Neural Computing and Applications (2022) 34:22141–22156
https://doi.org/10.1007/s00521-022-07701-8

ORIGINAL ARTICLE

CTSC-Net: an effectual CT slice classification network to categorize organ and non-organ slices from a 3-D CT image

Emerson Nithiyaraj[1] · Arivazhagan Selvaraj[1]

Received: 17 February 2022 / Accepted: 2 August 2022 / Published online: 13 August 2022

Abstract

Computed tomography (CT) is a non-invasive diagnostic imaging modality that reveals more insight into human organs than conventional X-rays. In general, the CT output is a 3-D image that is formed by combining multiple 2D images or slices together. It is essential to keep in mind that not all of the slices provide significant information to detect tumours. Usually, a 3-D CT image obtained from the CT scanners has a significant number of unwanted non-organ slices in it. Radiologists typically devote a significant amount of time to select the slices with organ from a 3-D CT image. The presence of a tumour is only evident in the organ slice; hence, radiologists must be cautious not to skip any organ slices. This work is evaluated on the LITS, 3DIRCADb and COVID-19 CT datasets. The three datasets collectively contain 22,435 organ slices and 53,661 non-organ slices, and there is a huge gap between the number of organ and non-organ slices. There is a need for the automatic elimination of non-organ slices in 3-D CT volumes to assist the physicians, and hence, this work focuses on the automatic recognition of organ slices from 3-D CT volumes. In this paper, a new deep model called the computed tomography slice classification network (CTSC-Net) is proposed for CT slice classification between organ and non-organ slices. The model is trained on 77,980 CT slices, validated on 9748 slices and tested on 12,571 slices. Nine CNN architectures with different layer settings are trained and tested to arrive at the final optimal model. The performance measures are computed in terms of true positive rate, true negative rate, sensitivity, specificity and accuracy. The 20-layer CTSC-Net achieves a validation accuracy of 95.04% and an overall testing accuracy of 99.96%. The proposed model is compared to eight different pre-trained CNN models, and the results of the proposed CTSC-Net surpassed all the comparable models. The activation feature maps of different layers of the CTSC-Net are visualized to verify the discriminative features learned by the network. Hence, the proposed CTSC-Net can be employed as a computer-aided diagnosis tool to help physicians discard unnecessary non-organ slices from the 3-D CT volume and to speed up the CT diagnosis process.

Keywords Medical imaging · CT scan image · Organ slice recognition · Convolutional neural network

1 Introduction

Significant advancements in the fields of artificial intelligence and machine learning have had a substantial impact on the interpretation of medical images for early diagnosis and treatment. Computer-aided diagnosis (CAD) systems have emerged as an assisting tool for doctors in diagnosing medical images. The advancements in healthcare image acquisition devices and growth in medical image modalities have made the task of medical image analysis even more challenging and interesting [1]. Medical imaging modalities like magnetic resonance imaging (MRI), positron emission tomography (PET), ultrasound, X-ray, computed tomography (CT) and mammography produce images of human anatomical structures for assessment and treatment purposes [2].

CT is an X-ray-based imaging technique in which a narrow beam of X-rays is aimed at a patient and quickly rotated around the body, producing many views of the same organ or tissue. The resulting signals from the patient

✉ Emerson Nithiyaraj
ej.jeshua@mepcoeng.ac.in

[1] Department of Electronics and Communication Engineering, Centre for Image Processing and Pattern Recognition, Mepco Schlenk Engineering College, Sivakasi 626005, India

are picked up by the X-ray detectors and analysed by the machine's computer to generate cross-sectional images or slices of the body. As these slices contain more information than traditional X-rays, they are referred to as tomographic images. Once the computer on the machine collects a number of successive slices, they can be displayed individually or digitally stacked together to create a three-dimensional (3-D) image of the patient. The skeleton, organs, and tissues, as well as any abnormalities are all visible in the CT images. The CT image makes it easier to detect and spot basic structures as well as possible lesions or anomalies [3]. These CT images are generally stored in digital imaging and communications in medicine (DICOM) file format with a header and image datasets, where the header contains the patient information like demographics, study parameters, etc. These images are mostly examined by radiologists and at the risk of potential fatigue of experts, poor imaging quality and wide variations in pathology, CAD systems began to perform better as an assisting tool for the experts [4].

