Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Enhancing X-Ray Image Classification: A Rank-Based Two-Stage Semi-Supervised Deep Learning Model

Pawan Kumar Mall[1], P K Singh[2]

[1, 2] Computer Science and Engineering Department, Madan Mohan Malaviya University of Technology Gorakhpur, India
Email: pawankumar.mall@gmail.com[1], Topksingh@gmail.com[2]
Corresponding Author: Pawan Kumar Mall (pawankumar.mall@gmail.com)

Deep neural networks (DNN) effectiveness is contingent upon access to quality-labelled training datasets since label mistakes (label noise) in training datasets may significantly impair the accuracy of models trained on clean test data. The primary impediments to developing and using DNN models in the healthcare sector include the lack of sufficient label data. Labeling data by a domain expert is a costly and time-consuming task. To overcome this limitation, the proposed Two-Stage Rank-based Semi-supervised deep learning (TSR-SDL) for Shoulder X-Ray Classification uses the small labelled dataset to generate a labelled dataset from unable dataset to obtain performance equivalent to approaches trained on the enormous dataset. The motivation behind the suggested model TSR-SDL approach is analogous to how physicians deal with unknown or suspicious patients in everyday life. Practitioners handle these questionable circumstances with the support of professional colleagues. Before initiating treatment, some patients consult with a range of skilled doctors. Patients are treated according to the most suitable professional diagnosis (vote count). In this article we have proposed a new ensemble learning technique called "Rank based Ensemble Selection with machine learning models" (TSR-SDL) approach. In this technique, multiple machine learning models are trained on a labeled dataset, and their accuracy is ranked. A dynamic ensemble voting approach is then used to tag samples for each base model in the ensemble. The combination of these tags is used to generate a final tag for an unlabeled dataset. Our suggested TSR-SDL model has attained the best accuracy and specificity, sensitivity, precision, Matthew's correlation coefficient, false discovery rate, false positive rate, f1 score, negative predictive value, and false negative rate negative 92.776%, 97.376%, 86.932%, 96.192%, 85.644%, 3.808%, 2.624%, 91.072%, 90.85%, and 13.068% for unseen dataset respectively. This approach has the potential to improve the performance of ensemble models by leveraging the strengths of multiple base models and selecting the most informative samples for each model. This study results in an improved Semi-supervised deep learning model that is more effective and precise.

Keywords: Co-teacher, Deep Learning, MentorNet, Self-assessment, Student-teacher, X-Ray.

1. Introduction

Semi-supervised deep learning (SSDL) has been highlighted as a potential new study track in the area of computer vision in the present era. SSDL was coined in the 1970s[1],[2] ,[3]. This method is applied to generate labelled training data from an available significant number of unlabeled data. Data labelling is a process of annotating or tagging data with relevant information that helps machine learning algorithms to learn from the data. The process involves manually adding labels or tags to the data by subject matter experts, who have expertise in the domain. Data labelling can be a time-consuming and expensive process, especially for large datasets. Subject matter experts may need to spend a significant amount of time analyzing the data and adding the relevant tags. Additionally, the cost of hiring subject matter experts can be high, as they typically have specialized knowledge and skills. However, accurate data labelling is essential for building high-quality machine learning models. The quality of the data labels directly impacts the performance of machine learning algorithms. Therefore, while data labelling can be a time-consuming and expensive process, it is a necessary investment for organizations that want to build effective machine learning models. SSDL techniques are better applicable to real-world problems where huge amounts of data are readily accessible. At the same time, labelled instances are often difficult to tag, expensive to collect, and time-consuming to process. SSDL is excellent at developing well-known classifiers that compensate for the shortage of tag data. In general, SSDL models are trained on large amounts of unlabelled data using unsupervised learning techniques, such as context tag or class label. The resulting pre-trained model can then be fine-tuned on a smaller amount of labelled data for specific tasks, such as image classification or object detection. When fine-tuning the pre-trained model for a specific task, the model is typically trained on a dataset that includes examples from all of the classes that it needs to recognize.

One well-known study that uses SSDL to reduce the need for annotated data is the work by Doersch et al. (2015), titled "Unsupervised Visual Representation Learning by Context Prediction". In this study, the authors propose a self-supervised learning method for training deep neural networks on large amounts of unlabeled data. The method involves training a neural network to predict the spatial arrangement of patches within an image. This is done by randomly selecting two patches from an image and training the network to predict the relative spatial relationship between the two patches, such as whether one patch is above or below the other. The network is trained on a large dataset of unlabeled images, allowing it to learn to extract useful visual features from the images without requiring manual annotations. The pre-trained network can then be fine-tuned on a smaller amount of labeled data for specific tasks, such as object recognition or scene classification. This approach has been shown to be effective in reducing the need for annotated data, while still achieving high accuracy on a range of image classification tasks. One application of this approach is in the development of deep learning models for medical image analysis. Medical image datasets are often small and expensive to annotate, making it challenging to train accurate deep learning models. However, by pre-training a neural network using SSDL on large amounts of unlabeled medical images, it is possible to reduce the need for annotated data and improve the accuracy of models for tasks such as tumor detection or disease classification. The SSDL models provides the pathway for well trained and strong classification models. However, using this method, incorrectly classified data might reduce the performance of the classification models. This may outcome in a considerable reduction in the performance of classification models. The SSDL models help to solve the requirement for labelled data in the pursuit of a more data-efficient deep learning strategy. Although Pseudo-Labeling is a native method, it gives us a great chance to comprehend SSDL's models to tag unlabeled dataset problems and lays the groundwork for improving the performance of the models. Fig 1 illustrates the SSDL framework.

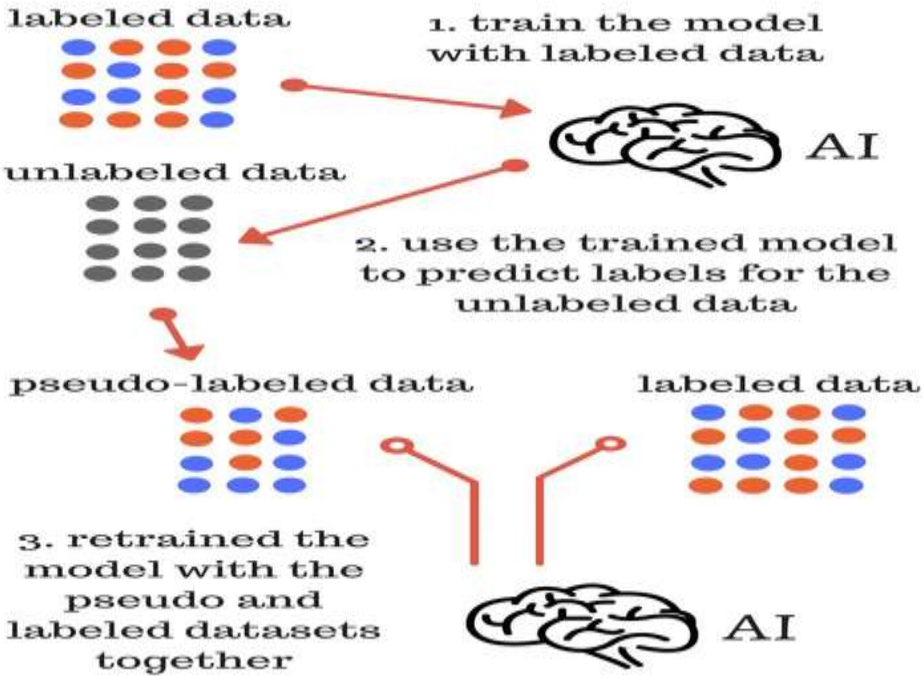

labeled data
1. train the model with labeled data
AI
unlabeled data
2. use the trained model to predict labels for the unlabeled data
pseudo-labeled data
labeled data
3. retrained the model with the pseudo and labeled datasets together
AI

Fig. 1 Illustrating semi-supervised deep learning

In this article, we propose a novel Two-Stage Rank-based Semi-supervised deep learning Model (TSR-SDL). The proposed (TSR-SDL) model is designed to enhance the performance of SSDL models. The deep learning models involved in research are trained on the benchmark MURA-SH dataset [4] collection of shoulder bone X-ray images. We established an unlabelled dataset including 1279 shoulder X-rays collected from the Department of Radiology State Government hospitals. All of these images are annotated by experienced Orthopedic surgeons. In the research, 598 unlabelled images are employed to generate a pseudo label, and 681 images are labelled to evaluate our model. The research studies show that the TSR-SDL model achieves adequate classification results and outperforms the traditional models by a significant margin. The paper is organized as follows. In the second section, the most appropriate related works are discussed. The standard deep learning models and benchmark dataset used for the proposed TSR-SDL model are described in the third section. The proposed work is discussed in the fourth section, simulation, and results of the proposed model are shown in the fifth section; and finally, we draw some conclusions and discuss the potential scope in the last section.

The higlight of this research work are listed below:

- We propose a method for Semi-supervised deep learning Model for Shoulder X-Ray Classification.
- The local data set collected from HATA CHC.
- The key component of our proposed model is to determine the rank of the benchmark DNN, retrain the models using both label and pseudo dataset, and an unseen HATA-SH dataset is used to verify the performance of the purposed model.

- The proposed model achieved accuracy, Specificity, sensitivity, precision, Matthew's correlation coefficient, false discovery rate, false positive rate, f1 score, negative predictive value, and false negative rate achieved 92.776%, 97.376%, 86.932%, 96.192%, 85.644%, 3.808%, 2.624%, 91.072%, 90.85%, 13.068% respectively.
- We can conclude that the model trained with (MURA) dataset from Stanford university was not enough to predict the local dataset. This proposed semi-supervised learning approach solve the issue of unlabelled dataset and also improves the performance of the model.

2. Related Work

There are numerous interesting and vital strategies for semi-supervised learning. This field is well-established and encompasses a diverse range of techniques, including Self-Training, Consistency Regularization, and Hybrid Methods.

Semi-supervised learning is a type of machine learning where a model is trained on both labelled and unlabelled data. The goal is to leverage the large amount of unlabelled data to improve the model's performance on the labelled data. The two major modules of "Student" and "Teacher" in semi-supervised learning can be described as follows:

Student Module: The Student Module in semi-supervised learning refers to the learning algorithm that uses both labelled and unlabelled data to improve its performance. The student algorithm is responsible for iteratively updating its parameters based on the labelled and unlabelled data it receives. In semi-supervised learning, the student algorithm typically consists of two components:

Supervised Component: The supervised component is responsible for training the model using the labelled data. This component is similar to the one used in supervised learning and is responsible for updating the parameters of the model using the labelled data.

Unsupervised Component: The unsupervised component is responsible for leveraging the unlabelled data to improve the model's performance. This component typically involves clustering or generative models to extract features from the unlabelled data and use them to improve the model's performance on the labelled data.

Teacher Module:

The Teacher Module in semi-supervised learning refers to the component that provides guidance to the student algorithm. The teacher algorithm is responsible for selecting the most informative examples from the unlabelled data to provide to the student algorithm. In semi-supervised learning, the teacher algorithm typically consists of two components:

Label Propagation: The label propagation component is responsible for propagating the labels from the labelled data to the unlabelled data. This component uses the similarities between the labelled and unlabelled data to infer the labels of the unlabelled data points. This is done by leveraging the similarities between the labelled and unlabelled data points. There are different techniques that can be used for label propagation, including graph-based methods, such as Laplacian regularization and random walk, and diffusion-based methods, such as label spreading and label propagation.

