-03-20 11:10:30.161318: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1
003] 0:  N N
2019-03-20 11:10:30.161346: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1
003] 1:  N N
2019-03-20 11:10:30.162428: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1

```

Annex1. Inception V3 Transfer Deep Learning

```

WARNING:tensorflow:From /usr/lib/python2.7/site-packages/tensorflow/python/train
ing/saver.py:1266: checkpoint_exists (from tensorflow.python.training.checkpoint
management) is deprecated and will be removed in a future version.
Instructions for updating:
Use standard file APIs to check for files with this prefix.
INFO:tensorflow:Restoring parameters from /tmp/retrain_checkpoint
INFO:tensorflow:Final test accuracy = 95.0% (N=24)
INFO:tensorflow:Save final result to : /tmp/output_graph.pb
INFO:tensorflow:Saver not created because there are no variables in the graph to
restore
2019-03-20 10:54:58.027742: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1
512] Adding visible gpu devices: 0, 1
2019-03-20 10:54:58.029076: I tensorflow/core/common_runtime/gpu/gpu_device.cc:9
84] Device interconnect StreamExecutor with strength 1 edge matrix:
2019-03-20 10:54:58.029121: I tensorflow/core/common_runtime/gpu/gpu_device.cc:9
90] 0 1
2019-03-20 10:54:58.029150: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1
003] 0:  N N
2019-03-20 10:54:58.029183: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1
003] 1:  N N
2019-03-20 10:54:58.032400: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1
115] Created tensorflow device (/job:localhost/replica:0/task:0/device:GPU:0 wit
h 7496 MB memory) -> physical GPU (device: 0, name: Quadro M4000, pci bus id: 00
00:03:00.0, compute capability: 5.2)

```

Annex2. AmeobaNet Transfer Learning

```

ing/saver.py:1266: checkpoint_exists (from tensorflow.python.training.checkpoint
management) is deprecated and will be removed in a future version.
Instructions for updating:
Use standard file APIs to check for files with this prefix.
INFO:tensorflow:Restoring parameters from /tmp/retrain_checkpoint
INFO:tensorflow:Final test accuracy = 87.5% (N=24)
INFO:tensorflow:Save final result to : /tmp/output_graph.pb
2019-03-20 10:43:25.666513: W tensorflow/core/graph/graph_constructor.cc:1272] I
mporting a graph with a lower producer version 26 into an existing graph with pr
oducer version 27. Shape inference will have run different parts of the graph wi
th different producer versions.
INFO:tensorflow:Saver not created because there are no variables in the graph to
restore
2019-03-20 10:43:25.665955: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1
512] Adding visible gpu devices: 0, 1
2019-03-20 10:43:25.666508: I tensorflow/core/common_runtime/gpu/gpu_device.cc:9
84] Device interconnect StreamExecutor with strength 1 edge matrix:
2019-03-20 10:43:25.666579: I tensorflow/core/common_runtime/gpu/gpu_device.cc:9
90] 0 1
2019-03-20 10:43:25.666614: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1
003] 0:  N N
2019-03-20 10:43:25.666661: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1
003] 1:  N N
2019-03-20 10:43:25.668357: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1

```

Annex3. Mobile Net Transfer Learning