The remaining sections of this paper are ordered as follows: The review of literature is briefed in Sect. 2. In Sect. 3, a detailed explanation of the proposed CTSC-Net model is elaborated. In Sect. 4, the experimental setup is provided. The results and discussion are elucidated in Sect. 5. Finally, a conclusion and areas for further research are given in Sect. 6.

2 Related work

Inspired by the success of deep learning techniques in computer vision tasks, deep learning models are employed to detect, classify and segment patterns in medical images [5, 6]. One of the core abilities of deep learning is that they learn features automatically from raw data instead of extracting hand-crafted features from data by the user. In medical sector, deep learning techniques are providing good solutions for diagnosing medical images and it provides an assist for medical experts to interpret and diagnose medical images [7–9]. There are numerous deep learning algorithms in the literature for diagnosing from CT images. In particular, there are many deep learning approaches for CT image classification [10]. Wang et al. [11] developed a deep learning model for nodal metastasis (Nmet) prediction for lung tumours using different energy level CT images. The model includes a principal feature enhancement block (PFE) that incorporates radiologist and feature knowledge for Nmet prediction. The model has resulted in an accuracy of 93%. Zhang et al. [12] proposed a deep learning model that can classify multiple organ specific cancers from CT/PET images. This model is a six-class classification system that includes normal, liver cancer, pancreatic cancer, esophageal cancer, gastric cancer and lung cancer. The model resulted in an F-score of 82.3%, which could assist physicians in screening cancers. Shigeru Kiryu et al. [13] developed a CNN to differentiate five liver tumours, where their CNN has five convolution, three max pooling and three fully connected layers. The developed system was validated using 100 test images which has resulted a median accuracy of 0.95 and 0.84 for training and test data, respectively. The sensitivity of each individual class for test data is 0.71, 0.33, 0.94, 0.90 and 0.10. The dynamic contrast agent-enhanced CT images are downsampled to 70 × 70 pixels. KyuJin Choi et al. [14] developed a deep learning system (DLS) for CT-based staging of liver fibrosis using portal venous phase CT images. The five stages of fibrosis (F1-F5) were no-fibrosis (F1), portal fibrosis (F2), periportal fibrosis (F3), septal fibrosis (F4) and cirrhosis (F5). The developed DLS has two steps such as liver segmentation and fibrosis staging, and both steps were developed on the basis of CNN. The overall staging accuracy of the DLS was 79.4%, and the dataset was not balanced for pathologic fibrosis stage which is a limitation. Chaunzwa et al. [15] developed a deep learning framework for lung cancer classification from CT images. The VGG-16 network architecture is fine-tuned and utilized for the proposed work. This model is validated on a dataset comprising 311 early-stage lung cancer patients and the model is able to predict tumour with an AUC of 0.71. Rahimzadeh et al. [16] developed a deep model to detect COVID-19 from the CT images. The dataset contains 15,589 COVID-19 images from 95 patients and 48,260 normal images from 282 persons. In this work, the images are pre-processed such that the non-lung slices from the 3-D CT image are eliminated based on dark pixels count and a threshold value. Based on prior experiments in the dataset, a fixed region is set for all the images in the area of 120 to 370 pixels in the x-axis and 240 to 340 pixels in the y-axis ([120,240] to [370,340]). For all the slices in 3-D CT, the number of dark pixels in the fixed region is counted and a threshold is set by dividing the difference between the maximum counted number and the minimum counted number by 1.5. The images with higher dark pixels than the threshold is termed as the lung slices, and the images with fewer dark pixels than the threshold are eliminated. The selected lung slices are classified using the proposed model based on ResNet50V2 and feature pyramid network. Still, this work has selected the lung slices by setting a manual fixed region in all the slices, which is a time-consuming process and the area of this fixed region may vary for different datasets.