Active Learning: The active learning component is responsible for selecting the most informative examples from the unlabelled data to provide to the student algorithm. This component uses uncertainty sampling or other heuristics to select the examples that are most likely to improve the model's performance on the labelled data. The goal is to maximize the information gained from each new labelled example while minimizing the number of labelled examples needed. There are different strategies that can be used for active learning, including uncertainty sampling, query-by-committee, and density-based sampling.

SSDL (Self-Supervised Deep Learning) is a technique that uses unsupervised learning to train deep learning models on large amounts of unlabeled data, which can reduce the need for annotated data. When combined with traditional deep learning algorithms, SSDL can help to improve the accuracy of models trained on limited labeled data. Traditional deep learning algorithms rely on a large amount of labeled data to train models for classification or object detection tasks. However, labeling large amounts of data can be time-consuming and expensive. SSDL overcomes this challenge by using unsupervised learning techniques to pre-train a model on unlabeled data. The pre-trained model can then be fine-tuned on a smaller amount of labeled data to improve its accuracy for a specific task. One approach for combining SSDL with deep learning algorithms is called "semi-supervised learning". In this approach, the pre-trained model is fine-tuned using a small amount of labeled data for a specific task, while leveraging the knowledge learned from the large amount of unlabeled data. This can significantly reduce the need for annotated data, while still achieving high accuracy on the task at hand. Another approach is called "transfer learning". In this approach, a pre-trained SSDL model is used as a starting point for training a model on a new task. The pre-trained model is first fine-tuned on a large amount of labeled data for a similar task, such as image classification or object detection. The fine-tuned model can then be further trained on a smaller amount of labeled data for the specific task of interest. This approach can also reduce the need for annotated data, while still achieving high accuracy on the target task.

The Self-Training paradigm is based on the notion of making model predictions on an unlabelled image. The model employs both pseudo-labels and ground truth labels simultaneously. The SSDL Pseudo-label [5] approach is a simple and efficient solution known as "Pseudo-label" that was introduced in the year 2013. In [6], authors have introduced "Noisy Student," a semi-supervised approach also known as Knowledge Distillation. The essential concept is to train two distinct modules termed “Student” and “Teacher." In this approach, the labelled images are employed to train the teacher module, while the unlabelled images are inferred using pseudo-labels. The aggregated unlabelled and labelled dataset is used to train student modules. After a student module has been trained, it takes over as the new teacher model and repeats the same process three times. The student model incorporates noise such as stochastic depth and dropout.

The core concept behind consistency regularization is that the SSDL model tags an unlabelled dataset should stay consistent even if noise (Gaussian noise and image augmentation) is introduced. The model should produce consistent results for given input and its realistically perturbed versions. We humans are highly resistant to little changes. For example, introducing modest amounts of noise (e.g., altering pixel values) to an image is unnoticeable to us. A deep learning model should be resistant to such disturbances. This is often accomplished by reducing the difference between the original input prediction and the perturbed version of that input [7] [8] [9] [10] [11]. The π model [12] uses the network outcomes as consistency. The main concept is to generate two random augmentations of the given input images for both unlabelled and labelled data [13], [14]. The dropout method is introduced to tag the class label of both augmented images. The consistency loss is calculated as a square difference of two predictions. The overall loss is calculated as the weighted sum of two-loss components. The two main modules of the mean teacher [15] approach are "Student" and "Teacher." The student module is a standard framework with dropout, while the teacher module is a duplicate of the student module. The only difference is weights assign an exponential moving average according to the weights of the student module. Both the labelled and unlabelled images are generated into two random augmented versions. The student module is used to tag class label distribution for the first augmented image and the teacher module is used to tag the class label distribution for the second image [16], [17], [18]. The consistency loss is measured by taking the square difference between two predictions. The cross-entropy loss is calculated on labelled images. The final loss is computed by summing the weighted sum of the two-loss components. Virtual Adversarial Training [19] model core idea is to generate adversarial transformation two image view of input labelled and unlabelled images. The same SSDL model is utilized to tag class label distributions for both images view. The consistency loss is evaluated using the KL-divergence method for both the view predictions. The cross-entropy loss is calculated on labelled images. The total loss is computed by summing the weighted sum of the two losses.

The Hybrid Process is a problem-solving methodology that combines the strengths of both deductive and inductive reasoning to arrive at a solution. The central concept underpinning the Hybrid Process is the idea that both types of reasoning are necessary to achieve a comprehensive understanding of a problem and develop a viable solution. Deductive reasoning starts with a general principle or theory and applies it to specific situations to arrive at a conclusion. It is a top-down approach that relies on logical reasoning and known facts to arrive at a specific answer. Deductive reasoning is useful when the problem is well-defined and there is a clear set of rules or principles to follow. Inductive reasoning, on the other hand, starts with specific observations or data and uses them to form a general theory or hypothesis. It is a bottom-up approach that relies on empirical evidence to arrive at a conclusion. Inductive reasoning is useful when the problem is complex and requires a deep understanding of the data to arrive at a solution. The Hybrid Process combines these two types of reasoning to create a more robust problem-solving methodology. It starts with deductive reasoning to establish a general understanding of the problem, identify key variables, and formulate a hypothesis. It then uses inductive reasoning to collect and analyze data, refine the hypothesis, and arrive at a solution. The Hybrid Process emphasizes the iterative nature of problem-solving, with each cycle of deductive and inductive reasoning refining the hypothesis and bringing the solution closer to reality. It also acknowledges the importance of creativity and intuition in the problem-solving process, allowing for a more flexible and adaptive approach to finding a solution [21], [22], [23], [24]..

In [25], the author have introduce a crow swarm optimization approach for COVID-19 diagnosis, this paper suggests an integrated method for choosing the best deep learning model. Utilizing a fitness function created for assessing the performance of the deep learning models, the crow swarm optimization method is used to identify the ideal set of coefficients. Using chest X-ray pictures from a dataset that contains the most COVID-19 images, in [26] this research assesses the effectiveness of deep learning models for COVID-19 diagnosis. In [27], It has been demonstrated that the suggested approach for early detection and categorization of COVID-19 utilizing image processing of X-ray pictures is practical in that it offers an end-to-end framework without the requirement for manual feature extraction and manual selection procedures. In [28], The most impactful features from the segmented photos that can aid in the identification of COVID-19 were extracted using the Visual Geometry Group Network, convolutional deep belief network, and high-resolution network. In [29], author have introduce a novel multi-agent deep reinforcement learning (DRL)-based mask extraction approach to reduce long-term manual mask extraction and improve medical picture segmentation frameworks. To address mask extraction concerns, a DRL-based technique is presented.

3. Materials and Models

3.1. X-ray Dataset:

In terms of the differences between X-rays and MRI scans for getting arm pictures, there are several key factors to consider:

- Radiation exposure: X-rays involve exposure to ionizing radiation, which can be harmful if a person is exposed to too much radiation over time. MRI scans, in contrast, do not use ionizing radiation and are considered safe for most people.
- Image detail: X-rays provide detailed images of bones and other hard tissues, while MRI scans provide detailed images of soft tissues. Depending on the suspected condition, one or the other may be more appropriate.
- Time: X-rays are generally quicker and easier to perform than MRI scans, which can take up to an hour or more to complete.
- Cost: X-rays are generally less expensive than MRI scans, although the cost can vary depending on the type of X-ray or MRI being performed and the location where it is done.

In summary, while both X-rays and MRI scans can be used to diagnose medical conditions affecting the arm, they differ in terms of the type of information they provide, the amount of radiation exposure involved, the time and cost required, and other factors. The choice of which imaging technique to use will depend on the specific condition being investigated and the preferences of the healthcare provider and patient. The musculoskeletal radiograph dataset is one of the largest collections of bone X-rays. The dataset contains a total of 58817 images from 21456 radiographic case studies, along with reports from January 2014 through December 2017 in 4 years at a children's hospital. The average age of the patients was 7.2 years, and 57 percent of them were male. The (MURA) musculoskeletal radiograph has contained a total of 40561 X-ray data. The dataset is collection of 44.36 % abnormal and 55.63% normal X-rays. This is the most popular X-ray dataset published by [4]. We have considered only the shoulder study from the MURA dataset, and the new dataset is renamed MURA-SH for our experiment. The MURA-SH dataset is prearranged into two groups train set and test set. The HATA-SH dataset consists of X-rays images of shoulder bones. The dataset contains 1279 images from different radiographic case studies and reports from January 2018 through December 2020 in 2 years at State Government hospital, Hata, Kushinagar, Uttar Pradesh. The MURA-SH X-ray and HATA-SH dataset details are listed in Table 1.

Table 1 — The MURA-SH and HATA-SH details

Dataset	**Train Set**	**Test Set**
MURA-SH	8942(Normal Abnormal)	194(Normal Abnormal)
HATA-SH (unlabelled)	598	0
HATA-SH (unseen)	0	681(Normal-381, Abnormal-300)
Complete Dataset Size involved in Experiment:10415		

3.2. Deep learning standard Models

Adjusting the design of a neural network architecture based on the layers assist enhance the network's overall performance. Using bulk normalization and ReLU activation functions before convolution layers, for example, can provide various advantages. Bulk normalization is a technique for normalizing layer inputs so that the mean activation is near to zero and the standard deviation is close to one. This helps to prevent internal covariate shift and increase network stability. The inputs to the convolution layers are adjusted by applying bulk normalization before convolution layers, which can assist enhance the network's convergence rate and overall performance. Overall, utilizing bulk normalization and the ReLU activation function before convolution layers can assist to increase the network's stability and convergence rate, as well as extract more complicated features from the input. The significant technical aspects of standard DNN are briefly described below:

3.2.1. MobileNet:

MobileNet is a more effective and lightweight framework [30]. MobileNet is a popular neural network architecture for mobile and embedded devices that is designed to be lightweight and efficient, while still achieving high accuracy on image classification tasks. One of the key design principles of MobileNet is the use of depth-wise separable convolutions, which allows for a significant reduction in the number of parameters and computations required compared to traditional convolutions. Depth-wise separable convolutions break down a convolutional layer into two separate operations: depth-wise convolutions and point-wise convolutions. Depth-wise convolutions apply a separate filter to each channel of the input, while point-wise convolutions combine the outputs of the depth-wise convolutions using a 1x1 convolution. This approach drastically reduces the number of parameters and computations required for a convolutional layer, making it much more efficient. In MobileNet, Depth-wise separable convolutions are used throughout the network architecture, including in the bottleneck layers that form the core of the network. A bottleneck layer consists of a depth-wise convolution followed by a point-wise convolution, with the point-wise convolution being used to increase the number of output channels. The use of depth-wise separable convolutions allows MobileNet to achieve a high level of accuracy on image classification tasks while using significantly fewer parameters and computations than traditional convolutional neural networks. This makes it well-suited for use in mobile and embedded devices, where computational resources are limited. The block layout of MobileNet is depicted in Fig 2.

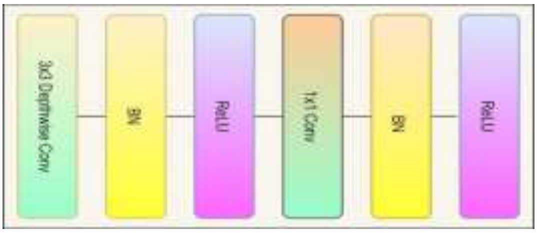


Fig. 2 —Illustrating block layout of MobileNet

3.2.2. Pre-Act ResNet:

The pre-activation Resnet model [31] is a variation of the Resnet model. The changes in architecture are based on layers, e.g., batch norm and relu before convolution. The batch normalization at each layer to reduce the internal covariate shift greatly improves the learning efficiency of the networks. The key benefit of employing the ReLU function over other activation functions is that it does not stimulate all neurons at once. Earlier, in the Resnet version, when the layers increase from 101 layers to 1202 layers, the error rate has increased from 6.43 percent to 7.93 percent. The block layout of the Pre-Act Resnet is depicted in Fig 3.