```
003] 0: N N
2019-03-20 11:02:23.136794: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1
003] 1: N N
2019-03-20 11:02:23.140213: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1
115] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 wit
h 7496 MB memory) -> physical GPU (device: 0, name: Quadro M4000, pci bus id: 00
00:03:00.0, compute capability: 5.2)
2019-03-20 11:02:23.145386: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1
115] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:1 wit
h 7860 MB memory) -> physical GPU (device: 1, name: Tesla K40c, pci bus id: 0000
:02:00.0, compute capability: 3.5)
WARNING:tensorflow:From /usr/lib/python2.7/site-packages/tensorflow/python/train
ing/saver.py:1266: checkpoint_exists (from tensorflow.python.training.checkpoint
_management) is deprecated and will be removed in a future version.
Instructions for updating:
Use standard file APIs to check for files with this prefix.
INFO:tensorflow:Restoring parameters from /tmp/retrain_checkpoint
INFO:tensorflow:Final test accuracy = 75.0% (N=74)
INFO:tensorflow:Save final result to /tmp/output_graph.pb
2019-03-20 11:02:05.043164: W tensorflow/core/graph/graph_constructor.cc:1272] I
mporting a graph with a lower producer version 20 into an existing graph with pr
oducer version 27. Shape inference will have run different parts of the graph wi
th different producer versions.
INFO:tensorflow:Saver not created because there are no variables in the graph to
```

Annex4. NasNet Transfer Learning Results

VII. CONCLUSION AND FUTURE WORK

In this paper, a transfer Deep Learning method has been performed for USCT image classifications into three classes such as healthy, fractured and osteoporosis USCT images. Using inception-V3, MobileNet, and AmeobaNet, we have overcome previously works and has achieved excellent results in the area of medical image classification with a top value of 96% of test classification accuracy for a short time process. For future work, we have to think about automatic USCT image segmentation to detect different structure automatically using a variable model structure of neural network in first step [28]. Then using CNN models. Furthermore, a hardware implementation of our code will be done on Pynq-FPGA. Thus we will get a medical real-time application.

REFERENCES

- [1] Shi, Z., He, L., Suzuki, K., Nakamura, T., & Itoh, H. (2009). Survey on neural networks used for medical image processing. *International journal of computational science*, 3(1), 86.
- [2] Review: NAS Net — Neural Architecture Search Network (Image Classification).
- [3] Altaf, F., Islam, S., Akhtar, N., & Janjua, N. K. (2019). Going Deep in Medical Image Analysis: Concepts, Methods, Challenges and Future Directions. arXiv preprint arXiv:1902.05655.
- [4] Y. Zhang, S. Chen, S. Wang, J. Yang, and P. Phillips, "Magnetic resonance brain image classification based on weighted-type fractional fourier transform and nonparallel support vector machine," *International Journal of Imaging Systems and Technology*, vol. 25, no. 4, pp. 317–327, 2015.
- [5] ZhiFei Lai, Hui Fang Deng "Medical Image Classification Based on Deep Features Extracted by Deep Model and Statistic Feature Fusion with Multilayer. In: Perceptron"Compt Intell Neuroscience Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA) (pp. 19-23). IEEE.
- [6] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, Gradient-based learning applied to document recognition, in *Proceedings of the IEEE*, vol. 86, no.11, pp. 2278-2324, Nov 1998
- [7] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).
- [8] Zheng, Y., Wang, H., & Hao, Y. (2020, April). CNN study of convolutional neural networks in classification and feature extraction applications. In *Big Data II: Learning, Analytics, and Applications* (Vol. 11395, p. 113950K). International Society for Optics and Photonics.
- [9] Sae-Lim, W., Wettayaprasit, W., & Aiyarak, P. (2019, July). Convolutional Neural Networks Using MobileNet for Skin Lesion

- Classification. In *2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE)* (pp. 242-247). IEEE.
- [10] LIU, Yannan, WEI, Lingxiao, LUO, Bo, et al. Fault injection attack on deep neural network. In : *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2017. p. 131-138.
- [11] Sae-Lim, W., Wettayaprasit, W., & Aiyarak, P. (2019, July). Convolutional Neural Networks Using MobileNet for Skin Lesion Classification. In *2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE)* (pp. 242-247). IEEE.
- [12] Weng, Y., Zhou, T., Li, Y., & Qiu, X. (2019). Nas-unet: Neural architecture search for medical image segmentation. *IEEE Access*, 7, 44247-44257.
- [13] Lai, Z., & Deng, H. (2018). Medical Image Classification Based on Deep Features Extracted by Deep Model and Statistic Feature Fusion with Multilayer Perceptron. *Computational intelligence and neuroscience*, 2018.
- [14] Fradi, M., Lasaygues, P., & Machhout, M. (2019, March). Auto-Organiser Neural Network Application for Ultrasound Computed Tomographic Image Classification. In *2019 19th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)* (pp. 19-23). IEEE.
- [15] Mehdy, M. M., Ng, P. Y., Shair, E. F., Saleh, N. I., & Gomes, C. (2017). Artificial neural networks in image processing for early detection of breast cancer. *Computational and mathematical methods in medicine*, 2017.
- [16] Ruiter, N. V., Zapf, M., Hopp, T., Gemmeke, H., & van Dongen, K. W. (2017, March). USCT data challenge. In *Medical Imaging 2017: Ultrasonic Imaging and Tomography* (Vol. 10139, p. 101391N). International Society for Optics and Photonics.
- [17] Kaur, P., Sharma, P., & Palmia, A. (2020). Fuzzy clustering-based image segmentation techniques used to segment magnetic resonance imaging/computed tomography scan brain tissues: Comparative analysis. *International Journal of Imaging Systems and Technology*.
- [18] Kaur, P., & Chaira, T. (2020). A novel fuzzy approach for segmenting medical images. *Soft Computing*, 1-11.
- [19] Marwa Fradi, Wajih Elhadj Youssef, Ghaith Bouallegue, Mohsen Machhout and Philippe Lasaygues "Automatic USCT Image Processing Segmentation for Osteoporosis Detection" Springer Nature Switzerland AG 2020M. S. Bouhleb and S. Rovetta (Eds.): SETIT 2018, SIST 146, pp. 372–381, 2020.https://doi.org/10.1007/978-3-030-21005-2_36
- [20] Fradi, M., Youssef, W. E., Lasaygues, P., & Machhout, M. (2018). Improved USCT of paired bones using wavelet-based image processing. *International Journal of Image, Graphics and Signal Processing*, 10(9), 1
- [21] Cai, H., Zhu, L., & Han, S. (2018). Proxyllessnas: Direct neural architecture search on target task and hardware. arXiv preprint arXiv:1812.00332
- [22] Real, E., Aggarwal, A., Huang, Y., and Le, Q. V. Regularized evolution for image classifier architecture search. *AAAI*, 2019
- [23] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861
- [24] SANDLER, Mark, HOWARD, Andrew, ZHU, Menglong, et al. Mobilenetv2: Inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018. p. 4510-4520
- [25] SLOSS, Andrew N. et GUSTAFSON, Steven. 2019 Evolutionary Algorithms Review. arXiv preprint arXiv:1906.08870, 2019.
- [26] B.Zoph,V.Vasudevan,J.Shlens,andQ.V.Le. Learning transferable architectures for scalable image recognition. In *CVPR*, 2018.
- [27] Nguyen, L. D., Lin, D., Lin, Z., & Cao, J. (2018). Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation. *2018 IEEE International Symposium*
- [28] Bouallegue, K. (2017). A new class of neural networks and its applications. *Neurocomputing*, 249, 28-47.