01 Sep 2024

Transfer Learning with Spinally Shared Layers

H. M.Dipu Kabir

Subrota Kumar Mondal

Syed Bahauddin Alam
Missouri University of Science and Technology, alams@illinois.edu

U. Rajendra Acharya

Recommended Citation

H. M. Kabir et al., "Transfer Learning with Spinally Shared Layers," *Applied Soft Computing*, vol. 163, article no. 111908, Elsevier, Sep 2024.
The definitive version is available at https://doi.org/10.1016/j.asoc.2024.111908

Applied Soft Computing Journal 163 (2024) 111908

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Transfer learning with spinally shared layers

H.M. Dipu Kabir [a,b,c,*], Subrota Kumar Mondal [d], Syed Bahauddin Alam [e,f,g], U. Rajendra Acharya [h,i]

[a] Artificial Intelligence and Cyber Futures Institute, Charles Sturt University, Australia
[b] Rural Health Research Institute, Charles Sturt University, Australia
[c] Independent Researcher, Australia
[d] FST, Macau University of Science and Technology, Macao
[e] Nuclear, Plasma & Radiological Engineering, University of Illinois Urbana-Champaign, USA
[f] National Center for Supercomputing Application, 205 W Clark Street, Urbana, IL, 61801, USA
[g] NERS, Missouri University of Science and Technology, USA
[h] Department of Biomedical Engineering, School of Science and Technology, SUSS University, Singapore
[i] School of Mathematics, Physics and Computing, University of Southern Queensland, Springfield, Australia

ARTICLE INFO

Keywords:
Uncertainty
Transformer
SpinalNet
COVID
ResNet
VGG

ABSTRACT

Transfer-learned models have achieved promising performance in numerous fields. However, high-performing transfer-learned models contain a large number of parameters. In this paper, we propose a transfer learning approach with parameter reduction and potential high performance. Although the high performance depends on the nature of the dataset, we ensure the parameter reduction. In the proposed SpinalNet shared parameters, all intermediate-split-incoming parameters except the first-intermediate-split contain a shared value. Therefore, the SpinalNet shared parameters network contains three parameter groups: (1) first input-split to intermediate-split parameters, (2) shared intermediate-split-incoming parameters, and (3) intermediate-split-to-output-split parameters. The total number of parameters becomes lower than the SpinalNet and traditional fully connected layers due to parameter sharing. Besides the overall accuracy, this paper compares the precision, recall, and F1-score of each class as performance criteria. As a result, both parameter reduction and potential performance improvement become possible for the ResNet-type models, VGG-type traditional models, and Vision Transformers. We applied the proposed model to MNIST, STL-10, and COVID-19 datasets to validate our claims. We also provided a posterior plot of the sample from different models for medical practitioners to understand the uncertainty. Example model training scripts of the proposed model are also shared to GitHub.

1. Introduction

Deep neural networks (DNNs) are getting huge attention due to their recent eye-catching performances. Researchers are investigating NN models of different structures to achieve improved performances. The performances of NNs are also improving due to the continuing research over decades [1]. Researchers have observed significantly higher accuracy with convolutional parameters. Convolutional layers are developed by observing the cat's cortex [2,3]. DNNs containing convolutional layers are often called convolutional neural networks (CNN) or deep convolutional neural networks (DCNN). Integration of residual units have further improved the performance of DNNs. However, the number of parameters of DNNs has increased dramatically with the improvement of accuracy over years. Some of the DNN models contain a large number of parameters on the fully connected part [4–6]. Therefore, a model with reduced fully connected layer parameters and improved accuracy can potentially be applied in many real-life Machine Learning (ML) applications.

Researchers are also concerned with the uncertainty in deep learning models [7–9]. DNNs often fail to predict, and traditional DNN models cannot express their confidence while predicting a sample. Moreover, popular models often fail to predict with high confidence [10,11]. Researchers are also improving models over time following different approaches, such as dropout and adversarial training [12–14]. In regression problems, the level of uncertainty is popularly presented as the width of the prediction interval [15]. However, classification problems are still lacking robust performance criteria. Therefore, the proposed method provides medical practitioners output posteriors from an ensemble of models to understand the exact uncertain situation [16].

The performances of top ML models for computer vision are comparable to human eyes classifying hand-written digits and natural

[*] Corresponding author at: Artificial Intelligence and Cyber Futures Institute, Charles Sturt University, Australia.
E-mail address: hmdkabir@connect.ust.hk (H.M.D. Kabir).

https://doi.org/10.1016/j.asoc.2024.111908
Received 19 January 2024; Received in revised form 22 May 2024; Accepted 15 June 2024
Available online 25 June 2024

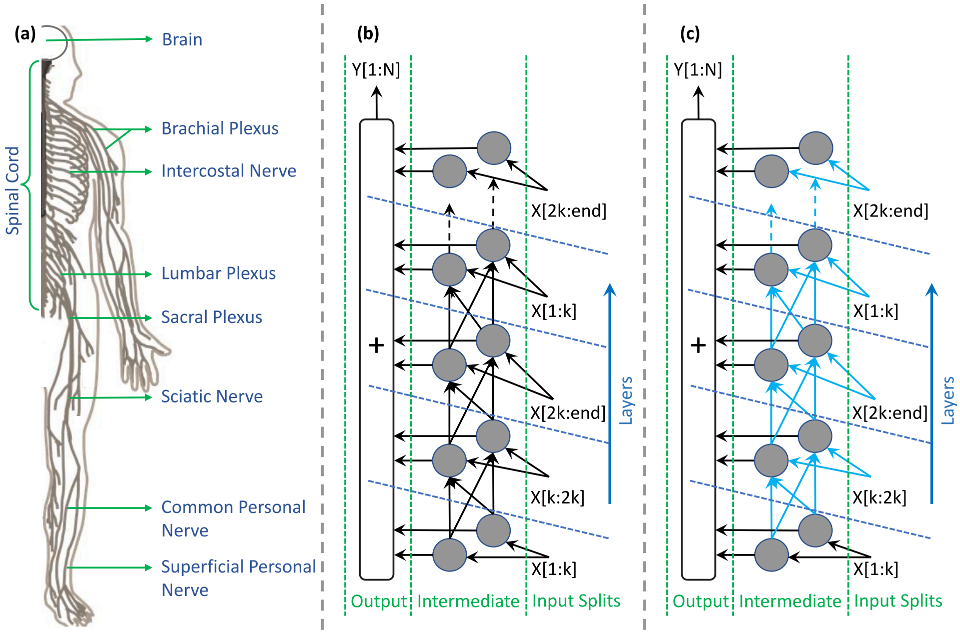

Fig. 1. SpinalNet and SpinalNet(Shared Weight) (a) Nerve connections in the spinal cord, (b) SpinalNet structure [25], (c) SpinalNet structure with shared weight. Blue arrows present weights with the same values. Except for the first layer in the intermediate split, the input of the intermediate split comes from the corresponding input split and the previous layer in the intermediate split. As the input is uniformly split, those input connections have the same number of parameters. The proposed method applies shared weight and bias values for those input connections.

images [17]. ML models are also good at regression and uncertainty quantification. However, researchers need to rethink while working on novel problems [18–20]. Specialists and scientists often struggle to detect and handle novel problems [21]. Datasets containing samples of novel diseases, often contain mislabeled data. There might be many patients with mild symptoms and mild lung conditions. There exists variance among medical practitioners in labeling the data. Many samples may contain partial symptoms [22,23]. The sample can be different based on recent eating, sleeping, and exercising patterns. The current process of collecting samples for COVID-19 diagnosis is troublesome for the sample collector and the patient. The patient and the sample collector face an awkward situation while collecting the nasal sample. Diagnosis of the COVID-19 disease from X-ray images can potentially be an optimal approach [16,24].

Recently researchers are focusing on Multitask Learning [26]. End layers are replicated, restructured, and re-trained for different tasks in Multitask Learning. Therefore, the requirement for reducing the fully connected layer can potentially increase greatly due to Multitask Learning. SpinalNet is getting popularity due to its eye-catching performance [25,27,28]. The Spinal fully connected layer can easily replace the fully-connected layer in many convolutional models. Moreover, previous convolutional layers can be pre-trained by a large and publicly available dataset, such as ImageNet [29]. SpinalNet fully connected layer with pre-trained convolutional layers has obtained state-of-the-art (SOTA) performance in several datasets. Researchers are also applying SpinalNet to new applications and receiving good performances over time [30]. However, SpinalNet does not ensure parameter reduction or performance enhancement in all situations. The performance enhancement depends on the dataset. The parameter reduction happens with the VGG-type models [31]. Therefore, in this paper, we propose the SpinalNet with shared weight to ensure parameter reduction with different types of models.

2. Theoretical background and proposal

This section presents a short overview of the current COVID-19 diagnosis, relevant DNNs, the proposal of SpinalNet, transferred initialization, and uncertainty in NN. This section can potentially help readers in getting a short overview of relevant literature.