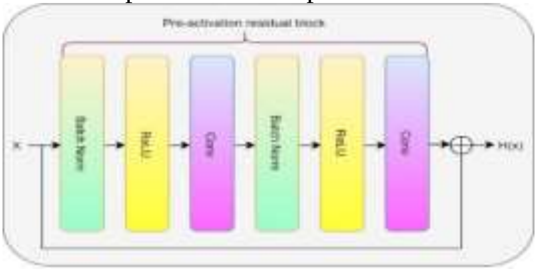


Fig. 3 —Illustrating block layout of Pre-Act ResNet

3.2.3. ResNet18:

The ResNet-18[32] is a residual deep learning model that is 18 layers deep. ResNet18 is a type of convolutional neural network (CNN) that is widely used in computer vision tasks such as image classification, object detection, and segmentation. One of the key features of ResNet18 is the use of skip connections or skip links, which help to alleviate the problem of vanishing gradients during training. Skip connections are essentially shortcuts that bypass one or more layers in a neural network. In ResNet18, the skip connections are added between adjacent residual blocks. A residual block is a basic building block of ResNet, which consists of a series of convolutional layers and nonlinear activation functions. The skip connection in ResNet18 allows the model to learn residual mappings, which are the differences between the input and output of a residual block. This is done by adding the input of a residual block to its output, which effectively creates a shortcut between the input and output. The skip connections in ResNet18 help to overcome the problem of vanishing gradients, which is a common issue in deep neural networks. When training deep neural networks, the gradients can become very small as they propagate through the network, which can slow down or even prevent learning. By adding skip connections, ResNet18 allows the gradients to bypass some of the layers, which helps to prevent them from vanishing. The block layout of ResNet-18 is depicted in Fig

4 [43].

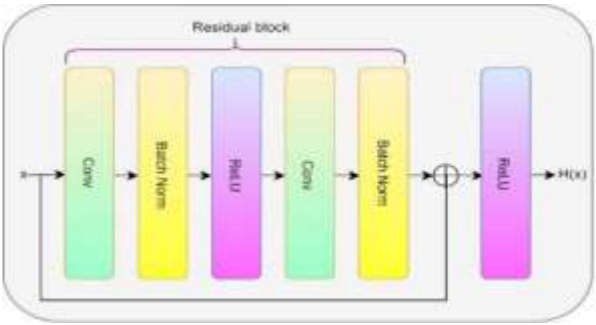


Fig. 4 —Illustrating block layout of ResNet-18

3.2.4. VGG-16:

VGG-16 [33] 16-layer deep model was the most successful architecture in ImageNet competition (ILSVRC challenge) 2014. According to the research findings, network depth is a critical component for increased performance. The block layout of VGG-16 is depicted in Fig 5.

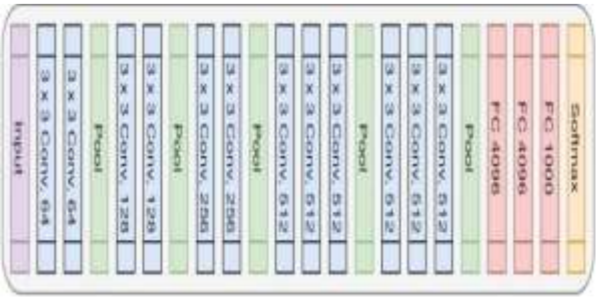


Fig. 5 —Illustrating block layout of VGG-16

3.2.5. VGG-19:

The VGG19 model [34] is a variation of the VGG model is consists19 layers. The ImageNet competition (ILSVRC challenge) 2014 consists of 1,000 different classes, the train, validation, and test dataset 1.2 million images,50 thousand images, and 150 thousand images, respectively. The model has learned extensive classification characteristics for a broad variety of data [35]. In VGG model "16" and "19" represent the number of weight layers in the network. VGG19 only has three more conv3 layers. The block layout of VGG-19 is depicted in Fig 6 [44][45].

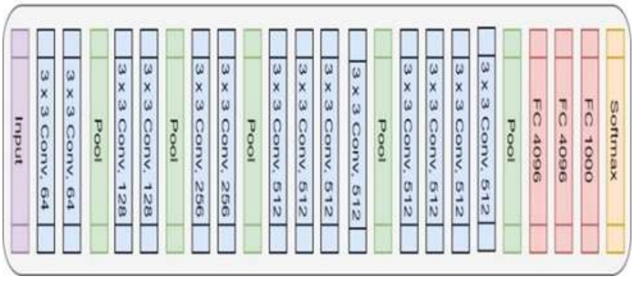


Fig. 6 —Illustrating block layout of VGG-19

4. Proposed Work

The diagnosis is generated with expensive equipment in the medical domain, and labels are derived from a time-consuming process of multiple health experts. In a variety of methods, semi-supervised deep learning models can compensate for a lack of labelled training data. Traditional supervised deep learning requires a huge quantity of labelled data to train the model to spot patterns and make predictions. Yet, getting labelled data can be costly or time-consuming in many real-world circumstances. This is where semi-supervised learning may help. During training, semi-supervised learning blends labelled and unlabeled data. The model can better generalize and predict on new, unknown data by exploiting the knowledge included within both the labelled and unlabeled data. In the context of compensating for a shortage of tag data, a semi-supervised deep learning model can learn certain broad patterns or features using the limited quantity of labelled data available, and then utilize the huge amount of unlabeled data to deepen its knowledge of these patterns. This improves the model's ability to recognize and categories data points that have not been explicitly labelled. Overall, semi-supervised deep learning models can be an effective technique for compensating for a lack of tag data, as well as for improving the accuracy and efficiency of machine learning algorithms in a range of applications.

The proposed model has six major phases: Image pre-processing, rank determination, model generation, generating pseudo dataset, retraining DNN, and evaluation. The key component of our proposed model is to determine the rank of the benchmark DNN, retrain the models using both label and pseudo dataset, and an unseen HATA-SH dataset is used to verify the performance of the purposed model. Fig 7 illustrates the workflow of the proposed model.

The central idea behind generating two random augmentations of raw pictures for both unlabeled and labeled data is to improve the performance of machine learning models in image recognition tasks. Augmentation refers to the process of modifying the original images in some way to create new images that still contain the same essential information. By generating multiple random augmentations of the raw pictures, the model is exposed to a wider range of variations in the data, which can help it learn more robust and generalizable features [46].

For unlabeled data, generating random augmentations can be used as a form of pre-processing to increase the size of the dataset and reduce the risk of overfitting. By creating multiple variations of each image, the model is forced to learn to recognize the underlying patterns that are common across all the variations, rather than relying on specific details of any one image.

For labeled data, generating random augmentations can be used as a form of data augmentation to improve the accuracy of the model. By training the model on multiple variations of each labeled image, the model is exposed to a wider range of variations in the data, which can help it learn more robust and generalizable features. This can lead to improved performance on new, unseen data.

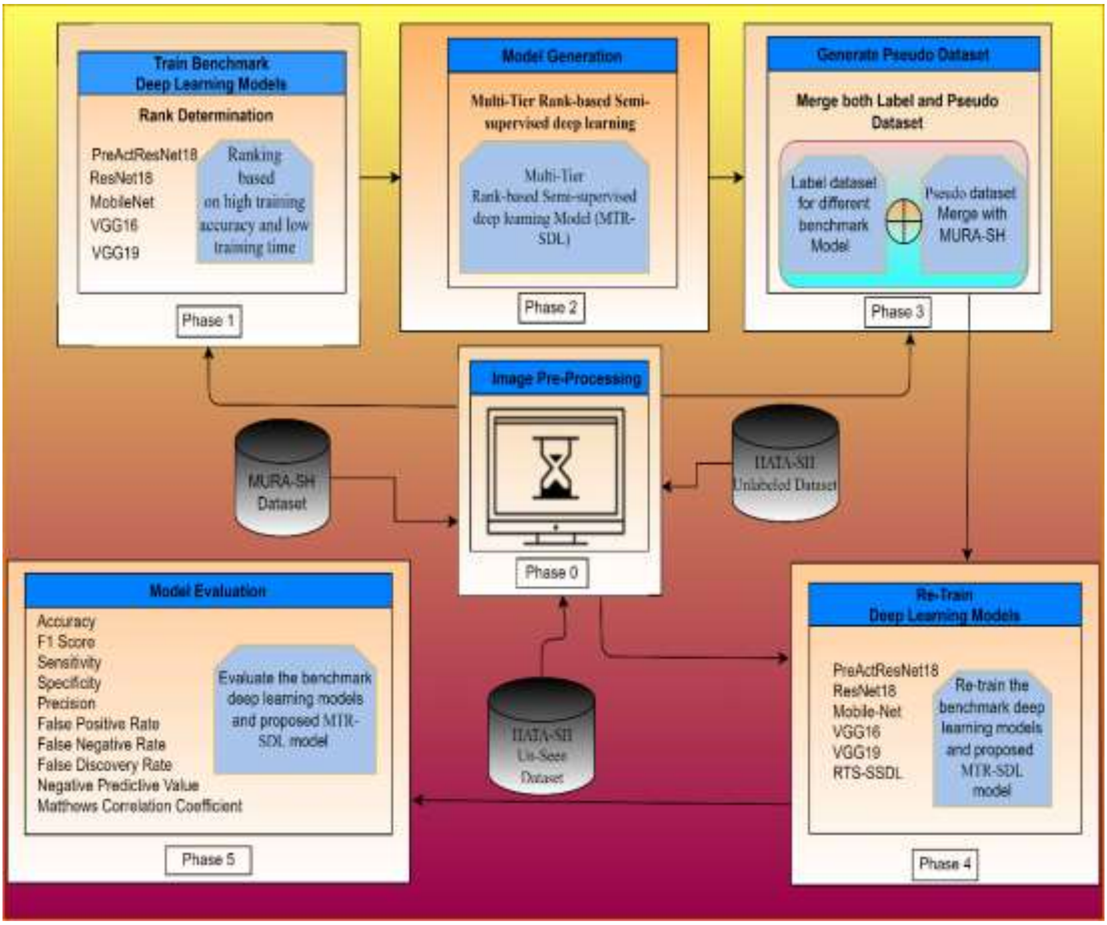

Train Benchmark
Deep Learning Models
Rank Determination
PreActResNet18
ResNet18
MobileNet
VGG16
VGG19
Ranking based on high training accuracy and low training time
Phase 1
Model Generation
Multi-Tier Rank-based Semi-supervised deep learning
Multi-Tier Rank-based Semi-supervised deep learning Model (MTR-SDL)
Phase 2
Generate Pseudo Dataset
Merge both Label and Pseudo Dataset
Label dataset for different benchmark Model
Pseudo dataset Merge with MURA-SH
Phase 3
Image Pre-Processing
Phase 0
MURA-SH Dataset
HATA-SH Unlabeled Dataset
HATA-SH Un-Seen Dataset
Model Evaluation
Accuracy
F1 Score
Sensitivity
Specificity
Precision
False Positive Rate
False Negative Rate
False Discovery Rate
Negative Predictive Value
Matthews Correlation Coefficient
Evaluate the benchmark deep learning models and proposed MTR-SDL model
Phase 5
Re-Train
Deep Learning Models
PreActResNet18
ResNet18
Mobile-Net
VGG16
VGG19
RTS-SSDL
Re-train the benchmark deep learning models and proposed MTR-SDL model
Phase 4

Fig. 7 — illustrating the workflow of the proposed model

4.1. Research environment

The experiment performed in the virtual environment. The virtual machine is loaded with Ubuntu operating system (OS), eight virtual CPUs and 14GB RAM. The proposed framework is implemented using Python 3.0.