CT scans produce 3-D medical images by stacking multiple 2-D images or slices on top of one another to provide a volumetric representation of the internal features of human bodies. The CT scan produces a 3-D volume with

50 to more than 1000 slices for each patient, and it depends upon factors such as the type of CT machine being used, the radiologist's interest, and the complexity of the region under diagnosis. Modern CT machines can produce hundreds of slices of a patient, which is very useful in determining the disease's location. However, all the slices do not contain enough useful information for the detection and diagnosis of disease. A major number of slices in the 3-D CT volume do not visualize the whole contour of the organ, and some slices do not contain any region of the organ in them. Firstly, it is important to recognize the CT slices that contain the organ to examine the presence of tumours in them. It is vital for a radiologist not to skip any slices that are candidates for further analysis. The advances in medical imaging tools enable the physician to select the slices of interest virtually. However, the practise is considered semi-automated because a radiologist must scroll through hundreds of slices manually in the software. This approach is not only time-consuming but also subjective, and there's a significant chance that a useful slice will get scrolled out in the process. So, it takes some time for radiologists to cross over the unwanted or non-organ slices and then interpret the organ slices to diagnose any abnormality. A tumour may appear in any organ slice of a 3-D CT volume and in any region within the organ's contour of that slice, so it is important to recognize and diagnose all the slices that contain the organ. In all the above-mentioned literature works, CT image classification is carried out by manually selecting the 2-D slices from a 3-D CT volume that contains the whole contour of the organ and there is no automatic procedure to segregate organ and non-organ slices. Consider a deep model that is trained to classify some types of tumours from a CT dataset that contains only the organ slices. If the same model is tested with images of both organ and non-organ slices, the network may fail. Since the network has learnt to classify only the types of tumours, the network may perform poorly when a non-organ slice is fed in. There are currently no studies in the literature for the classification of CT slices into organ and non-organ slices. It is always important to remove non-organ slices from a CT dataset in order to achieve a faster and more robust diagnosis. This will allow radiologists to focus only on a subset of CT images that contain the most useful information. The prime objective of this work is to develop an automatic model to classify organ and non-organ slices from a 3-D CT image.

In this work, a novel model called computed tomography slice classification network (CTSC-Net) is proposed for CT slice classification between organ and non-organ slices. The proposed CTSC-Net is a 20-layer fully convolutional neural network. The proposed model is evaluated on three clinical datasets, namely LiTS, 3DIRCADb and COVID-19 CT. The performance evaluation metrics such as true positive rate, true negative rate, sensitivity, specificity and accuracy are adopted for statistical analysis. The results of the proposed CTSC-Net model are compared with the pre-trained CNN classification models like AlexNet [17], SqueezeNet [18], Vgg-16 [19], ResNet18 [20], GoogleNet [21], MobileNetV2 [22], ShuffleNet [23] and DarkNet19 [24]. The contributions of this work are summarized as follows:

1. A novel computed tomography slice classification network (CTSC-Net) is proposed to classify organ and non-organ slices from a 3-D CT volume.
2. The proposed model is evaluated on three datasets such as LITS, 3DIRCADb and COVID-19 CT and nine different CNN architectures are developed to arrive at the optimal CTSC-Net.
3. The proposed CTSC-Net is experimented with different activation functions and normalization techniques and promising results are obtained compared to the state-of-the-art deep models.
4. The proposed 20-layer CTSC-Net has attained faster convergence during training, resulting in a testing accuracy of 99.96% for the test set of 12,571 CT slices. The organ slices can be effectively recognized by the proposed CTSC-Net model irrespective of the size and shape of the organ in it.

3 Proposed CTSC-Net

The work flow of the proposed computed tomography slice classification system is depicted in Fig. 1. The proposed work is fed with the 3-D CT image as the input and the individual CT slices enter the pre-processing step. The processed CT slices are fed into the CTSC-Net for the classification between organ and non-organ slices.