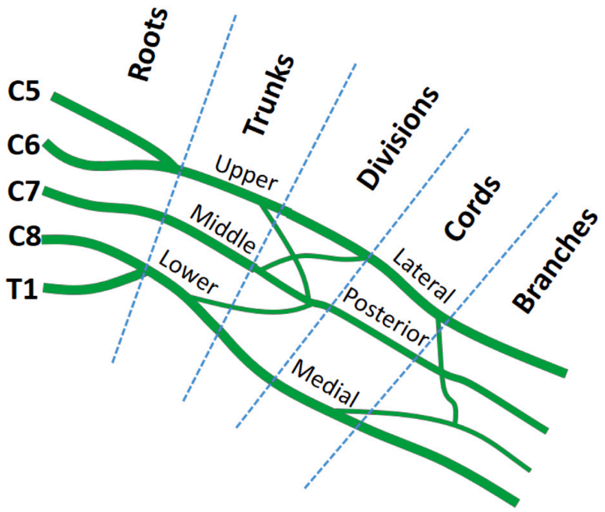

Fig. 2. Brachial plexus: a portion of the human nerve plexus. Sense of any touch or pain conveys to our brain via the spinal cord and the nerve plexus. A network of convoluted nerves is the nerve plexus. Our spinal cord receives information gradually. Sensory carried by a branch may reach multiple roots, entering vertebrae. Reproduced with the permission of authors [25].

2.1. Current approach of COVID-19 diagnosis

Specimens from upper and lower respiratory are popularly used for COVID-19 diagnosis through popular approaches. Popular approaches are real-time reverse transcription polymerase chain reaction (RT-PCR), and Rapid Antigen Testing (RAT). The COVID-19 patient gets the symptom of the disease and reaches health authorities. Common symptoms are sore throat, fever, cough, and altered sense of smell or taste. Health authorities usually do RT-PCR or RAT tests for the initial diagnosis [32]. Chest X-ray is collected from the hospitalized patient to observe the condition of internal organs. However, the commonly applied methods of COVID-19 diagnosis require nasal swabs. The collection method of nasal swabs is troublesome to both the patient and the sample collector. Taking a chest X-ray is more convenient and can be applied for the diagnosis of multiple diseases. Some of those diseases, such as pneumonia, or any infection near the respiratory system can potentially have several common symptoms.

2.2. Deep neural networks

Deep Neural Networks (DNNs) are getting attention as state-of-the-art (SOTA) models in various domains. The accuracy of the neural network (NN) models has increased greatly after the inception of convolutional blocks. Convolutional blocks have reduced the number of parameters in NNs and helped researchers to make deeper and high-performing NNs. However, researchers got the opportunity to make NNs of different convolutional structures. Different research groups have proposed different standard NNs over time [33]. Most of the proposed DNNs have shown improved results, parameter reduction, or improved run-time when their work was proposed. Some of the most popular and relevant DNNs are as follows:

2.2.1. VGG

The VGG Net is a popular convolutional neural network (CNN) in image processing. VGG Net is also known as a DNN due to its high depth. The DNN was proposed in 2014 by Zisserman and Simonyan at the Visual Geometry Group [34]. VGG Net is still one of the high-performing models in several domains. Especially in medical image datasets and handwritten character classification datasets.

Fig. 3 presents how SpinalNet can be integrated with the VGG neural network to achieve potentially better performance with parameter reduction. Fig. 3(a) presents the VGG-19 neural network. The VGG-19 has sixteen convolutional layers and three fully connected layers. Three fully connected layers receive flattened data. Therefore, it is possible to replace fully connected layers with traditional shallow neural networks. Three fully connected layers in the VGG network are also known as the classifier. We replace the classifier with SpinalNet shared weight in the current work. Fig. 3(b) presents the VGG-19 neural network with the SpinalNet classifier layer.

2.2.2. ResNet

Residual NN (ResNet), proposed by Kaiming He is one of the most investigated neural networks. The paper proposed the ResNet model in 2016 is also one of the most cited papers of all time [1]. The ResNet structure is a deep convolutional neural network model with skip connections. In a DNN, one wrongly trained layer can potentially degrade the overall performance greatly. Moreover, there exists an increasing vanishing gradient problem while training a deeper neural network. Skip connections bring good performance by reducing the vanishing gradient problem and by facilitating a good information flow across layers.

2.2.3. SqueezeNet

SqueezeNet was developed by researchers at DeepScale, the University of California, Berkeley, and Stanford University [35]. The motivation behind the development of the SqueezeNet model was to reduce the number of parameters. They succeeded to achieve AlexNet-level accuracy with fifty times fewer parameters. However, the accuracy of the SqueezeNet is much lower than the accuracy of WideResNets. SqueezeNets can achieve about 58% accuracy on the ImageNet dataset. WideResNets can achieve about 80% accuracy on the ImageNet dataset. Although the parameter was reduced drastically, people did not observe competitive accuracy compared to SOTA models.

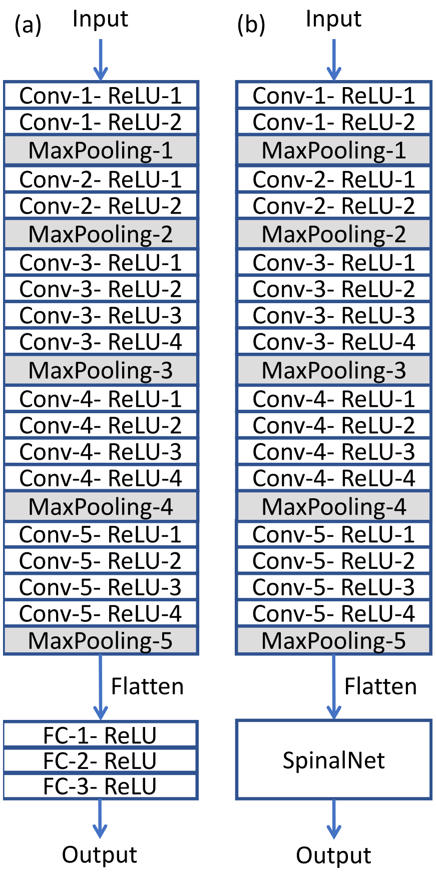

Fig. 3. Transfer learning with the modified SpinalNet fully connected layer on the VGG model. (a) The VGG-19 model. (b) VGG-19 model with SpinalNet fully connected layers.

2.2.4. Vision transformer

Vision transformer is a powerful transformer-type model that takes images as inputs. Transformer models have brought revolutionary performance in natural language processing [36]. Observing the remarkable success of transformers in natural language processing, Alexey et al. proposed the vision transformer, also known as (ViT) [37]. Alexey et al. split images into several fixed-sized patches and added position embeddings. They also added a trainable classification token. The transformer encoder consists of a multi-head attention network and a multi-level perceptron network. We have downloaded a pre-trained transformer model named *vit_large_patch16_224* through Pytorch Image Models (timm) library. That model takes 224 × 224 sized images and provides classification results. Fig. 4 presents the structure of the transformer. The transformer splits images into segments and linearly adds patch embeddings to each segment. There is one class-dependent trainable embedding. Embedded patches are sent to the transformer encoder. A multi-level-perceptron head receives the output of the encoder and predicts the class. Transferred initialization with this model has brought SOTA performance in several classification datasets [26,38]. The end layers of transformer-type models are known as the head. We replace the previous head with the traditional head, SpinalNet head, and SpinalNet shared weight head layers and compare results.

2.3. Transferred initialization

Transferred initialization is a powerful technique used to achieve promising performance within a short training time and with fewer training samples [25]. The transferred initialization method of training a neural network is quite similar to transfer learning. In transfer

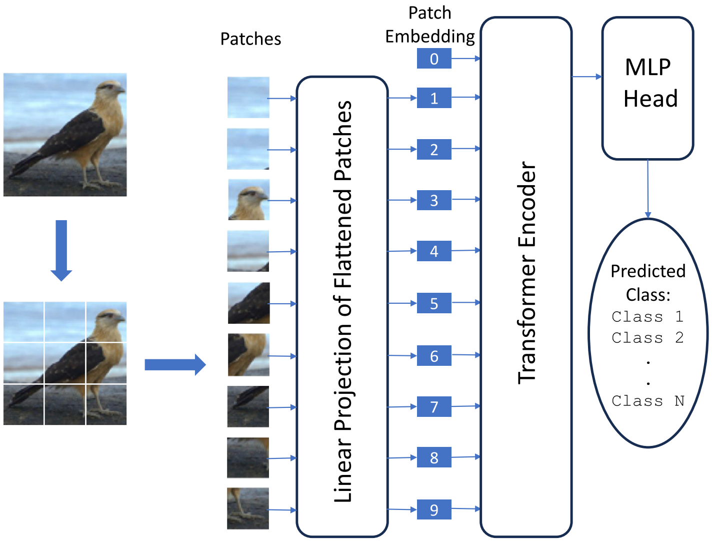

Fig. 4. The transformer architecture. This sketch is inspired by the paper [37].

learning, initial weights are frozen. However, in the transferred initialization, initial weights are not frozen. As a result, it becomes possible for the training algorithms to bring an optimization near the global minima. In transfer learning, it often happens that some important information is not propagated through the initial layers. The potential reason for the missing information is the difference between the pre-training dataset and the current dataset. The pre-training dataset may contain only natural images whereas the current dataset may contain grey-scale images or medical images.

We apply Adam [39] and SGD [40] optimizers in the NN training. Both of them are variants of stochastic gradient descent (SGD). The SGD method replaces the actual gradient with an approximation. The computational burden is significantly reduced in high-dimensional problems. In such optimization, the outcome greatly depends on the initialization. Moreover, initial layer weights get a small gradient during the training. Therefore, training initial layers with a large and similar dataset can potentially provide an overall good performance within a few training iterations. As initial layers of the NN get small gradients, initial layers remain almost the same and mid-layers get slight training. That often brings superior performance over traditional transfer learning, which only trains the end layers.