4.2. Image Pre-processing

The pre-processing phase improves the significant information of the input image—the train and test data dataset. The SSDL model is trained using training data and then evaluated on other testing data to determine how well it performs. This phase transformation of all input datasets involved in the experiment. In this phase, all datasets are resized to 64×64-pixel images. The Normalized technique provides an edge on both label and unable datasets. This technique helps out to reduce distortion and noise. The function of Gaussian noise is to introduce random perturbations in the pixel values of an image. This can be used as a form of data augmentation to improve the robustness of machine learning models to variations and distortions in the input data. When applied to an image, Gaussian noise adds random variations to the pixel values, which can simulate effects such as image blurring, pixelation, and noise that can occur due to camera or sensor limitations, compression artifacts, or other factors. In the context of consistency regularization for semi-supervised learning (SSDL), Gaussian noise is used to perturb the input images in an unlabelled dataset during training. The aim is to encourage the model to learn features that are robust to small variations and distortions in the input data. By adding noise to the input images and enforcing consistency between the predictions of the model on the original and perturbed images, the model can learn to generalize better to new and unseen data [47].

4.3. Rank Determination

The training and validation of the DNN (MobileNet, Pre-Act Resnet18, ResNet18, VGG16, VGG19) have been implemented from scratch. The training, validation, and testing are performed on the X-ray image (MURA-SH 64×64) dataset. This phase will support us figure out the rank from the different DNN. The essential phase of this work is to determine the ranks of deep learning models. In this section, elaborate on our methodology in further detail. Although there are various techniques for ranking, each has its own merits and demerits. The main demerit of pointwise, listwise, and pairwise ranking techniques is that they necessitate a significant amount of effort from domain experts to establish precise ranks. Our proposed ranking technique focuses on parameters such as training accuracy, test accuracy, and training time elapsed of different DNN. The ranking of the model is determined using a threshold-based ranking algorithm 1 [48].

Algorithm 1: Procedure for the threshold-based ranking among the standard deep learning models:

```
INPUT: Train_acc , Test_acc , Elsp_train_time , RankTrain_acc , RankTest_acc , RankElsp_train_time, N
OUTPUT: Rank[ ]
Initialization;
Train_acc = Train accuracy,
Test_acc= Test accuracy,
Elsp_train_time = Training time elapsed,
RankTrain_acc = Rank Train accuracy,
RankTest_acc = Rank Test accuracy
$\mathrm{Rank}_{\mathrm{Elsp\_Traintime}}$ = Rank Training time elapsed
N = number of deep learning model
Rank[ ]= Rank assign to deep learning model
$\alpha = \sum_{i=1}^{N} \mathrm{Train}_{\mathrm{Acc}}/N$          /* Compute train threshold*/
$\beta = \sum_{i=1}^{N} \mathrm{Test}_{\mathrm{Acc}}/N$          /* Compute test threshold*/
$\gamma = \sum_{i=1}^{N} \mathrm{Elsp}_{\mathrm{Traintime}}/N$   /* Compute elapsed threshold*/
For i=1 to N do                              /* Compute rank for N deep learning model*/
    If $\mathrm{Elsp}_{\mathrm{Traintime}} < \gamma$            /* Compare Elasp train time with threshold*/
        $\mathrm{Rank[i]} = \mathrm{Rank}_{\mathrm{Elsp\_Traintime}}[\mathrm{i}]$ /* Assign elapsed train time rank to model rank*/
    If $\mathrm{Test}_{\mathrm{Acc}} > \beta$              /* Compare test accuracy with test threshold */
        $\mathrm{Rank[i]} = \mathrm{Rank}_{\mathrm{test\_Acc}}[\mathrm{i}]$   /* Assign test rank to model rank*/
    If $\mathrm{Train}_{\mathrm{Acc}} > \alpha$             /* Compare train accuracy with train threshold*/
```

```
        $Rank[i] = Rank_{Train\_Acc}[i]$    /* Assign train rank to model rank*/
    Else
        $Rank[i] = Rank_{Elsp\_Traintime}[i]$ /* Assign elapsed train time rank to model rank*/
End For
```

Note: In case of a tie between ranking priority is $Rank_{Elsp\_Traintime}$,$Rank\ Train_{Acc}$,$Rank\ Test_{Acc}$

4.4. Model Generation:

The motivations behind the research work (TSR-SDL) that you mentioned are:

1. Understanding how doctors tackle unseen or suspected cases in real life: The aim of this motivation is to study how doctors approach cases that are not clear-cut or have not been encountered before. This can help in improving the diagnostic process and identifying potential gaps in medical knowledge.

2. Lack of sufficient labeled data: This motivation is related to the challenges of obtaining labeled data, which is essential for training machine learning models. The high cost and time required for labeling data can limit the size and diversity of available datasets, which can affect the performance of machine learning models.

3. Proving the importance of local dataset before using pre-trained models: This motivation highlights the importance of using locally collected data to fine-tune pre-trained models for specific tasks. This can improve the accuracy and generalization of machine learning models in real-world scenarios.

The research work (TSR-SDL) involves experimenting with five standard deep learning models (MobileNet, Pre-Act Resnet18, ResNet18, VGG16, VGG19) to address these motivations. By studying how doctors approach unseen cases and exploring the use of locally collected data, this research aims to improve the accuracy and efficiency of machine learning models in medical diagnosis [42].

The model generation phase is vital part of our proposed work.
The proposed semi-supervised model (TSR-SDL) is depicted in Fig 8. The suggested model is generated using algorithm 2. In the first stage of the algorithm, three fusion classifiers are generated according to the top three ranks. Fusion classifier set contains one of the top three ranks, left out the other two, and contains all other ranked standard deep learning models. The second step of the algorithm is to produce pseudo labels for all three sub-models. In the second stage, we club all three fusion classifiers and generate a master fusion classifier (MFC) model. The next step of the algorithm is to produce pseudo labels from the proposed model [48][49].

Algorithm 2: Procedure for the model TSR-SDL model generation:

```
INPUT: Ud, N, M_Rank [], M
OUTPUT: Pdpm
Initialization;
Ud = Unlabled dataset,
N = Number of Standard Deep learning model,
\
M_Rank[]=Array of Standard Deep learning models sorted according to ascending order,
M = Size of unlable dataset,
fusion_classifier_Rank =Deep learning models
MFC=Master fusion classifier
Pd_MFC  = Pseudo_dataset
GENERATE_PSEUDO_LABEL () method to generate pseudo label
For k=1 to 3 do                              /* Compute Pseudo dataset for three models*/
    Fusion_classifier_Rank[k] ={M_Rank[k], M_Rank [4], …... M_Rank[N]}
    /*Generate Three sub model according to top three Rank */
    Pd_Fusion_classifier←GENERATE_PSEUDO_LABEL(Ud, Fusion_classifier_Rank[k])
End for
MFC [] = {Fusion_classifier_Rank [1], Fusion_classifier_Rank [2], Fusion_classifier_Rank [3]}
Pd_ MFC ← GENERATE_PSEUDO_LABEL (Ud, MFC []}
```

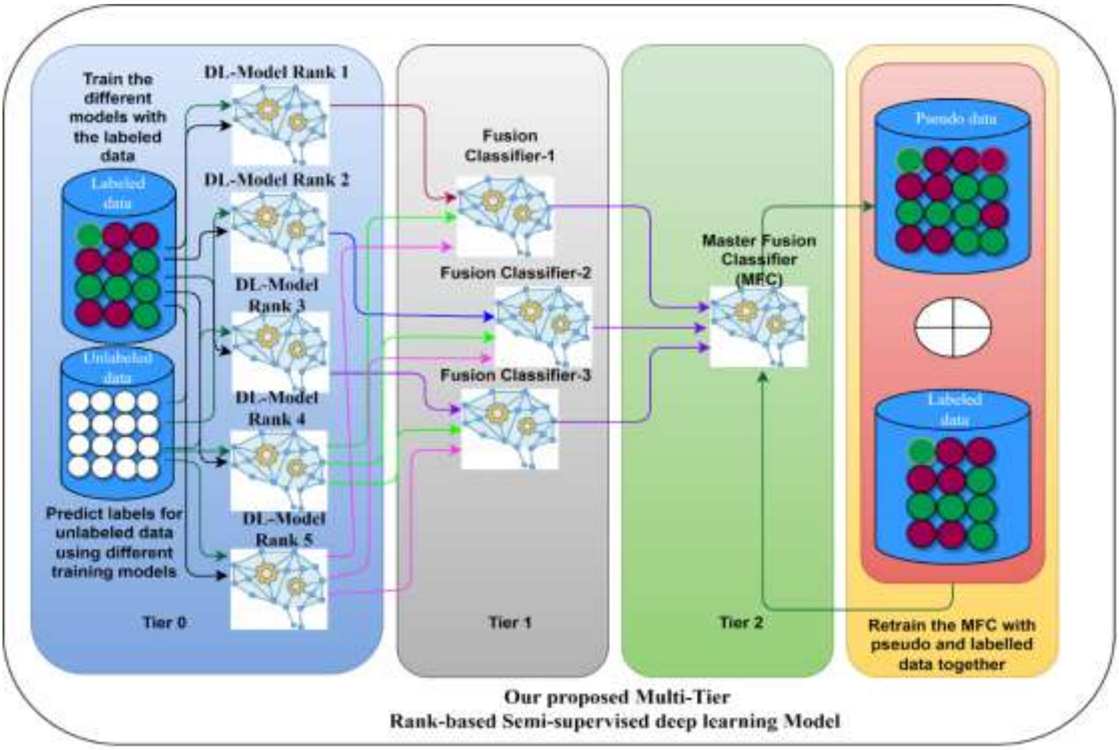

Train the different models with the labeled data
DL-Model Rank 1
DL-Model Rank 2
DL-Model Rank 3
DL-Model Rank 4
DL-Model Rank 5
Labeled data
Unlabeled data
Predict labels for unlabeled data using different training models
Tier 0
Fusion Classifier-1
Fusion Classifier-2
Fusion Classifier-3
Tier 1
Master Fusion Classifier (MFC)
Tier 2
Pseudo data
Labeled data
Retrain the MFC with pseudo and labelled data together
Our proposed Multi-Tier
Rank-based Semi-supervised deep learning Model

Fig. 8 — Proposed Two-Stage Rank-based Semi-supervised deep learning Model model

4.5. Generate Pseudo Dataset:

The objective of this phase is to generate a pseudo dataset for the unlabeled HATA-SH dataset. The pseudo datasets are generated based on the vote count, with the highest vote count label is considered pseudo labels for different standard models and proposed models. Step by step, algorithm 3 is followed to generate the pseudo dataset [50].

Algorithm 3: Procedure for Pseudo label generation for unlabeled dataset:

```
INPUT: Ud, N, M_Rank [], M
OUTPUT: Pdpm
Initialization;
Ud = Unlabled dataset,
N = Number of Standard Deep learning model,
M_Rank[]=Array of Standard Deep learning models sorted according to ascending order,
M = Size of unlable dataset,
Fusion_classifier_Rank =Set of deep learning models
Pdpm = Pseudo dataset,
    GENERATE_PSEUDO_LABEL (Ud, Fusion_classifier_Rank [])
M_size ← Ud
for k=1 to M_size do                /* Predict pseudo label for Unlabelled dataset Ud*/
        For i=1 to N do
            PdN[k][i] = Fusion_classifier_Rank []    /*Predict the Pseudo label according to trained standard deep learning model */
            If PdN[k][i] is Class1          /* Compare test accuracy with test threshold */
            Vote_c1= Vote_c1+1
            Else
            Vote_c2= Vote_c1+1
        End for
          IF Vote_c1> Vote_c2           /* Compare test accuracy with test threshold */
          Pd[k][i] =  Class1            /* Assign test rank to model rank*/
          ELSE IF Vote_c1= Vote_c2
          Pd[k][i] =  Top Rank Class     /* Assign elapsed train time rank to model rank*/
          ELSE
          Pd[k][i] = Class2
End for
```

4.6. Retrain the models using both label and pseudo dataset:

In this phase, the standard deep learning models are retrained with the combined MURA-SH, and pseudo dataset generated by each model and (TSR-SDL) proposed model.

4.7. Validation of proposed model on Unseen HATA-SH dataset:

In the last phase of the experiment, an unseen dataset HATA-SH is used to assess and validate the performance of the (TSR-SDL) proposed model.

5. Simulation, evaluation, and validation

The proposed model is categorized into six different phases Image pre-processing, rank determination, model generation, generating pseudo dataset, retraining deep learning models, and evaluation. Python version 3.0 is used to implement the proposed model. The evaluation and validation are explained in detail in the upcoming subsections.

Model Evaluations: The performance of our model is evaluated using the confusion matrix tool. A confusion matrix is a table that is used to evaluate the performance of a classification model by comparing the predicted class labels to the true class labels. The confusion matrix is a powerful tool for evaluating the performance of a binary classification model, which predicts one of two possible outcomes. It is a table that summarizes the performance of a model by comparing its predictions to the true values of the target variable. The efficacy of the confusion matrix lies in its ability to provide a detailed and comprehensive evaluation of the model's performance. The metrics in the confusion matrix can be used to calculate a variety of other performance measures, including accuracy, precision, recall, F1 score, and ROC curve. These measures can help you identify the strengths and weaknesses of your model and make improvements as needed.

The benefits of using the confusion matrix include:

- Easy to interpret: The confusion matrix provides a simple and intuitive representation of the performance of a binary classification model.

- Provides detailed information: The confusion matrix allows you to see the number of true positives, true negatives, false positives, and false negatives, which can be useful for identifying specific areas of improvement for your model.

- Useful for comparing models: The confusion matrix can be used to compare the performance of multiple models, allowing you to identify the best-performing model for a given problem.

- Useful for adjusting thresholds: The confusion matrix can be used to adjust the threshold for the model's predictions, which can improve its performance on specific metrics.

The matrix contains information about the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions made by the model.

Here are the parameters of a confusion matrix:

- True Positive (TP): This is the number of instances that were correctly predicted as positive by the model.

- True Negative (TN): This is the number of instances that were correctly predicted as negative by the model.

- False Positive (FP): This is the number of instances that were predicted as positive by the model, but were actually negative in reality.

- False Negative (FN): This is the number of instances that were predicted as negative by the model, but were actually positive in reality.

These parameters can be used to calculate various performance metrics for the classification model, such as accuracy, specificity, sensitivity (Recall), precision, matthews correlation coefficient (MCC), false discovery rate (FDR), false positive rate (FPR), f1 score, negative predictive value (NPV), and false negative rate (FNR).

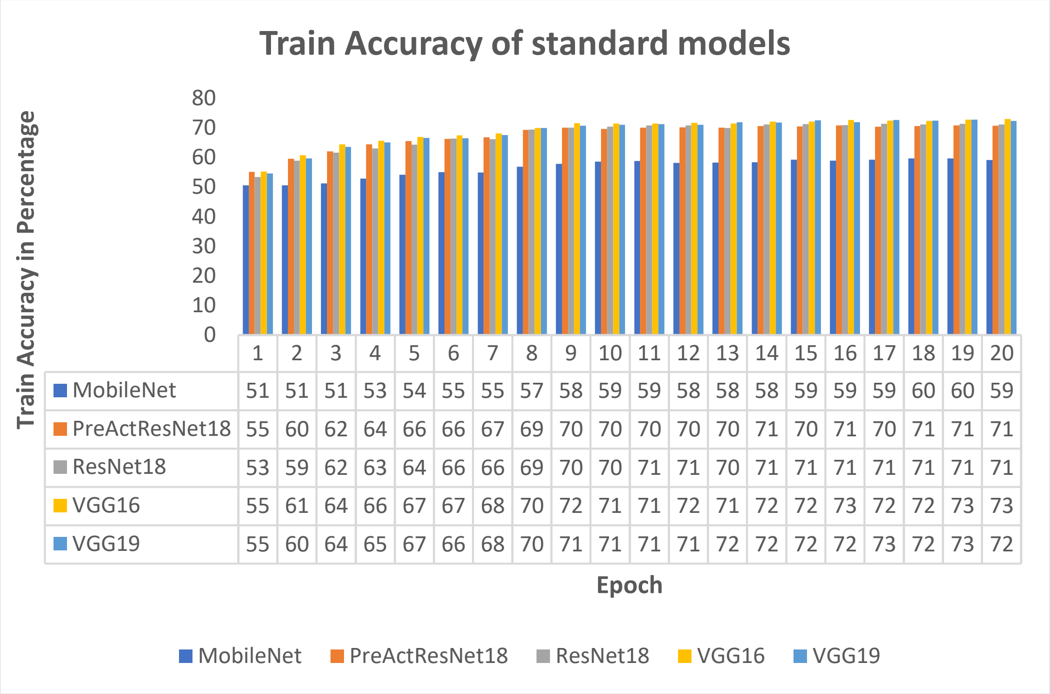
Accuracy = (TP + TN) / (TP + TN + FP + FN) (1)
Precision = TP / (TP + FP) (2)
Recall = TP / (TP + FN) (3)
Specificity = TN / (TN + FP) (4)
F1-score = 2 * (Precision * Recall) / (Precision + Recall) (5)
MCC = (TP * TN - FP * FN) / sqrt((TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)) (6)
NPV = TN / (TN + FN) (7)
FPR = FP / (FP + TN) (8)
FDR = FP / (FP + TP) (9)
FNR = FN / (FN + TP) (10)

5.1. Experiment result for Rank determination:

Our proposed model has been implemented in Python version 3.0, and it provides an opportunity to increase the accuracy of SSDL models. On a standard medical image dataset, we performed a set of experiments to investigate and evaluate the effectiveness of our suggested approach.

The impact of label errors on model performance depends on the severity and frequency of the errors. In some cases, the model may be able to compensate for a small amount of label noise, but in other cases, the errors can have a significant impact on the model's accuracy and generalization ability. One common effect of label errors is that they can lead to overfitting. Overfitting occurs when a model is too complex and learns the noise in the training data instead of the underlying patterns. In the case of label noise, the model may learn to predict the incorrect labels in the training data, resulting in poor performance on new, unseen data. Label errors can also cause underfitting, where the model is not complex enough to capture the underlying patterns in the data. In this case, the model may be too simple to account for the variations in the data caused by the label errors, leading to poor performance on both the training and test data.

The standard models (MobileNet, Pre-Act Resnet18, ResNet18, VGG16, VGG19) are trained and evaluated from scratch. The training accuracy of MobileNet, Pre-Act Resnet18, ResNet18, VGG16, and VGG19 standard models are depicted in Fig 9.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MobileNet	51	51	51	53	54	55	55	57	58	59	59	58	58	58	59	59	59	60	60	59
PreActResNet18	55	60	62	64	66	66	67	69	70	70	70	70	70	71	70	71	70	71	71	71
ResNet18	53	59	62	63	64	66	66	69	70	70	71	71	70	71	71	71	71	71	71	71
VGG16	55	61	64	66	67	67	68	70	72	71	71	72	71	72	72	73	72	72	73	73
VGG19	55	60	64	65	67	66	68	70	71	71	71	71	72	72	72	72	73	72	73	72

Fig. 9 — Train Accuracy of standard deep learning models

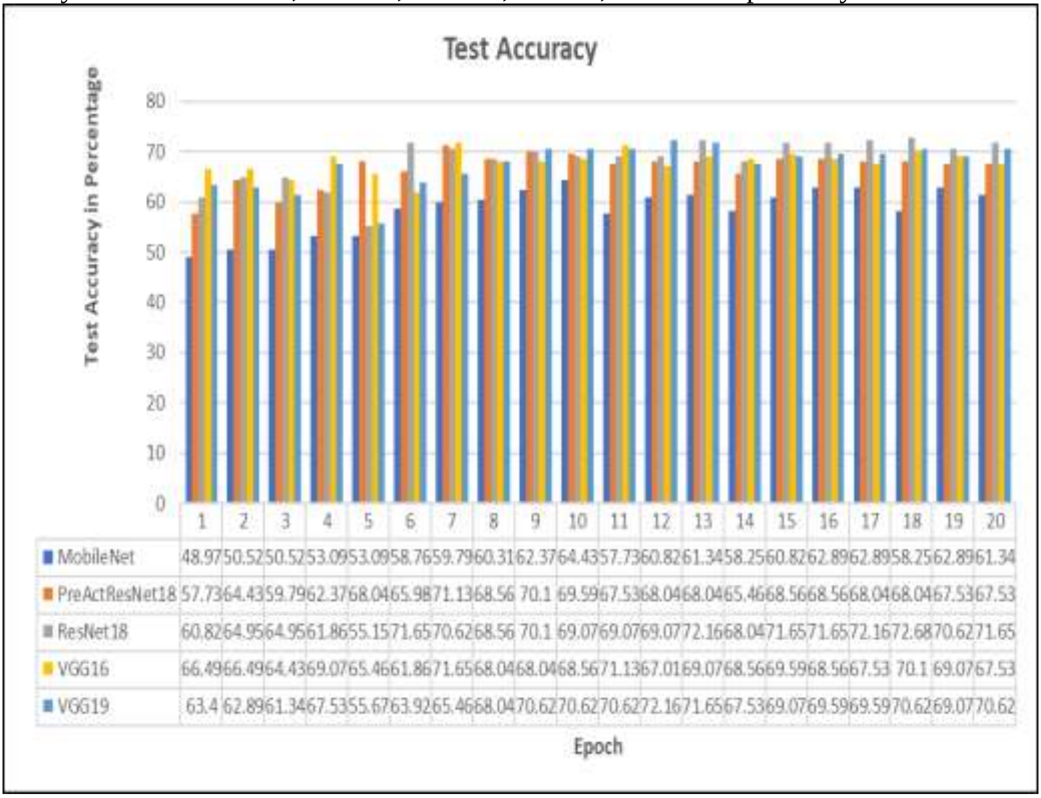
Table 2 contains the best train accuracy, test accuracy, and training time for the standard deep learning models trained for 20 epochs. These parameters are used to evaluate the rank based on algorithm 1. The rank of standard deep learning models is depicted in table 2.