2.4. Uncertainty in neural network

Neural networks often exhibit high uncertainty in rare and critical samples. Neural networks are known as universal predictors. Theoretically, Neural Networks of sufficient size can capture any pattern. However, there exists uncertainty in datasets [41,42]. Moreover, the training set often lacks a sufficient number of critical patterns. As a result, both aleatory and epistemic uncertainty arises in the trained NN model [43]. In regressive problems, the *true regression mean* of the prediction system is expressed as follows:

$$y_j = t_j - \epsilon_j \tag{1}$$

where, y_j is the regression mean, t_j is the target of the jth sample, and ϵ_j is the error signal. That error signal can happen due to uncertainty in data or the model. That error affects both the regression type NNs and classification type NNs. Fig. 5 presents several confusing samples on the MNIST test dataset. Some samples on the dataset can be highly uncertain. Both well-trained NN and experienced humans can potentially be unable to predict those samples confidently.

The inherent randomness of the data is called the aleatoric and the model limitation is known as the epistemic uncertainty. The epistemic uncertainty is getting huge attention with the advancement of deep learning models. Epistemic uncertainty is often formulated over the distribution of model parameters. Bayes theorem is applied to posterior distribution to achieve the level of epistemic uncertainty. In Bayesian statistics, when an observer perceive event X_k, the probability of happening Y_k simultaneously is as follows [44]:

$$P(Y_k|X_k) = \frac{P(Y_k \cap X_k)}{P(X_k)} \tag{2}$$

where ∩ is the intersection sign. $P(.)$ is the probability function. When the fraction of X_k domain has in common with Y_k has a higher value than the overall probability of Y_k, the conditional probability of getting Y_k increases. Otherwise, the conditional probability of getting Y_k decreases.

Classification-type NNs contain a sharp decision boundary from output posteriors. Output posteriors are the numeric outputs before applying a sharp decision boundary of the class. In most situations, the class number becomes the index of the maximum value of the output posterior matrix. The class number equation is as follows:

$$C_j = MaxIndex(P_j) \tag{3}$$

where C_j is the predicted class number of jth sample.P_j is the predicted posterior matrix of jth sample. $MaxIndex(.)$ function returns the index of maximum value.

The Bayesian method is the most popular one in uncertainty quantification. The posterior probability results from updating the prior probability in the Bayesian method. However, in the proposed method, we name the numeric outputs of different classes as output posteriors. In the traditional classification method, the index of the maximum numeric output becomes the class number. However, in this method, we plot numeric posteriors obtained from different NNs on the same sample. Observing that posterior plot, medical practitioners can get an understanding of the level of opacity and make decisions.

Uncertainty can arise due to both data and model [45]. As handwritten digits are taught from our childhood, most of us are experts in MNIST digits. Still, we are confused about several MNIST samples. The same type of confusion may arise among medical specialists while detecting a novel disease. One solution can be looking at posteriors. In the first example, shown in Fig. 5, the output posterior of the model may have high values for both '2' and '7'. However, a sharp decision boundary makes the prediction '2'. Experts can potentially have an idea of the uncertainty based on the values of output posteriors.

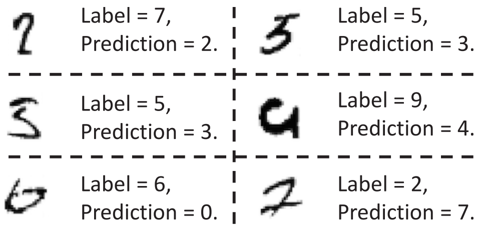

Fig. 5. Several common mistakes of AI models on the MNIST dataset. Many humans might get confused with those samples. Someone may also conclude that those samples are wrongly labeled or unconsciously written.

2.5. *The spinal cord and the SpinalNet*

The SpinalNet was developed by mimicking the human somatosensory system [25]. The SpinalNet tried to address the issue of the increased number of parameters due to a large input. The human body also takes a large input through our skin. Human skin can sense touch, heat, texture, vibration, wind flow, etc. All of these senses are sent to the human brain through the spinal cord. Fig. 1(a) presents a rough diagram showing rough connections of the human somatosensory system. Fig. 1(b) presents the structure of SpinalNet. Fig. 2 presents a portion of the human nerve plexus. Our nerves are also convoluted. There exist roots, trunks, divisions, cords, and branches of nerves. There exist connections between parallel nerves. Fig. 2 sketches a few of them. SpinalNet was developed by allowing gradual and repetitive input by mimicking that connection. Although it seems in Fig. 1 that the structure is quite different from traditional structures, the proposed structure is quite similar to NNs with skip connections. A portion of the input is going to the first hidden layer and the same input is going to a later hidden layer.

2.6. *Proposed SpinalNet shared weight structure*

Fig. 1(c) presents the structure of SpinalNet with shared weights. In Fig. 1(c), connections with shared weights are drawn as light blue color. Except for the first layer in the intermediate split, we take the split of inputs for the corresponding layer and concatenate them with the outputs of the previous layer's intermediate split and apply the shared fully connected layer. The SpinalNet takes inputs gradually and repetitively and a narrow intermediate split reduces the number of multiplication.

The shared weights in Fig. 1(c) are the weights, containing the same value. Each layer from the second hidden layer to the last hidden layer has the same number of inputs and outputs. The input size is the summation of the number of inputs in a split, and the number of outputs from the previous hidden layer. Output size is the number of outputs in the current hidden layer. We apply the same parameter values to all these hidden layers. The concept is similar to weight sharing in convolutional layers.

The number of parameters becomes lower than the traditional fully connected layers and the original SpinalNet layer in the proposed method. The first hidden layer of the NN model contains a large number of parameters in the traditional method. The first hidden layer contains weights and biases. The number of weights is equal to the multiplication of the number of inputs and the number of first hidden layer parameters. The number of biases is equal to the number of the first hidden layer parameters. The parameter reduction is possible with a narrow first hidden layer. However, a narrow first hidden is unable to propagate all important features from the input layer to the second hidden layer, resulting in a degraded overall performance of NN. The SpinalNet takes a small portion of the input to the first hidden layer. In Fig. 2(b), $X[1 : k]$ is the portion of the input that is going to the first hidden layer. $X[1 : k]$ is a portion of X and the first hidden layer can be narrower, as we are taking inputs repetitively. When $X[1 : k]$ contains half a portion of X and the size of the hidden layer becomes half of the traditional hidden layer, the number of weights on that layer becomes one-fourth of the traditional hidden layer. However, the original SpinalNet model takes inputs both gradually and repetitively to achieve improved performances. Therefore, the number of parameters becomes slightly higher than a single hidden layer NN. In the proposed method, input-split to intermediate hidden layer parameters share a common value. Therefore, although inputs are repeated, the number of parameters from the input to the intermediate layer does not increase. As a result, the number of parameters becomes lower than both the original SpinalNet and single hidden layer NN.

While any parameter squeezing is proposed on the end layers, there can be potential loss of information. However, we may get good results in many datasets. The SqueezeNet also got very good results with parameter reduction [35]. However, they did not provide any theoretical proof. Similarly, we investigate our parameter reduction method on multiple datasets to prove the concept.

3. Investigated datasets and augmentations

This section presents investigated datasets, their augmentations with reasons, and effects of augmentations. Augmentation is a good way to increase the robustness of NNs. It prevents NN from being overfitted. The augmentation increases the number of training samples. When a random rotation augmentation is applied to images, the NN receives training on both original and rotated images. As a result, the NN becomes capable of noticing any rotation-related difference between the training and the test data. We investigate the following datasets by training NNs with augmentations.

3.1. *MNIST*

The Modified National Institute of Standards and Technology (MNIST) dataset [46] is a handwritten digit classification dataset. It is the most used dataset in image classification. The major reason for its popularity is its simplicity and low memory requirement. The MNIST dataset contains seventy thousand images. Among those images, ten thousand images are test images and sixty thousand images are training images. Images are 28×28 sized grayscale images. Images are collected from participants in several American high schools. A well-trained machine learning model can receive 99.7%+ accuracy on this dataset. Receiving 99.8%+ accuracy requires both a very good model and an extremely lucky training session. The reason for that limitation is the variation in the writing of humans. Some participants' handwritten digit four is quite similar to another participant's nine. Both machines and humans can get confused by such highly uncertain images.

On the MNIST dataset, we perform the following augmentations: (i) random rotation, (ii) random shift, and (iii) random perspective. The handwritten characters are not augmented for horizontal or vertical flips. Flips do not happen on handwritten digits or characters unless someone has dyslexia. Moreover, the flip or 180-degree rotation of a sample can potentially make the sample belong to another class. Fig. 6(a) presents thirty-two representative samples of the MNIST dataset without augmentations. Fig. 6(b) presents those representative samples with augmentations.

3.2. *STL-10*

STL-10 [47] is one of the most popular datasets in image vision. The Stanford AI Lab proposed that dataset by observing the CIFAR-10 dataset. The CIFAR-10 dataset contains 32×32-sized RGB images. The STL-10 dataset contains 96×96-sized RGB images, labeled as ten classes. Each class contains five hundred training images and eight hundred test images. Classes are `truck`, `ship`, `monkey`, `horse`, `dog`,

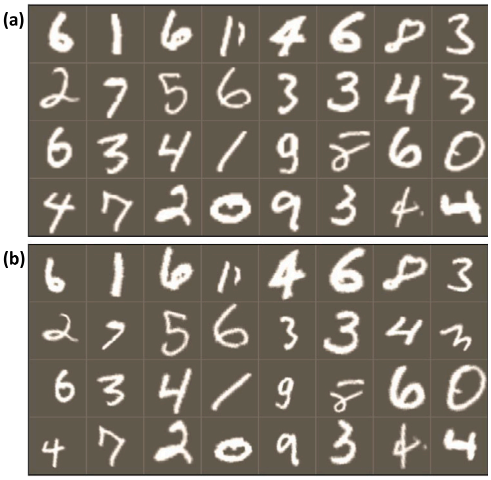

Fig. 6. Typical samples of the handwritten digit (MNIST) dataset. (a) After normalization and without augmentations; (b) after augmentations.