Table 2 Detail performance and rank of standard deep learning models

Models	**Best Train Accuracy**	**Best Test Accuracy**	**Training Time for 20 Epoch**	**Rank**
MobileNet	59.65	64.43	41 Minutes 52 Seconds	1
PreActResNet18	70.78	71.13	127 Minutes 51 Seconds	5
ResNet18	71.29	72.68	121 Minutes 39 Seconds	4
VGG16	72.88	71.64	70 Minutes 5 Seconds	2

VGG19	72.76	72.16	81 Minutes 9 Seconds	3
Threshold value (Average)	69.47	70.40	88 minutes 31 second	-

Test Accuracy of standard deep learning models The top 1 accuracy achieved 59.65%, 70.78%, 71.88%, 72.88%, 72.76%, respectively. The test accuracy of MobileNet, Pre-Act Resnet18, ResNet18, VGG16, VGG19 standard models is depicted in Fig 10. The top 1 accuracy achieved 64.43 %, 71.13%, 72.68%,71.64%, 72.16% respectively.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MobileNet	48.97	50.52	50.52	53.09	53.09	58.76	59.79	60.31	62.37	64.43	57.73	60.82	61.34	58.25	60.82	62.89	62.89	58.25	62.89	61.34
PreActResNet18	57.73	64.43	59.79	62.37	68.04	65.98	71.13	68.56	70.1	69.59	67.53	68.04	68.04	65.46	68.56	68.56	68.04	68.04	67.53	67.53
ResNet18	60.82	64.95	64.95	61.86	55.15	71.65	70.62	68.56	70.1	69.07	69.07	69.07	72.16	68.04	71.65	71.65	72.16	72.68	70.62	71.65
VGG16	66.49	66.49	64.43	69.07	65.46	61.86	71.65	68.04	68.04	68.56	71.13	67.01	69.07	68.56	69.59	68.56	67.53	70.1	69.07	67.53
VGG19	63.4	62.89	61.34	67.53	55.67	63.92	65.46	68.04	70.62	70.62	70.62	72.16	71.65	67.53	69.07	69.59	69.59	70.62	69.07	70.62

Fig. 10 — Test Accuracy of standard deep learning models

5.2. Retrain the models using both label and pseudo dataset:

In this phase, the standard deep learning models are retrained with the combined MURA-SH and pseudo dataset generated by each standard deep learning model and our proposed model. In the next phase, these trained models will be validated on the HATA-SH unseen dataset.

5.3. Validation of proposed model on Unseen HATA-SH dataset:

In the last phase of the experiment, an unseen dataset HATA-SH is used to assess and validate the performance of the proposed model. The validation process is carried out in several parts to assess the performance. In the first part, we assess the performance of the standard deep learning models on the HATA-SH unseen dataset when no semi supervised learning technique was implemented. The result is depicted in table 3. The average measure for standard deep learning models such as accuracy, Specificity, sensitivity, precision, matthews correlation coefficient, false discovery rate, false positive rate, f1 score, negative predictive value, and false negative rate achieved 27.784%, 7.874%, 53.068%, 30.626%, -44.932, 69.374%, 92.126%, 38.732%, 16.936%, and 46.932% respectively.

Table 3 The Evaluation measure for Standard deep learning models without any semi-supervised approach

Measure	**MobileNet**	**PreActResNet18**	**ResNet18**	**VGG 16**	**VGG 19**	**Average**
Accuracy	36.27	19.97	27.17	21.59	33.92	27.784
Specificity	11.29	10.5	14.17	0.79	2.62	7.874
Sensitivity	68	32	43.67	48	73.67	53.068
Positive Predictive Value (Precision)	37.64	21.97	28.6	27.59	37.33	30.626
Matthews Correlation Coefficient	-25.51	-59.53	-44.6	-60.1	-34.92	-44.932
False Discovery Rate	62.36	78.03	71.4	72.41	62.67	69.374
False Positive Rate	88.71	89.5	85.83	99.21	97.38	92.126
F1 Score	48.46	26.05	34.56	35.04	49.55	38.732
Negative Predictive Value	30.94	16.39	24.22	1.89	11.24	16.936
False Negative Rate	32	68	56.33	52	26.33	46.932

In the second part, we assess the performance of the standard deep learning model on the HATA-SH unseen dataset when a pseudo dataset is generated through MobileNet. The result is depicted in table 4. The average measure for standard deep learning models such as accuracy, Specificity, sensitivity, precision, matthews correlation coefficient, false discovery rate, false positive rate, f1 score, negative predictive value, and false negative rate achieved 75.264%, 93.964%, 51.026, 86.534, 51.146%, 13.466%, 6.036%, 62.946%, 72.13%, and 48.974% respectively.

Table 4 The Evaluation measure for Standard deep learning models, pseudo dataset generated through MobileNet

Measure	MobileNet	PreActResNet18	ResNet 18	VGG16	VGG19	Average
Accuracy	64.61	74.45	68.58	79.15	89.53	75.264
Specificity	83.73	95.8	97.64	96.85	95.8	93.964
Sensitivity	40.33	47.33	31.67	56.67	79.13	51.026
Positive Predictive Value (Precision)	66.12	89.87	91.35	93.41	91.92	86.534
Matthews Correlation Coefficient	26.95	50.73	40.44	60.04	77.57	51.146
False Discovery Rate	33.88	10.13	8.65	6.59	8.08	13.466
False Positive Rate	16.27	4.2	2.36	3.15	4.2	6.036
F1 Score	50.1	62.01	47.03	70.54	85.05	62.946
Negative Predictive Value	64.06	69.79	64.47	73.95	88.38	72.13
False Negative Rate	59.67	52.67	68.33	43.33	20.87	48.974

In the third part, we assess the performance of the standard deep learning model on the HATA-SH unseen dataset when a pseudo dataset is generated through PreActResNet18. The result is depicted in table 5. The average measure for standard deep learning models such as accuracy, Specificity, sensitivity, precision, matthews correlation coefficient, false discovery rate, false positive rate, f1 score, negative predictive value, and false negative rate achieved 90.134%, 95.538%, 83.268%, 93.476%, 80.188%, 6.524%, 4.462%, 87.946%, 88.136%, 16.732% respectively.

Table 5 The Evaluation measure for Standard deep learning models, pseudo dataset generated through PreActResNet18

Measure	MobileNet	PreActResNet18	ResNet 18	VGG16	VGG19	Average
Accuracy	81.94	92.66	88.11	94.27	93.69	90.134
Specificity	91.34	96.33	97.11	96.85	96.06	95.538
Sensitivity	70	88	76.67	91	90.67	83.268
Positive Predictive Value (Precision)	86.42	94.96	95.44	95.79	94.77	93.476
Matthews Correlation	63.56	85.17	76.6	88.41	87.2	80.188

Coefficient						
False Discovery Rate	13.58	5.04	4.56	4.21	5.23	6.524
False Positive Rate	8.66	3.67	2.89	3.15	3.94	4.462
F1 Score	77.35	91.35	85.03	93.33	92.67	87.946
Negative Predictive Value	79.45	91.07	84.09	93.18	92.89	88.136
False Negative Rate	30	12	23.33	9	9.33	16.732

In the fourth part, we assess the performance of the standard deep learning model on the HATA-SH unseen dataset when a pseudo dataset is generated through ResNet18. The result is depicted in table 6. The average measure for standard deep learning models such as accuracy, Specificity, sensitivity, precision, matthews correlation coefficient, false discovery rate, false positive rate, f1 score, negative predictive value, and false negative rate achieved 89.368%, 88.662%, 90.268%, 86.288%, 78.622%, 13.712%, 11.338%, 88.228%, 92.03%, 9.732% respectively.

Table 6 The Evaluation measure for Standard deep learning models, pseudo dataset generated through ResNet18

Measure	**MobileNet**	**PreActResNet18**	**ResNet18**	**VGG16**	**VGG19**	**Average**
Accuracy	79.88	92.22	92.95	92.07	89.72	89.368
Specificity	79.79	90.29	93.18	91.34	88.71	88.662
Sensitivity	80	94.67	92.67	93	91	90.268
Positive Predictive Value (Precision)	75.71	88.47	91.45	89.42	86.39	86.288
Matthews Correlation Coefficient	59.51	84.49	85.73	84.03	79.35	78.622
False Discovery Rate	24.29	11.53	8.55	10.58	13.61	13.712
False Positive Rate	20.21	9.71	6.82	8.66	11.29	11.338
F1 Score	77.8	91.47	92.05	91.18	88.64	88.228
Negative Predictive Value	83.52	95.56	94.16	94.31	92.6	92.03
False Negative Rate	20	5.33	7.33	7	9	9.732

In the fifth part, we assess the performance of the standard deep learning model on the HATA-SH unseen dataset when a pseudo dataset is generated through VGG16. The result is depicted in table 7. The average measure for standard deep learning models such as accuracy, Specificity, sensitivity, precision, matthews correlation coefficient, false discovery rate, false positive rate, f1 score, negative predictive value, and false negative rate achieved 90.602%, 96.85%, 82.668%, 95.282%, 81.254%, 4.718%, 3.15%, 88.472%, 87.764%, 17.332% respectively.

Table 7 The Evaluation measure for Standard deep learning models, pseudo dataset generated through VGG16

Measure	**MobileNet**	**PreActResNet 18**	**ResNet 18**	**VGG16**	**VGG19**	**Average**
Accuracy	83.85	93.39	91.19	91.78	92.8	90.602
Specificity	93.7	97.9	97.64	98.16	96.85	96.85
Sensitivity	71.33	87.67	83	83.67	87.67	82.668
Positive Predictive Value (Precision)	89.92	97.05	96.51	97.29	95.64	95.282
Matthews Correlation Coefficient	67.71	86.79	82.52	83.74	85.51	81.254
False Discovery Rate	10.08	2.95	3.49	2.71	4.36	4.718
False Positive Rate	6.3	2.1	2.36	1.84	3.15	3.15
F1 Score	79.55	92.12	89.25	89.96	91.48	88.472
Negative Predictive Value	80.59	90.98	87.94	88.42	90.89	87.764
False Negative Rate	28.67	12.33	17	16.33	12.33	17.332

In the sixth part, we assess the performance of the standard deep learning model on the HATA-SH unseen dataset when a pseudo dataset is generated through VGG19. The result is depicted in table 8; the average measure for standard deep learning models such as accuracy, Specificity, sensitivity, precision, matthews correlation coefficient, false discovery rate, false positive rate, f1 score, negative predictive value, and false negative rate achieved 69.308%, 97.742%, 33.198%, 89.522%, 40.81%, 10.478%, 2.258%, 47.456%, and 65.28%, 66.802% respectively.

Table 8 The Evaluation measure for Standard deep learning models, pseudo dataset generated through VGG19

Measure	**MobileNet**	**PreActResNet 18**	**ResNet 18**	**VGG16**	**VGG19**	**Average**
Accuracy	59.32	67.69	68.72	75.48	75.33	69.308
Specificity	97.11	97.11	96.85	98.69	98.95	97.742
Sensitivity	11.33	30.33	33	46	45.33	33.198
Positive Predictive Value (Precision)	75.56	89.22	89.19	96.5	97.14	89.522
Matthews Correlation Coefficient	16.88	38.18	40.12	54.47	54.4	40.81
False Discovery Rate	24.44	10.78	10.81	3.5	2.86	10.478
False Positive Rate	2.89	2.89	3.15	1.31	1.05	2.258
F1 Score	19.71	45.27	48.18	62.3	61.82	47.456
Negative Predictive Value	58.18	63.9	64.74	69.89	69.69	65.28
False Negative Rate	88.67	69.67	67	54	54.67	66.802

In the seventh part, we assess the performance of the standard deep learning model on the HATA-SH unseen dataset when a pseudo dataset is generated through our proposed model. The result is depicted in table 9. The average measure for standard deep learning models such as accuracy, Specificity, sensitivity, precision, matthews correlation coefficient, false discovery rate, false positive rate, f1 score, negative predictive value, and false negative rate achieved 92.776%, 97.376%, 86.932%, 96.192%, 85.644%, 3.808%, 2.624%, 91.072%, 90.85%, 13.068% respectively. In table 9 we have compare our result with state of art and improve the accuracy by 1 percentage.

Table 9 The Evaluation measure for Standard deep learning models, pseudo dataset generated through our proposed model

Measure	**MobileNet**	**PreActResNet 18**	**ResNet 18**	**VGG16**	**VGG19**	**Proposed Model (average)**	**state of art** [41]
Accuracy	83.7	93.83	95.89	94.57	95.89	92.776	91.83
Specificity	96.59	96.85	98.69	96.85	97.9	97.376	85.26

Sensitivity	67.33	90	92.33	91.67	93.33	86.932	97.00
Positive Predictive Value (Precision)	93.95	95.74	98.23	95.82	97.22	96.192	89.35
Matthews Correlation Coefficient	68.27	87.54	91.74	88.99	91.68	85.644	83.64
False Discovery Rate	6.05	4.26	1.77	4.18	2.78	3.808	10.64
False Positive Rate	3.41	3.15	1.31	3.15	2.1	2.624	14.73
F1 Score	78.45	92.78	95.19	93.7	95.24	91.072	93.01
Negative Predictive Value	78.97	92.48	94.24	93.65	94.91	90.85	95.69
False Negative Rate	32.67	10	7.67	8.33	6.67	13.068	2.99

In Fig 11, we show some representative samples together with our proposed model correct and incorrect predictions for the shoulder bone fracture classification on the unseen HATA-SH dataset. We can see that our model classification performance still has space for improvement. We used the basic deep learning models as the mainstay in this work rather than more sophisticated model designs since we are interested in exploring how to efficiently use unlabelled data and assist the medical domain.

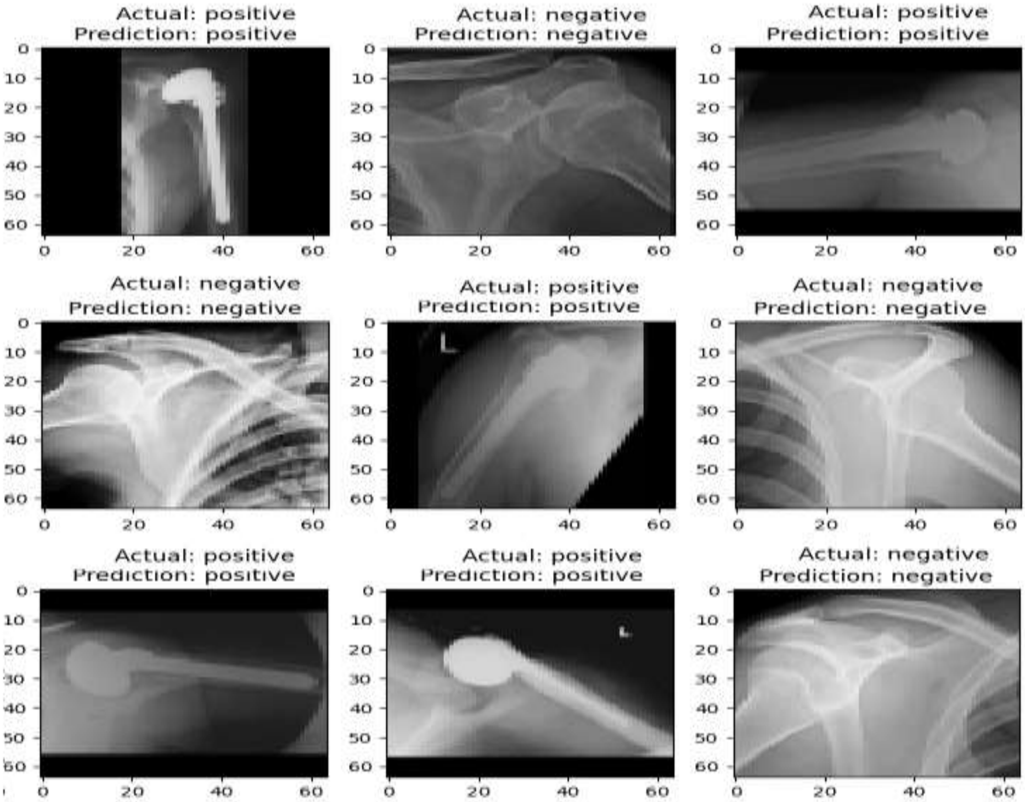


Fig. 11 — A typical example of a correct and incorrect prediction and actual label through the proposed model.

It has been observed that our proposed model has outperformed other models. Several performance evaluation measures are considered, such as accuracy, specificity, sensitivity, precision, matthews correlation coefficient, false discovery rate, false positive rate, f1 score, negative predictive value, and false negative rate. 92.776%, 97.376%, 86.932%, 96.192%, 85.644%, 3.808%, 2.624%, 91.072%, 90.85%, and 13.068% respectively for unseen dataset. Overall, our proposed models exhibited high performance without the semi-supervised approach being the weakest approach compared to others.

The proposed model achieves an increase by 234%, 1137%, 64%, 214%, 135%, 436% inaccuracy, specificity, sensitivity, precision, fi score, matthews correlation coefficient, negative predictive value, respectively and decreases by -95%, -97%, -72% in FDR, FPR, and FNR respectively. The result is depicted in Fig 12.

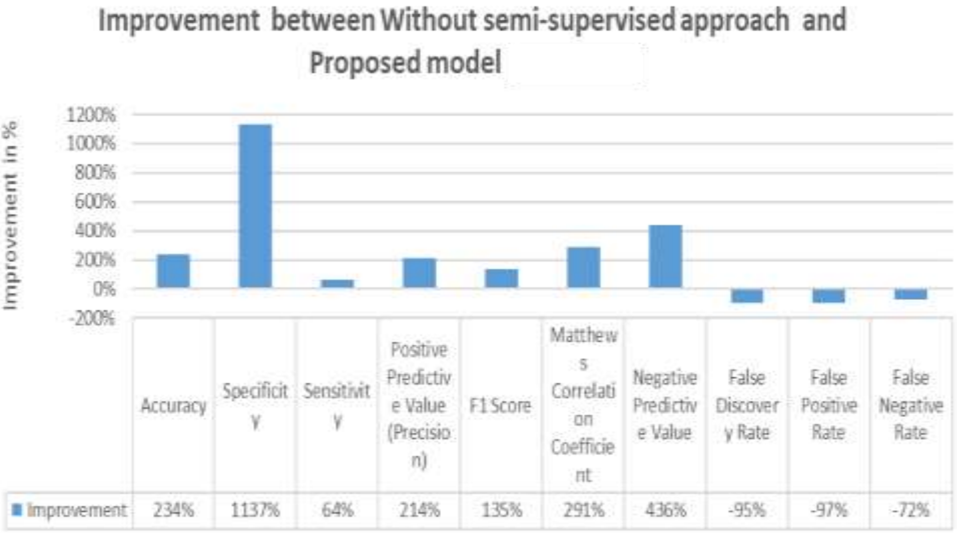


Fig. 12 — Improvement between without semi-supervised approach and proposed model (TSR-SDL)

Limitation: Proposed methods are a powerful technique for improving the performance and robustness of machine learning models. However, there are some limitations that need to be considered when using ensemble methods. Here are some of the limitations of the ensemble method:

1. Increased computational complexity: Proposed methods require training multiple models, which can significantly increase the computational complexity and time required to train and evaluate the models.

2. Overfitting: Proposed methods can be prone to overfitting if the models in the ensemble are too complex and/or the training data is limited. This can lead to poor generalization performance on new data.

3. Limited interpretability: Proposed methods are often considered as "black box" models, meaning that it can be difficult to interpret the results and understand how the ensemble arrived at its predictions. This can be a limitation in certain applications, especially in high-risk domains such as healthcare and finance.

4. Sensitivity to individual model performance: Proposed methods depend on the performance of individual models in the ensemble. If one or more models in the ensemble perform poorly, it can negatively impact the overall performance of the ensemble.

5. Difficulty in selecting the right models: Proposed methods require selecting a set of diverse and complementary models to achieve optimal performance. However,

selecting the right models can be a challenging task, and the performance of the ensemble is highly dependent on the choice of models.

In summary, while proposed methods are a powerful technique for improving machine learning performance, they are not without limitations. Careful consideration is required when selecting models, evaluating performance, and interpreting results to ensure that the benefits of ensemble methods outweigh their limitations.

6. Conclusions:

In this research paper, Semi supervised deep learning models have shown promising results in medical imaging, where labeled data is often scarce and expensive to acquire. By utilizing both labeled and unlabeled data, semi supervised deep learning models can improve the accuracy of medical image analysis and diagnosis. Here are some practical applications of semi supervised deep learning models in medical imaging:

- Detection of Abnormalities: Semi supervised deep learning models can be used to detect abnormalities in medical images, such as tumors or lesions. By training the model on both labeled and unlabeled data, the model can learn to identify subtle patterns and features that are indicative of an abnormality.

- Disease Diagnosis: Semi supervised deep learning models can be used to aid in the diagnosis of diseases, such as Alzheimer's or Parkinson's. By utilizing both labeled and unlabeled data, the model can learn to identify patterns and features in medical images that are indicative of the disease.

Overall, semi supervised deep learning models have the potential to significantly improve the accuracy and efficiency of medical image analysis and diagnosis, especially in cases where labeled data is scarce or expensive to acquire. Although accuracy is extensive applicability, it is not always the best performance statistic to use, especially when the target variable classes in the dataset are imbalanced. Low (FPR, FNR, and FDR) and high (sensitivity, specificity, precision, and Matthew’s correlation coefficient) indicate an efficient and effective model. The proposed model achieves an increase in 234%, 1137%, 64%, 214%, 135% 436% accuracy, specificity, sensitivity, precision, f1 score, Matthew’s correlation coefficient, negative predictive value respectively and decrease by -95%, -97%, -72% in FDR, FPR, and FNR respectively. We can conclude that the model trained with (MURA) dataset from Stanford University was not enough to predict the data set collected HATA CHC. This proposed semi-supervised learning approach solve the issue of unlabelled dataset and also improves the performance of the model. In the future, this proposed method may be applied to a wide variety of other medical imaging datasets.

Compliance with Ethical Standards

The authors declare that they do not have any Conflict of Interest. This research does not include any human or animal participation.

Availability of data and materials

Data shall be made available from corresponding author on request.

Funding

This work does not receive any kind of funding in any form.

Acknowledgements

Not applicable

References

[1] A. Agrawala, "Learning With A Probabilistic Teacher," Ieee Trans. Inf. Theory, Vol. 16, No. 4, Pp. 373–379, 1970, Doi: 10.1109/Tit.1970.1054472.

[2] S. Fralick, "Learning To Recognize Patterns Without A Teacher," Ieee Trans. Inf. Theory, Vol. 13, No. 1, Pp. 57–64, 1967, Doi: 10.1109/Tit.1967.1053952.

[3] H. Scudder, "Probability Of Error Of Some Adaptive Pattern-Recognition Machines," Ieee Trans. Inf. Theory, Vol. 11, No. 3, Pp. 363–371, 1965, Doi: 10.1109/Tit.1965.1053799.

[4] P. Rajpurkar Et Al., "Mura: Large Dataset For Abnormality Detection In Musculoskeletal Radiographs," Arxiv Prepr. Arxiv1712.06957, 2017.

[5] D.-H. Lee And Others, "Pseudo-Label: The Simple And Efficient Semi-Supervised Learning Method For Deep Neural Networks," In Workshop On Challenges In Representation Learning, Icml, 2013, Vol. 3, No. 2.

[6] Q. Xie, M.-T. Luong, E. Hovy, And Q. V Le, "Self-Training With Noisy Student Improves Imagenet Classification," Jun. 2020.