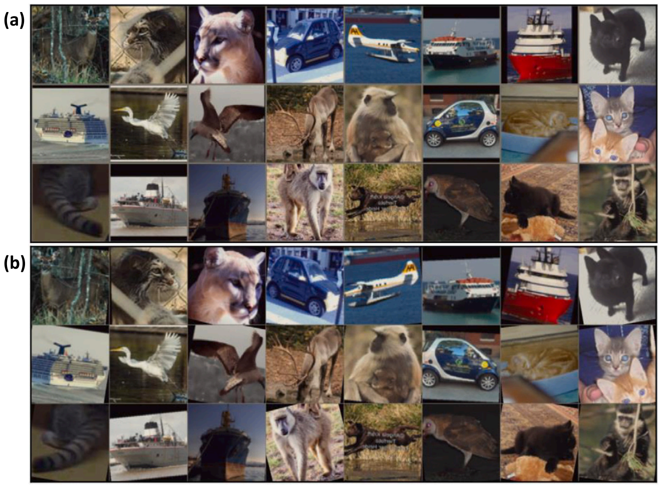

Fig. 7. Typical samples of the STL-10 dataset. (a) After normalization and without augmentations; (b) after augmentations.

`deer`, `cat`, `car`, `bird`, and `airplane`. Images are a small subset of the ImageNet dataset.

On the STL-10 dataset, we perform the following augmentations: (i) random rotation, (ii) random crop, and (iii) random horizontal flip. We do not perform some other augmentations, such as random perspective, and random vertical flip. Fig. 7(a) presents twenty-four representative samples of the STL-10 dataset without augmentations. Fig. 7(b) presents those representative samples with augmentations.

We investigate the effect of changing parameters on the STL-10 dataset. Fig. 8 presents the accuracy vs the number of trainable parameters on the Head layer plot on the STL-10 dataset. The accuracy and parameter count of the traditional model is represented by the blue star. The black curve presents the SpinalNet model. The green curve presents the SpinalNet Shared Weight model. Red, green, blue, and black circles represent a hidden layer width of 10, 20, 50, and 100 respectively. The traditional model has 1024 trainable Head parameters but the average accuracy is 99.50%. The proposed model can receive almost the same accuracy on average when the hidden layer width is 10. The number of parameters becomes 3490 for a hidden layer width of 10. The parameter reduction becomes possible by maintaining the same average accuracy over the traditional model. However, we planned to achieve both parameter reduction and accuracy improvement over the traditional model.

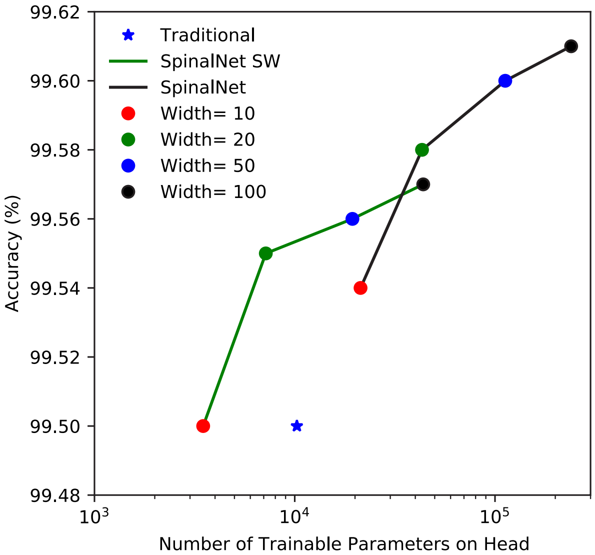

Fig. 8. Accuracy vs the number of trainable parameters on the Head layer plot on the STL-10 dataset. The black curve presents the SpinalNet model. The green curve presents the SpinalNet Shared Weight model. The accuracy and parameter count of the traditional model is represented by the blue star. Red, green, blue, and black circles represent a hidden layer width of 10, 20, 50, and 100 respectively.

We achieve both improved accuracy and parameter reduction over the traditional model when the number of hidden layer neurons is double the number of classes. STL-10 data contains 10 classes. The hidden layer width of 20 brings higher accuracy and parameter reduction with the proposed model. However, SpinalNet brings higher accuracy but costs much higher parameters. The proposed model with a width of 20 outperforms SpinalNet of a 10 width in terms of both accuracy and parameter count. However, if we increase the width of the proposed model to 100 it brings inferior performance compared to the SpinalNet of a 20 width in both categories. Therefore, we conclude that a width equal to twice the class number brings optimal performance in terms of both accuracy and parameter count.

3.3. SIIM-FISABIO-RSNA COVID-19 detection

After investigating the effectiveness of the proposed DNN model in different types of datasets, we apply a popular coronavirus disease 2019 (COVID-19) dataset, proposed by Society for Imaging Informatics in Medicine (SIIM) to the model. The title of the dataset is “SIIM-FISABIO-RSNA COVID-19 Detection” [48,49]. The dataset contains images in the Digital Imaging and Communications in Medicine (DICOM) format. However, available high-performing pre-trained models are developed using RGB image datasets. We convert DICOM images to RGB format images of 512 × 512-size. First, we convert DICOM images to grayscale images. Then, the Sobel filter is applied to grayscale images to achieve magnitude and edges [50]. The RGB image contains magnitude, edge magnitude, and edge angles of pixel intensities, in R, G, and B layers respectively. We perform data pre-processing in a Kaggle notebook[1] and share the script for transparency.

We perform the following augmentations on the pre-processed Chest X-ray (CXR) images: (i) random rotation, (ii) random crop, (iii) random perspective, and (iv) random horizontal flip. Fig. 9(a) presents sixteen representative samples of the COVID-19 dataset without augmentations. Fig. 9(b) presents those representative samples with augmentations. We do not perform random vertical flip augmentation. Vertical

[1] https://www.kaggle.com/dipuk0506/dicom-files-to-rgb-mag-edge1-edge2-512x512

Table 1
Effect of different augmentations while performing transfer learning from VGG-19 neural network. Besides augmentations, mentioned in this table, we also apply random horizontal flips, resizing, and normalization for all learning combinations.

Random Grayscale	Random Perspective	Random Rotation	Mean Train Accuracy (%)	Mean Validation Accuracy (%)	Mean Test Accuracy (%)	Test Accuracy Variation (%)
×	×	×	77.35%	74.42%	72.86%	3.06%
×	×	✓	77.67%	74.92%	74.02%	2.12%
×	✓	×	77.91%	75.06%	73.99%	2.39%
×	✓	✓	79.02%	77.29%	**76.37%**	1.42%
✓	×	×	75.82%	73.02%	71.87%	2.59%
✓	×	✓	77.95%	75.13%	73.97%	2.13%
✓	✓	×	77.98%	75.09%	73.93%	2.26%
✓	✓	✓	78.16%	75.28%	74.41%	2.01%

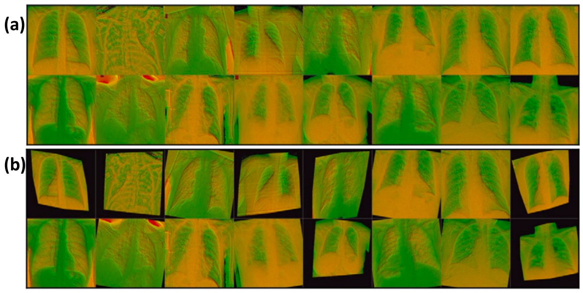

Fig. 9. Typical samples of the SIIM COVID dataset. (a) After pre-processing, normalization, and without augmentations; (b) after augmentations.

flip in CXR images does not happen in reality unless someone makes an erroneous entry.

4. Results

4.1. Selection of augmentations

Different augmentations bring optimal performance for different kinds of datasets [51]. Some augmentations bring a better performance. Random horizontal flips of natural images increase the number of samples. The augmentation increases the validation and test accuracies of NN. However random vertical flips do not happen in natural images. Unless the dataset contains some wrong entries. Applying random vertical flip augmentation adds some irrelevant samples. As a result, the training time of NN increases, and the accuracy of NN decreases. Some augmentations, such as random grayscale and random perspective have mixed effects on different datasets. Therefore, we perform trial and error to achieve optimum augmentations. MNIST and STL-10 are two well-known datasets. We have achieved optimal augmentation combinations for training NNs in our previous papers [16,25]. Table 1 presents the effect of different augmentation combinations on the SIIM COVID dataset. We trained each DNN model ten times to achieve average results. According to this table, random rotation and random perspective improve the overall accuracy. We observed that models providing higher accuracies also have slightly lower variations. The most optimum performance with the VGG-19 model is indicated in bold letters.

4.2. Results on MNIST dataset

We have investigated VGG, ResNet, and transformer-type models on this dataset to evaluate our proposed model. With the VGG-19 model, we received similar accuracies with fewer parameters. The fully-connected portion of the VGG networks contains a large number of parameters. SpinalNet contains fewer parameters and can provide superior accuracy. SpinalNet with shared weight brings further reduction in the number of parameters. The number of parameters is reduced from 119.59M to 7.56M. The first segment of Table 2 presents the results of different models obtained using the MNIST dataset. Here, the 'M' sign represents a million and the 'k' sign in the table represents a kilo or thousand. WideResNet-101 transfer learning can provide good accuracy using both color and grey-scale images. We achieved both higher accuracy and parameter reduction over the traditional WideResNet-101 model using the SpinalNet shared weight. Models are trained over twenty epochs to achieve results. The learning rate is set to 0.01 for the first ten epochs and then the learning rate is set to 0.001 for the next ten epochs. Momentum is kept at 0.9, the step size is set to seven, and gamma is set to 0.1. We have achieved slightly lower accuracy with significant head-layer parameter reduction using the proposed model compared to SpinalNet. We have achieved both parameter reduction and improved accuracy over the traditional head. We train transformers with a learning rate of 1e−4.

The confusion matrix represents an indication of the heteroscedastic uncertainty of the model. Fig. 10 is the confusion matrix obtained for the MNIST dataset. According to the figure, '9' and '4' are confusing to NNs. '4' is predicted as '9', thrice, and '9' is predicted as '4', four times. There is also a high uncertainty between '2' and '7'. Also, we present the heteroscedastic performance of trained models in Table 3. The first segment of Table 3 shows the results of an example model on the MNIST dataset. The table presents the precision, recall, and F1 score values of each class for all datasets. When a sample is predicted as class A, the probability of the sample being labeled to class A is the precision. When a sample is labeled as class A in the dataset, the probability of the sample being predicted to class A is the recall. The F1 score is the harmonic mean of precision and recall.

4.3. Results on STL-10 dataset

We have investigated VGG, ResNet, and transformer-type models on this dataset. With the VGG-19 model, we also received similar accuracies with a much lower number of parameters on the STL-10 dataset. Parameter reductions are also the same as the MNIST dataset. The second segment of Table 2 presents the results of different models

Table 2
Performance of the VGG-19 and WideResNet-101 with normal, spinal, and spinal-shared-weight fully connected layers.

Data	Model	Parameters in			Number	Test Accuracy		Error
		Fully Connected Layer			of	Average	Best	Reduction
		Split	Mid-Layer	Count	Epoch			(Best)
MNIST [46]	VGG-19_bn [34]	–	–	119.59M	20	99.69%	99.74%	–
	VGG-19_bn (Spinal FC) [25]	2	1024	54.57M	20	99.70%	99.75%	3.8%
	VGG-19_bn (Spinal FC SW)	8	1024	**7.56M**	20	99.70%	99.74%	0.0%
	WideResNet-101_2 [52]	–	–	20.49k	20	99.52%	99.62%	–
	WideResNet-101_2 (Spinal FC) [25]	8	20	45.53k	20	99.66%	99.77%	39.5%
	WideResNet-101_2 (Spinal FC SW)	8	20	**12.29k**	20	99.68%	99.72%	26.3%
	ViT-L/16 [38]	–	–	10.25k	2	99.72%	99.75%	–
	ViT-L/16 (Spinal FC) [38]	8	20	25.05k	2	99.75%	99.76%	4.0%
	ViT-L/16 (Spinal FC SW)	8	20	**7.17k**	2	99.74%	99.75%	0.0%
STL-10 [47]	VGG-19_bn [34]	–	–	119.59M	15	94.97%	95.44%	–
	VGG-19_bn (Spinal FC) [25]	2	1024	54.57M	15	95.03%	95.57%	2.9%
	VGG-19_bn (Spinal FC SW)	8	1024	**7.56M**	15	95.01%	95.57%	2.9%
	WideResNet-101_2 [52]	–	–	20.49k	15	97.83%	98.40%	–
	WideResNet-101_2 (Spinal FC) [25]	8	20	45.53k	15	98.23%	98.66%	16.3%
	WideResNet-101_2 (Spinal FC SW)	8	20	**12.29k**	15	98.05%	98.45%	3.1%
	ViT-L/16 [38]	–	–	10.25k	2	99.50%	99.61%	–
	ViT-L/16 (Spinal FC) [38]	8	20	25.05k	2	99.58%	99.70%	23.1%
	ViT-L/16 (Spinal FC SW)	8	20	**7.17k**	2	99.55%	99.64%	7.7%
COVID by SIIM [49]	VGG-19_bn [34]	–	–	119.55M	10	76.52%	76.91%	–
	VGG-19_bn (Spinal FC) [25]	8	512	381.0k	10	76.23%	77.74%	3.6%
	VGG-19_bn (Spinal FC SW)	8	512	**29.2k**	10	76.37%	77.41%	2.2%
	WideResNet-101_2 [52]	–	–	4.1k	10	77.26%	78.84%	–
	WideResNet-101_2 (Spinal FC) [25]	8	4	8.4k	10	77.88%	80.01%	5.5%
	WideResNet-101_2 (Spinal FC SW)	8	4	**2.14k**	10	78.11%	79.82%	4.6%
	ViT-L/16 [38]	–	–	2.05k	2	82.69%	85.03%	–
	ViT-L/16 (Spinal FC) [38]	8	4	4.31k	2	84.81%	86.88%	12.4%
	ViT-L/16 (Spinal FC SW)	8	4	**1.11k**	2	82.97%	85.09%	0.4%

on the STL-10 dataset. We achieved higher accuracy and parameter reduction over the traditional WideResNet-101 model on the STL-10 dataset using the SpinalNet shared weight. Models are trained over fifteen epochs to achieve results. The learning rate is set to 1e-3 for the first five epochs and then the learning rate is set to 1e-4 for the next ten epochs. Momentum is kept at 0.9, the step size is set to seven, and gamma is set to 0.1. We have achieved slightly lower accuracy with significant head-layer parameter reduction using the proposed model compared to SpinalNet. We have achieved both improved accuracy and parameter reduction over the traditional head. We train transformers with a learning rate of 1e-4.

Fig. 11 presents the confusion matrix obtained using the STL-10 dataset. The model has obtained the highest accuracy in classifying ships. The model accurately predicted 798 samples out of eight hundred ship samples. The `airplane` class also obtained the same accuracy. The model often gets confused with `car` and `truck`. Distinguishing `deer`, `dog`, and `horse` classes are often troublesome for NN models. Also, we presented the heteroscedastic performance using trained models in Table 3. The second segment of Table 3 shows the results of an example model on the STL-10 dataset.

4.4. Results on SIIM COVID dataset

We have applied both the VGG-19 and WideResNet-101 pre-trained models from PyTorch Torchvision model libraries. Also, we have applied *vit_large_patch16_224* model, downloaded through Pytorch Image Models (timm) library. As the SIIM COVID dataset contains DICOM images, we converted those images to RGB images. Section 3.3 presents details of the conversion and augmentations. We receive both performance enhancement and parameter reduction over traditional models. The third segment of Table 2 presents the performances of models obtained using the SIIM COVID dataset. The proposed model outperforms the traditional head in terms of both accuracy and parameter count. Although the SpinalNet provides slightly higher performance, SpinalNet contains a significantly large number of parameters in the head layer. The users of the model have two options. If the user wants the highest performance, he can use the SpinalNet model. When the user has limited computing ability, he can use the SpinalNet Shared Weight model.

Fig. 12 presents the confusion matrix of a well-trained WideResNet transfer learned model on SIIM COVID data. Also, we present the heteroscedastic performance of trained models on COVID data in Table 3. This table presents the precision, recall, and F1 score values obtained for each class of all datasets.

4.5. Parameter reduction in proposed model

As inputs are split and neuron per layer is reduced, weights in layers decrease significantly. For example, the VGG-19 model has 25088 inputs to the classifier layer group. This results in a large number of parameters in the first fully connected hidden layer. As the first fully connected hidden layer input contains 4096 × 25088 number of weights. Two hidden layers of 4096 size also introduce 4096 × 4096 number of weights between two hidden layers. However, when SpinalNet is applied as the fully connected layer of the VGG-19 model, SpinalNet fully connected layer splits the input into two parts and makes the number of hidden neurons for each layer one-fourth (1024).

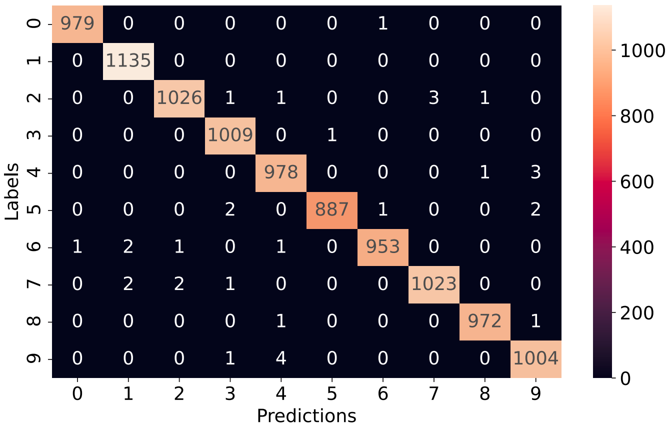

Fig. 10. An example confusion matrix on the MNIST test dataset. The accuracy observed in this figure is 99.66%.

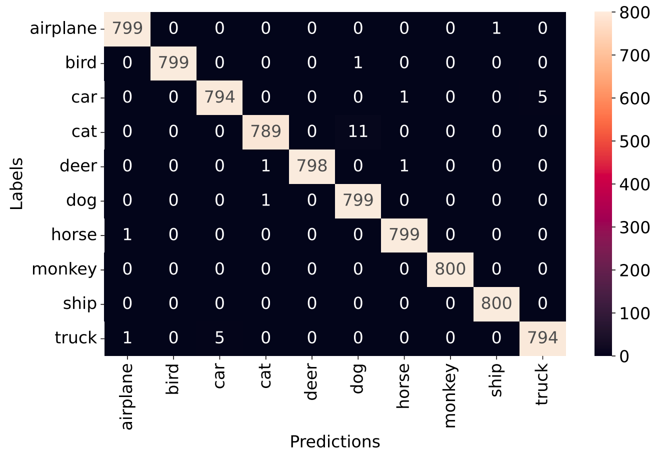

Fig. 11. An example confusion matrix on the STL-10 test dataset. The accuracy observed in this figure is 99.64%.