[7] M. Ragab, S. Alshehri, N. A. Alhakamy, R. F. Mansour, And D. Koundal, "Multiclass Classification Of Chest X-Ray Images For The Prediction Of Covid-19 Using Capsule Network," Comput. Intell. Neurosci., Vol. 2022, 2022.

[8] A. Sharma, K. Singh, And D. Koundal, "A Novel Fusion Based Convolutional Neural Network Approach For Classification Of Covid-19 From Chest X-Ray Images," Biomed. Signal Process. Control, Vol. 77, P. 103778, 2022.

[9] P. Malhotra, S. Gupta, D. Koundal, A. Zaguia, M. Kaur, And H.-N. Lee, "Deep Learning-Based Computer-Aided Pneumothorax Detection Using Chest X-Ray Images," Sensors, Vol. 22, No. 6, P. 2278, 2022.

[10] A. S. Al-Ghamdi, M. Ragab, S. A. Alghamdi, A. H. Asseri, R. F. Mansour, And D. Koundal, "Detection Of Dental Diseases Through X-Ray Images Using Neural Search Architecture Network," Comput. Intell. Neurosci., Vol. 2022, 2022.

[11] P. Malhotra, S. Gupta, And D. Koundal, "Comparative Analysis Of Deep Learning Based Automated Segmentation Of Pneumothorax On Chest X-Ray Images," Ecs Trans., Vol. 107, No. 1, P. 8905, 2022.

[12] S. Laine And T. Aila, "Temporal Ensembling For Semi-Supervised Learning," Arxiv Prepr. Arxiv1610.02242, 2016.

[13] S. Choudhary, V. Narayan, M. Faiz, And S. Pramanik, "Fuzzy Approach-Based Stable Energy-Efficient Aodv Routing Protocol In Mobile Ad Hoc Networks," In Software Defined Networking For Ad Hoc Networks, Springer, 2022, Pp. 125–139.

[14] Ouali, Y., Hudelot, C., & Tami, M. (2020). An Overview Of Deep Semi-Supervised Learning. Arxiv Preprint Arxiv:2006.05278.

[15] A. Tarvainen And H. Valpola, "Mean Teachers Are Better Role Models: Weight-Averaged Consistency Targets Improve Semi-Supervised Deep Learning Results," Arxiv Prepr. Arxiv1703.01780, 2017.

[16] Li, X., Lu, P., Hu, L., Wang, X., & Lu, L. (2022). A Novel Self-Learning Semi-Supervised

Deep Learning Network To Detect Fake News On Social Media. Multimedia Tools And Applications, 81(14), 19341-19349.

[17] P. Smiti, S. Srivastava, And N. Rakesh, “Video And Audio Streaming Issues In Multimedia Application,” In 2018 8th International Conference On Cloud Computing, Data Science Engineering (Confluence), 2018, Pp. 360–365, Doi: 10.1109/Confluence.2018.8442823.

[18] S. Srivastava And S. Sharma, “Analysis Of Cyber Related Issues By Implementing Data Mining Algorithm,” In 2019 9th International Conference On Cloud Computing, Data Science Engineering (Confluence), 2019, Pp. 606–610, Doi: 10.1109/Confluence.2019.8776980.

[19] T. Miyato, S. Maeda, M. Koyama, And S. Ishii, “Virtual Adversarial Training: A Regularization Method For Supervised And Semi-Supervised Learning,” Ieee Trans. Pattern Anal. Mach. Intell., Vol. 41, No. 8, Pp. 1979–1993, 2018, Doi: 10.1109/Tpami.2018.2858821.

[20] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, And C. Raffel, “Mixmatch: A Holistic Approach To Semi-Supervised Learning,” Arxiv Prepr. Arxiv1905.02249, 2019.

[21] Dong, S., Xia, Y., & Peng, T. (2021). Network Abnormal Traffic Detection Model Based On Semi-Supervised Deep Reinforcement Learning. Ieee Transactions On Network And Service Management, 18(4), 4197-4212.

[22] Dong-Dongchen, W., & Weigao, Z. H. (2018, July). Tri-Net For Semi-Supervised Deep Learning. In Proceedings Of Twenty-Seventh International Joint Conference On Artificial Intelligence (Pp. 2014-2020).

[23] Shi, W., Gong, Y., Ding, C., Tao, Z. M., & Zheng, N. (2018). Transductive Semi-Supervised Deep Learning Using Min-Max Features. In Proceedings Of The European Conference On Computer Vision (Eccv) (Pp. 299-315).

[24] Chen, T., Kornblith, S., Swersky, K., Norouzi, M., & Hinton, G. E. (2020). Big Self-Supervised Models Are Strong Semi-Supervised Learners. Advances In Neural Information Processing Systems, 33, 22243-22255.

[25] M. A. Mohammed Et Al., “Novel Crow Swarm Optimization Algorithm And Selection Approach For Optimal Deep Learning Covid-19 Diagnostic Model,” Comput. Intell. Neurosci., Vol. 2022, 2022.

[26] A. T. Nagi, M. J. Awan, M. A. Mohammed, A. Mahmoud, A. Majumdar, And O. Thinnukool, “Performance Analysis For Covid-19 Diagnosis Using Custom And State-Of-The-Art Deep Learning Models,” Appl. Sci., Vol. 12, No. 13, P. 6364, 2022.

[27] J. N. Hasoon Et Al., “Covid-19 Anomaly Detection And Classification Method Based On Supervised Machine Learning Of Chest X-Ray Images,” Results Phys., Vol. 31, P. 105045, 2021.

[28] D. A. Ibrahim, D. A. Zebari, H. J. Mohammed, And M. A. Mohammed, “Effective Hybrid Deep Learning Model For Covid-19 Patterns Identification Using Ct Images,” Expert Syst., P. E13010, 2022.

[29] H. Allioui Et Al., “A Multi-Agent Deep Reinforcement Learning Approach For Enhancement Of Covid-19 Ct Image Segmentation,” J. Pers. Med., Vol. 12, No. 2, P. 309, 2022.

[30] A. G. Howard Et Al., “Mobilenets: Efficient Convolutional Neural Networks For Mobile Vision Applications,” Arxiv Prepr. Arxiv1704.04861, 2017.

[31] K. He, X. Zhang, S. Ren, And J. Sun, “Identity Mappings In Deep Residual Networks,” Mar. 2016, [Online]. Available: Http://Arxiv.Org/Abs/1603.05027.

[32] K. He, X. Zhang, S. Ren, And J. Sun, “Deep Residual Learning For Image Recognition,” Corr, Vol. Abs/1512.0, 2015, [Online]. Available: Http://Arxiv.Org/Abs/1512.03385.

[33] X. Zhang, J. Zou, K. He, And J. Sun, “Accelerating Very Deep Convolutional Networks For Classification And Detection,” Ieee Trans. Pattern Anal. Mach. Intell., Vol. 38, No. 10, Pp. 1943–1955, 2015.

[34] K. Simonyan And A. Zisserman, “Very Deep Convolutional Networks For Large-Scale Image

Recognition," Arxiv Prepr. Arxiv1409.1556, 2014.

[35] F. Setiawan, B. N. Yahya, And S.-L. Lee, "Deep Activity Recognition On Imaging Sensor Data," Electron. Lett., Vol. 55, No. 17, Pp. 928–931, 2019.

[36] Li, Z., Ko, B., & Choi, H. J. (2019). Naive Semi-Supervised Deep Learning Using Pseudo-Label. Peer-To-Peer Networking And Applications, 12, 1358-1368..

[37] Y. Benjamini, "Discovering The False Discovery Rate," J. R. Stat. Soc. Ser. B (Statistical Methodol., Vol. 72, No. 4, Pp. 405–416, 2010.

[38] Xiao, Y., Wu, J., Lin, Z., & Zhao, X. (2018). A Semi-Supervised Deep Learning Method Based On Stacked Sparse Auto-Encoder For Cancer Prediction Using Rna-Seq Data. Computer Methods And Programs In Biomedicine, 166, 99-105.

[39] D. Chicco And G. Jurman, "The Advantages Of The Matthews Correlation Coefficient (Mcc) Over F1 Score And Accuracy In Binary Classification Evaluation," Bmc Genomics, Vol. 21, No. 1, Pp. 1–13, 2020.

[40] Yao, H., Fu, D., Zhang, P., Li, M., & Liu, Y. (2018). Msml: A Novel Multilevel Semi-Supervised Machine Learning Framework For Intrusion Detection System. Ieee Internet Of Things Journal, 6(2), 1949-1959.

[41] P Mall And P. Singh, "Credence-Net: A Semi-Supervised Deep Learning Approach For Medical Images," Int. J. Nanotechnol., Vol. 20, 2022.

[42] Narayan, Vipul, Et Al. "7 Extracting Business Methodology: Using Artificial Intelligence-Based Method." Semantic Intelligent Computing And Applications 16 (2023): 123.

[43] Narayan, Vipul, Et Al. "A Comprehensive Review Of Various Approach For Medical Image Segmentation And Disease Prediction." Wireless Personal Communications 132.3 (2023): 1819-1848.

[44] Mall, Pawan Kumar, Et Al. "Rank Based Two Stage Semi-Supervised Deep Learning Model For X-Ray Images Classification: An Approach Toward Tagging Unlabeled Medical Dataset." Journal Of Scientific & Industrial Research (Jsir) 82.08 (2023): 818-830.

[45] Chaturvedi, Pooja, A. K. Daniel, And Vipul Narayan. "A Novel Heuristic For Maximizing Lifetime Of Target Coverage In Wireless Sensor Networks." Advanced Wireless Communication And Sensor Networks. Chapman And Hall/Crc, 2023. 227-242.

[46] Narayan, Vipul, Et Al. "Severity Of Lumpy Disease Detection Based On Deep Learning Technique." 2023 International Conference On Disruptive Technologies (Icdt). Ieee, 2023.

[47] Narayan, Vipul, Et Al. "Fuzzynet: Medical Image Classification Based On Glcm Texture Feature." 2023 International Conference On Artificial Intelligence And Smart Communication (Aisc). Ieee, 2023

[48] Narayan, Vipul, Et Al. "Deep Learning Approaches For Human Gait Recognition: A Review." 2023 International Conference On Artificial Intelligence And Smart Communication (Aisc). Ieee, 2023.

[49] Mall, Pawan Kumar, Et Al. "Fuzzynet-Based Modelling Smart Traffic System In Smart Cities Using Deep Learning Models." Handbook Of Research On Data-Driven Mathematical Modeling In Smart Cities. Igi Global, 2023. 76-95.

[50] Narayan, Vipul, And A. K. Daniel. "Energy Efficient Protocol For Lifetime Prediction Of Wireless Sensor Network Using Multivariate Polynomial Regression Model." Journal Of Scientific & Industrial Research 81.12 (2022): 1297-1309.

Biographies

Pawan Kumar Mall: Pawan Kumar Mall received M. Tech degree in Computer Science Engineering from A.KT.U. India in 2016. Presently he is working as a research scholar in the Department of Computer Science & Engineering. His current research interests are Wireless

Sensor Networks and Cloud computing and Image Processing He has published various papers in International Journals and International conferences.

Pradeep Kumar Singh: Pradeep Kumar Singh received the B.E. Degree in Computer Science from D. D. U. University of Gorakhpur, Gorakhpur, India, the MTech degree in Computer Science and Technology from University of Roorkee, Roorkee (now IITR), India and Ph.D. in Computer Science and Engineering from D.D.U. University of Gorakhpur, Gorakhpur, India. He is currently working as Professor with Department of Computer Science and Engineering, Madan Mohan Malaviya University of Technology, Gorakhpur, India. His current research interests include memory and parallelism optimization for embedded systems, multi core architectures and compiler optimization.