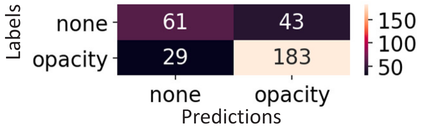

Fig. 12. An example confusion matrix on the SIIM COVID test dataset. The accuracy observed in this figure is 77.22%.

Therefore, the number of weights in the first hidden layer becomes one-eighth. The second hidden layer receives a split of the inputs and outputs of the first hidden layer. When the splitting is two, the second hidden layer receives half of the inputs and outputs of the first hidden layer. Therefore, incoming weight to the second hidden layer becomes $(25088/2 + 1024) \times 1024$. Two hidden layers together contain much lower parameters than the first classifier layer of the traditional VGG classifier. Moreover, in the proposed SpinalNet shared weight model, input weights of the second hidden layer to the last hidden layer contain a set of shared values. That further reduces the number of parameters. WideResNet-101 has 2048 features and the transformer model has 1024 features. When features are divided, and weights are shared the number of parameters becomes significantly lower.

4.6. Uncertainty of samples

The COVID-19 dataset contains highly uncertain samples. Although the F1 score of models on other investigated datasets is more than 97%, the F1 score for opacity detection is about 83.5%. Therefore, medical specialists cannot conclude with a highly certain result based on the output of models. Therefore, we provide a plot of the output posterior with the depiction of a sharp decision boundary. Highly trained models make mistakes in critical situations. Fig. 5 presents such critical situations. In such a situation, a sample may seem to be a member of more than one class. In a critical situation, both classes can potentially have a high value on the posterior. Therefore, we investigate the location of output posteriors compared to the decision boundary in Fig. 13. Subplot (a) of Fig. 13 shows the location of output posteriors, where the sample has the `None` class label. Subplot (b) of Fig. 13 shows the location of output posteriors, where the sample has the `Opaque` class label. Posteriors of `None` predicted samples stay below the margin line and posteriors of `Opaque` predicted samples stay over the margin line. We draw samples based on labels to observe the location of posteriors while the prediction is wrong.

According to Fig. 13, there is a weak positive correlation between the label and values of posteriors. The majority of the samples stay in the corresponding domain. Some samples stay in the wrong domain. However, a few `None` samples stay on the extremely wrong end, and no `Opaque` sample stays on the extremely wrong end. Therefore, it is possible to find a clearer view of the uncertain situation by seeing the posterior; most of the situation. A medical specialist can also see the posteriors of a sample derived from multiple models to observe a closer uncertain situation.

4.7. Performance variation among datasets

Different datasets have different levels of uncertainty. Therefore, performances are also different. It is possible to achieve more than 99.5% accuracy on the MNIST dataset. However, the STL-10 dataset has much higher uncertainty. The SOTA performance on the STL-10 dataset was less than 98% before 2020. In the year 2020, the WideResNet-101(Spinal FC) model obtained 98.66% accuracy with the transfer learning [25]. WideResNet-101 model is pre-trained on the ImageNet training dataset and is publicly available by the Torchvision module. In 2022, a group of researchers from Google Research reported a new SOTA accuracy of 99.64% through multitask learning technique [53]. The SIIM COVID dataset has a much higher level of uncertainty. According to our investigation, Transfer-learned models can achieve about 80% accuracy on that dataset.

4.8. Performance variation among different classes

There exist similarities among different classes, and those similarities create uncertainties. The precision of a class-A is the accuracy of the model on samples predicted as that class. The recall of a class-A is the model accuracy on samples labeled as that class. Both the precision and recall become lower due to the similarity between classes. In the MNIST dataset, the ‘0’ class has the highest precision and the ‘1’ class has the highest recall. The ‘4’ has the lowest precision and the ‘5’ class has the lowest recall. This happens due to similarities in hand-written digits and variations in writing among people. For example, some people confusingly write ‘4’ and some other people consider that writing as ‘9’. Machine learning models also get confused when they are trained with the writings of thousands of different people. In STL-10, the class `airplane` has the highest precision. The class `ship` has the highest recall. The class `deer` has the lowest precision. The class `horse` has the lowest recall. That happens due to similarity among classes. Non-living classes of STL-10 datasets are quite dissimilar to living classes. However, classes containing quadrupeds are more similar. Due to their different pose and variation in training and testing datasets, the ML model can potentially get confused. Humans have higher experience than machine learning models in classifying animals. Also, humans usually pay more attention to the unique characteristics of animals instead of their poses.

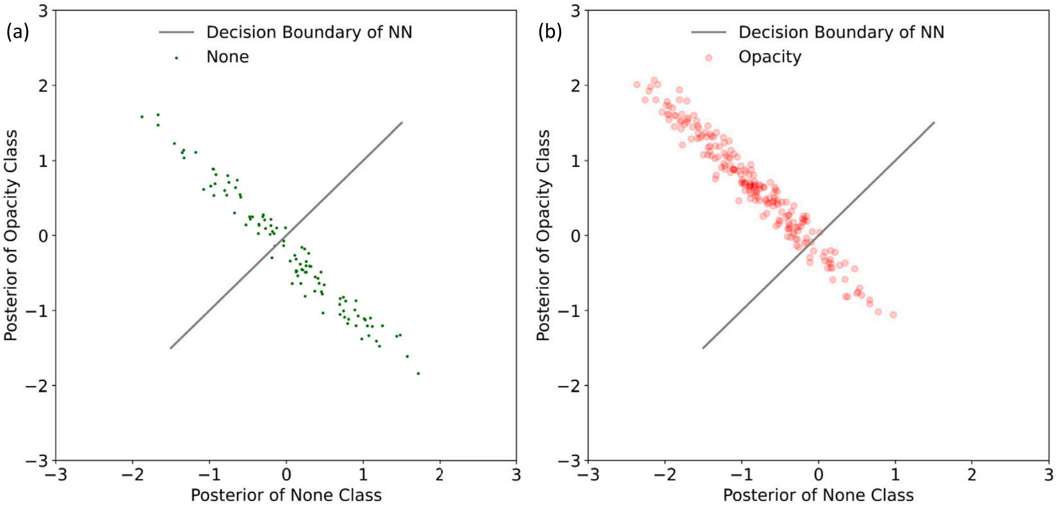

Fig. 13. Posteriors of samples in the test dataset. Subplot (a) shows posteriors of samples labeled as None, (a) shows posteriors of samples labeled as Opaque at the test dataset. A strict decision boundary cannot determine the level of uncertainty on the prediction. The distance from the decision boundary can be an indicator of uncertainty. However, NNs can also be wrong with high confidence in some samples.

Table 3
Classwise performance of NNs with various datasets. Numeric values in this table are generated from the example confusion matrix, shown in Figs. 10, 11, and 12.

Data	Class	Precision	Recall	F1 Score
MNIST	0	0.99898	0.99898	0.99898
	1	0.99649	1.00000	0.99824
	2	0.99708	0.99419	0.99563
	3	0.99507	0.99901	0.99704
	4	0.99289	0.99593	0.99441
	5	0.99887	0.99439	0.99663
	6	0.99791	0.99478	0.99634
	7	0.99708	0.99514	0.99611
	8	0.99795	0.99795	0.99795
	9	0.99406	0.99504	0.99455
STL-10	airplane	0.99874	0.99250	0.99561
	bird	0.99747	0.98500	0.99119
	car	0.98618	0.98125	0.98371
	cat	0.97264	0.97750	0.97506
	deer	0.96683	0.98375	0.97522
	dog	0.96782	0.97750	0.97264
	horse	0.99104	0.96750	0.97913
	monkey	0.98625	0.98625	0.98625
	ship	0.98885	0.99750	0.99315
	truck	0.97640	0.98250	0.97944
COVID	None	0.67778	0.58654	0.62887
by SIIM	Opaque	0.80973	0.86321	0.83562

4.9. Prediction uncertainty of a sample

There exist several representations of uncertainty. The prediction interval is the most popular form of presenting uncertainty in regression problems. Researchers also provide heteroscedastic variance value as an indication of uncertainty [54]. Users get a rough understanding of uncertainty from precision, recall, and confusion matrix. The user gets an idea of the probability of misclassification from precision. The user also gets an idea of other potential classes as the original label when the prediction is wrong based on the confusion matrix. However, all samples predicted as Opaque do not have the same level of uncertainty. Fig. 14 presents a posterior plot of a sample labeled Opaque. We train ten WideResNet-101_2 (Spinal FC SW) models and obtain predictions. Eight models predict the sample as Opaque and two models predict the sample as None. Moreover, the location of posteriors in None prediction is very close to the decision boundary. Observing those posteriors, the medical practitioner can easily predict that the sample is Opaque.

In this paper, we consider the Precision, Recall, and F1 Score of each class, overall accuracy, and confusion matrix as indications of uncertainties. Also, we provide a 2D plot of output posteriors to the medical practitioners. By seeing the posterior plot, the medical practitioner may get a better understanding of the level of uncertainties.

There exist several other uncertainty quantification matrices. Several researchers have considered the distribution of classification results

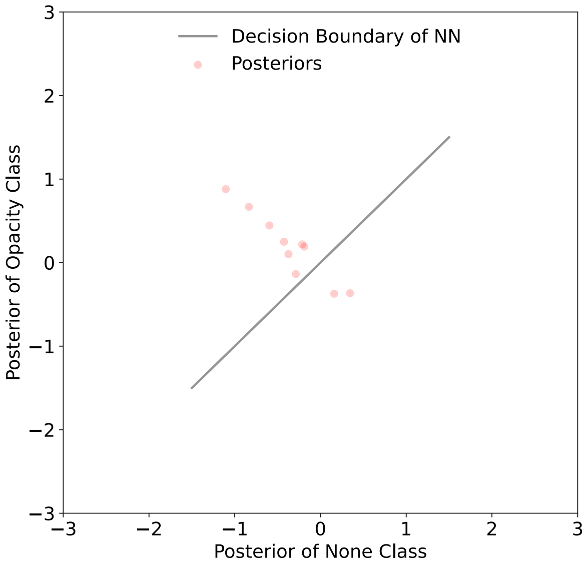

Fig. 14. Output posterior of an `Opaque` labeled sample on the SIIM COVID dataset. The sample has StudyInstanceUID of `39a80a14bfda`. We train ten WideResNet-101_2 (Spinal FC SW) models and obtain predictions. Eight models predict the sample as `Opaque` and two models predict the sample as `None`. Moreover, the location of posteriors in `None` prediction is very close to the decision boundary.

for a sample [55] as an indicator of the level of uncertainty. The distribution of classification results contains indexes of the maximum value. A slight difference between two output posterior can alter classification results while considering only the maximum value. The posterior plot in the proposed method presents the numeric value of posteriors. Several other researchers use *Label stability* as a measure of the level of uncertainty [55]. The *Label stability* is the absolute difference between two numbers: (i) the number of times the sample is predicted as positive, and (ii) the number of times the sample is predicted as negative. This approach also faces the limitations of a sharp boundary. A recent paper [16] proposed uncertainty scores based on k-nearest neighbors output posterior. That approach overcomes the limit of sharp boundaries. However, a few unusual samples in the neighborhood may result in a poor prediction. Therefore, showing the position of the posterior provides a deeper understanding of the level of uncertainty.

4.10. Comparison with the state-of-the-art performance

We achieve near SOTA performance in several datasets while applying the proposed head on the transformer. As we significantly reduce the number of parameters, the performance is always slightly lower than the SpinalNet. The SOTA accuracy on the STL-10 dataset is 99.71%. We have achieved 99.64% top accuracy with SpinalNet shared weight. That is equal to the previous SOTA on this dataset [53]. With the SpinalNet, we have achieved 99.70% accuracy. As MNIST is a highly investigated dataset, the best performance on this model has come due to a lucky session [56,57]. We have achieved near SOTA performance in the MNIST dataset. We have also investigated SpinalNet shared weight on CIFAR-10 and CIFAR-100 datasets and received 99.01% and 93.12% accuracies respectively. Current SOTA performances on these datasets are 99.50% [37] and 96.08% [58] respectively.

5. Notes and conclusion

In this paper, we have presented a parameter-optimized transfer-learned classification model with uncertainty awareness. We have proposed SpinalNet with shared weights. Also, we have prescribed health practitioners to observe the overall precision, recall, accuracy, and a posterior plot so that they can get an idea of the associated uncertainty. During the discussion with colleagues, we noticed that readers can potentially get several common questions. Probable questions and answers are as follows:

How medical practitioners can comprehend posterior plots: When a sample is `Opaque` most of the posteriors usually stay in the `Opaque` region. Some posteriors can potentially stay in the `Opaque` regions by maintaining a large distance from the decision boundary. A few posteriors can potentially stay in the `None` region. However, those posteriors usually stay close to the decision boundary.

How the aleatoric uncertainty is captured: Aleatoric uncertainty is the inherent randomness of the sample [7]. Although two samples may belong to the same class in terms of both labels and predictions, the level of opacity might not be the same. One sample may have a small region with opacity while another sample may have a larger region with opacity. The average position from the distribution of posteriors indicates the level of aleatoric uncertainty. Medical practitioners can get an idea of the level of aleatoric uncertainty from the median position of posteriors.

How the epistemic uncertainty is captured: Epistemic uncertainty is the error happening in models [7]. Different models predict the sample differently. When the epistemic uncertainty is low, there exists a low variance among posterior values. When the epistemic uncertainty is high, there exists a high variance among posterior values. Medical practitioners can get a rough estimation of epistemic uncertainty by seeing the distribution of posteriors.

How the Heteroscedasticity of uncertainty is captured: The uncertainty is heteroscedastic [7]. The level of uncertainty can be different for two samples. In regression problems, the width of the prediction interval varies from sample to sample. In classification, two samples with the same label or two samples with the same prediction may have a different level of uncertainty. When two images are slightly different, the posterior distribution also becomes slightly different. When the level of epistemic uncertainty is high, the variation among posterior increases. When the level of aleatoric uncertainty is high, the median of posteriors gets closer to the decision boundary. Therefore, the proposed method is capable of representing the heteroscedasticity of both types of uncertainties.

Discussion with Doctors: In this paper, we proposed a posterior plot to help medical practitioners. Therefore, we invited several doctors to review the posterior plot. According to doctors, binary results of Healthy/Diseased contain insufficient information. According to Dr. Sadia Khanam from Dhaka Dental College and Dr. Farjana-Binte-Habib from Dept of Microbiology, Dhaka Medical College, posterior plots are the best option. According to Dr. Sayem Sorwer Bappy from Mymensingh Medical College Hospital, observing numeric values of posteriors is the most convenient option.

The uncertainty-aware classification and COVID-19 methodology proposed in this paper can potentially help future researchers, statisticians, engineers, and medical practitioners. Future researchers can apply the proposed NN to similar datasets to achieve superior performance. However, this method provides posterior plots instead of any numeric score. Medical practitioners may need knowledge and experience in understanding these plots. Future researchers may develop scores based on the values of posteriors. Example training scripts are shared at the following GitHub repository: github.com/dipuk0506/SpinalNet

CRediT authorship contribution statement

H.M. Dipu Kabir: Writing – review & editing, Writing – original draft, Resources, Project administration, Methodology, Data curation, Conceptualization. **Subrota Kumar Mondal:** Writing – review & editing, Investigation. **Syed Bahauddin Alam:** Writing – review & editing. **U. Rajendra Acharya:** Writing – review & editing, Supervision, Project administration, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

We used publicly available data.

Acknowledgments

We would like to thank Dr. Sadia Khanam from Dhaka Dental College, Dr. Farjana-Binte-Habib from Dept of Microbiology, Dhaka Medical College, and Dr. Sayem Sorwer Bappy from Mymensingh Medical College Hospital for reviewing posterior plots.

References

[1] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[2] W. Zhang, C. Li, G. Peng, Y. Chen, Z. Zhang, A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load, Mech. Syst. Signal Process. 100 (2018) 439–453.

[3] S. Wang, S. Suo, W.-C. Ma, A. Pokrovsky, R. Urtasun, Deep parametric continuous convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2589–2597.

[4] F. Khozeimeh, D. Sharifrazi, N.H. Izadi, J.H. Joloudari, A. Shoeibi, R. Alizadehsani, J.M. Gorriz, S. Hussain, Z.A. Sani, H. Moosaei, et al., Combining a convolutional neural network with autoencoders to predict the survival chance of COVID-19 patients, Sci. Rep. 11 (1) (2021) 1–18.

[5] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90.

[6] H. Asgharnezhad, A. Shamsi, R. Alizadehsani, A. Khosravi, S. Nahavandi, Z.A. Sani, D. Srinivasan, S.M.S. Islam, Objective evaluation of deep uncertainty predictions for Covid-19 detection, Sci. Rep. 12 (1) (2022) 1–11.

[7] H.D. Kabir, A. Khosravi, M.A. Hosen, S. Nahavandi, Neural network-based uncertainty quantification: A survey of methodologies and applications, IEEE Access (2018).

[8] R. Theisen, H. Wang, L.R. Varshney, C. Xiong, R. Socher, Evaluating state-of-the-art classification models against Bayes optimality, 2021, arXiv preprint arXiv:2106.03357.

[9] S. Kokyay, E. Kilinc, F. Uysal, H. Kurt, E. Celik, M. Dugenci, A prediction model of artificial neural networks in development of thermoelectric materials with innovative approaches, Eng. Sci. Technol. Int. J. 23 (6) (2020) 1476–1485.

[10] D. Kumar, M. Marchi, S.B. Alam, C. Kavka, Y. Koutsawa, G. Rauchs, S. Belouettar, Multi-criteria decision making under uncertainties in composite materials selection and design, Compos. Struct. 279 (2022) 114680.

[11] R. Alizadehsani, A. Khosravi, M. Roshanzamir, M. Abdar, N. Sarrafzadegan, D. Shafie, F. Khozeimeh, A. Shoeibi, S. Nahavandi, M. Panahiazar, et al., Coronary artery disease detection using artificial intelligence techniques: A survey of trends, geographical differences and diagnostic features 1991–2020, Comput. Biol. Med. 128 (2021) 104095.

[12] M. Wen, E.B. Tadmor, Uncertainty quantification in molecular simulations with dropout neural network potentials, npj Comput. Mater. 6 (1) (2020) 1–10.

[13] Y. Gao, M.K. Ng, Wasserstein generative adversarial uncertainty quantification in physics-informed neural networks, J. Comput. Phys. 463 (2022) 111270.

[14] D. Kumar, S.B. Alam, T. Ridwan, C.S. Goodwin, Quantitative risk assessment of a high power density small modular reactor (SMR) core using uncertainty and sensitivity analyses, Energy 227 (2021) 120400.

[15] H.D. Kabir, A. Khosravi, D. Nahavandi, S. Nahavandi, Uncertainty quantification neural network from similarity and sensitivity, in: 2020 International Joint Conference on Neural Networks, IJCNN, IEEE, 2020, pp. 1–8.

[16] H.D. Kabir, S. Khanam, F. Khozeimeh, A. Khosravi, S.K. Mondal, S. Nahavandi, U.R. Acharya, Aleatory-aware deep uncertainty quantification for transfer learning, Comput. Biol. Med. (2022) 105246.

[17] E. Augustine, C. Pryor, C. Dickens, J. Pujara, W.Y. Wang, L. Getoor, Visual Sudoku puzzle classification: A suite of collective neuro-symbolic tasks, in: NeSy 2022, 16th International Workshop on Neural-Symbolic Learning and Reasoning, 2022.

[18] S. Pedrammehr, M.R.C. Qazani, H. Asadi, S. Nahavandi, Control system development of a Hexarot-based high-G centrifugal simulator, in: The 20th IEEE International Conference on Industrial Technology IEEE-ICIT, 2019, pp. 13–15.

[19] M.R.C. Qazani, S. Pedrammehr, M.J. Nategh, An investigation on the motion error of machine tools' hexapod table, Int. J. Precis. Eng. Manuf. 19 (4) (2018) 463–471.

[20] S. Pedrammehr, M.R.C. Qazani, S. Nahavandi, A novel axis symmetric parallel mechanism with coaxial actuated arms, in: 2018 4th International Conference on Control, Automation and Robotics, ICCAR, IEEE, 2018, pp. 476–480.

[21] M.J. Tajaril, S. Pedrammehr, M.R.C. Qazani, M.J. Nategh, The effects of joint clearance on the kinematic error of the hexapod tables, in: 2017 5th RSI International Conference on Robotics and Mechatronics, ICRoM, IEEE, 2017, pp. 39–44.

[22] K. Kakhi, R. Alizadehsani, H.D. Kabir, A. Khosravi, S. Nahavandi, U.R. Acharya, The Internet of Medical Things and artificial intelligence: Trends, challenges, and opportunities, Biocybern. Biomed. Eng. (2022).

[23] M.S. Kamal, A. Northcote, L. Chowdhury, N. Dey, R.G. Crespo, E. Herrera-Viedma, Alzheimer's patient analysis using image and gene expression data and explainable-AI to present associated genes, IEEE Trans. Instrum. Meas. 70 (2021) 1–7.

[24] R. Alizadehsani, D. Sharifrazi, N.H. Izadi, J.H. Joloudari, A. Shoeibi, J.M. Gorriz, S. Hussain, J.E. Arco, Z.A. Sani, F. Khozeimeh, et al., Uncertainty-aware semi-supervised method using large unlabeled and limited labeled Covid-19 data, ACM Trans. Multimed. Comput. Commun. Appl. (TOMM) 17 (3s) (2021) 1–24.

[25] H.D. Kabir, M. Abdar, A. Khosravi, S.M.J. Jalali, A.F. Atiya, S. Nahavandi, D. Srinivasan, Spinalnet: Deep neural network with gradual input, IEEE Trans. Artif. Intell. (2022).

[26] A. Gesmundo, J. Dean, An evolutionary approach to dynamic introduction of tasks in large-scale multitask learning systems, 2022, arXiv preprint arXiv:2205.12755.

[27] J. Zheng, C. Luo, T. Li, H. Chen, A novel hierarchical feature selection method based on large margin nearest neighbor learning, Neurocomputing 497 (2022) 1–12.

[28] Z. Abbas, H. Tayara, K.T. Chong, ZayyuNet–a unified deep learning model for the identification of epigenetic modifications using raw genomic sequences, IEEE/ACM Trans. Comput. Biol. Bioinform. 19 (4) (2021) 2533–2544.

[29] F. Uysal, F. Hardalaç, O. Peker, T. Tolunay, N. Tokgöz, Classification of shoulder X-ray images with deep learning ensemble models, Appl. Sci. 11 (6) (2021) 2723.

[30] P.-Y. Yang, S.-Y. Jhong, C.-H. Hsia, Green coffee beans classification using attention-based features and knowledge transfer, in: 2021 IEEE International Conference on Consumer Electronics-Taiwan, ICCE-TW, IEEE, 2021, pp. 1–2.

[31] F. Albardi, H.D. Kabir, M.M.I. Bhuiyan, P.M. Kebria, A. Khosravi, S. Nahavandi, A comprehensive study on torchvision pre-trained models for fine-grained inter-species classification, in: 2021 IEEE International Conference on Systems, Man, and Cybernetics, SMC, IEEE, 2021, pp. 2767–2774.

[32] G. Caruana, L.-L. Lebrun, O. Aebischer, O. Opota, L. Urbano, M. de Rham, O. Marchetti, G. Greub, The dark side of SARS-CoV-2 rapid antigen testing: Screening asymptomatic patients, New Microb. New Infect. 42 (2021) 100899.

[33] Y. Huang, L. Qing, S. Xu, L. Wang, Y. Peng, HybNet: A hybrid network structure for pain intensity estimation, Vis. Comput. (2022) 1–12.

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

[35] F.N. Iandola, S. Han, M.W. Moskewicz, K. Ashraf, W.J. Dally, K. Keutzer, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size, 2016, arXiv preprint arXiv:1602.07360.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, vol. 30, 2017.

[37] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, 2020, arXiv preprint arXiv:2010.11929.

[38] H. Kabir, Reduction of class activation uncertainty with background information, 2023, arXiv preprint arXiv:2305.03238.

[39] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[40] M. Zinkevich, M. Weimer, L. Li, A. Smola, Parallelized stochastic gradient descent, in: Advances in Neural Information Processing Systems, vol. 23, 2010.

[41] S. Alam, J. Palau, C. De Saint Jean, Nuclear data adjustment using Bayesian inference, diagnostics for model fit and influence of model parameters, in: EPJ Web of Conferences, vol. 239, EDP Sciences, 2020, p. 13003.

[42] S. Alam, D. Vučinić, C. Lacor, Uncertainty quantification and robust optimization in engineering, Adv. Visual. Optim. Tech. Multidisc. Res.: Trends Model. Simul. Eng. Appl. (2020) 63–93.

[43] A. Khosravi, S. Nahavandi, D. Creighton, A neural network-GARCH-based method for construction of prediction intervals, Electr. Power Syst. Res. 96 (2013) 185–193.

[44] R. Van de Schoot, S. Depaoli, R. King, B. Kramer, K. Märtens, M.G. Tadesse, M. Vannucci, A. Gelman, D. Veen, J. Willemsen, et al., Bayesian statistics and modelling, Nat. Rev. Methods Prim. 1 (1) (2021) 1–26.

[45] T. Pearce, F. Leibfried, A. Brintrup, Uncertainty in neural networks: Approximately bayesian ensembling, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 234–244.

[46] L. Deng, The mnist database of handwritten digit images for machine learning research [best of the web], IEEE Signal Process. Mag. 29 (6) (2012) 141–142.

[47] A. Coates, A. Ng, H. Lee, An analysis of single-layer networks in unsupervised feature learning, in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 2011, pp. 215–223.

[48] SIIM, Society for imaging informatics in medicine (SIIM), SIIM-FISABIO-RSNA COVID-19 detection, 2021, URL https://www.kaggle.com/c/siim-covid19-detection.

[49] M.d.l.I. Vayá, J.M. Saborit, J.A. Montell, A. Pertusa, A. Bustos, M. Cazorla, J. Galant, X. Barber, D. Orozco-Beltrán, F. García-García, et al., Bimcv Covid-19+: A large annotated dataset of rx and ct images from Covid-19 patients, 2020, arXiv preprint arXiv:2006.01174.

[50] Y.S. Chakrapani, N.V. Rao, M. Kamaraju, A survey of sobel edge detection VLSI architectures, J. Phys.: Conf. Ser. 1804 (2021) 012151.

[51] E.D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, Q.V. Le, Autoaugment: Learning augmentation policies from data, 2018, arXiv preprint arXiv:1805.09501.

[52] S. Zagoruyko, N. Komodakis, Wide residual networks, 2016, arXiv preprint arXiv:1605.07146.

[53] A. Gesmundo, A continual development methodology for large-scale multitask dynamic ML systems, 2022, arXiv preprint arXiv:2209.07326.

[54] H.D. Kabir, A. Khosravi, M.A. Hosen, S. Nahavandi, Partial adversarial training for prediction interval, in: 2018 International Joint Conference on Neural Networks, IJCNN, IEEE, 2018, pp. 1–6.

[55] A. Ashukha, A. Lyzhov, D. Molchanov, D. Vetrov, Pitfalls of in-domain uncertainty estimation and ensembling in deep learning, 2020, arXiv preprint arXiv:2002.06470.

[56] V. Mazzia, F. Salvetti, M. Chiaberge, Efficient-capsnet: Capsule network with self-attention routing, Sci. Rep. 11 (1) (2021) 14634.

[57] A. Byerly, T. Kalganova, I. Dear, No routing needed between capsules, Neurocomputing 463 (2021) 545–553.

[58] P. Foret, A. Kleiner, H. Mobahi, B. Neyshabur, Sharpness-aware minimization for efficiently improving generalization, 2020, arXiv preprint arXiv:2010.01412.