The proposed CTSC-Net architecture for the task of CT slice classification is shown in Fig. 2. The architecture has 4 Conv blocks named Conv 1, Conv 2, Conv 3 and Conv 4. Each Conv block has a convolutional layer followed by a batch normalization layer, ReLU layer and a max pooling layer. All the convolutional layers have a fixed kernel size of 5 × 5, and each convolutional layer in each Conv block has a varying number of filters. Zero padding with one row and one column is performed before every convolution operation to maintain the same size of input and output of the convolution layer.

3.1 Convolutional neural network

Classical machine learning techniques [25, 26] require the domain expertise of a human expert to extract significant features from images that define the pattern or regularities

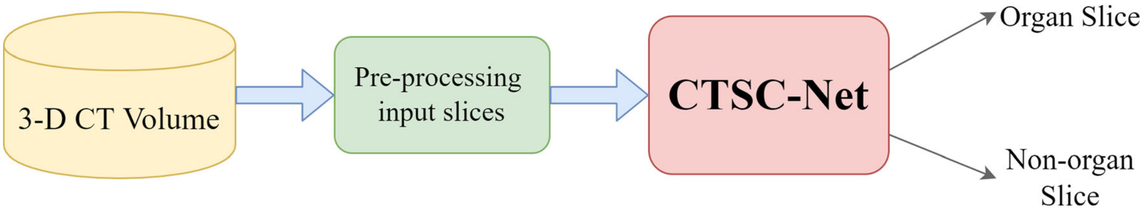


Fig. 1 Work flow of the proposed CT slice classification system

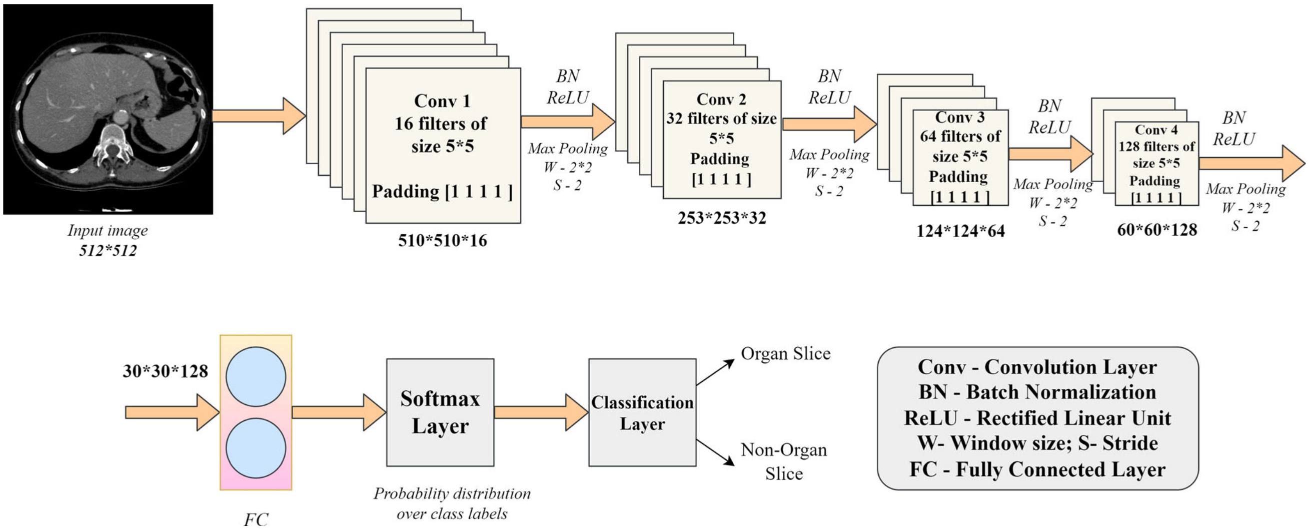


Fig. 2 Proposed CNN architecture for CTSC-Net

in the image, making it difficult for non-experts to understand. Deep learning, on the other hand, automates the learning process by requiring only the input data and discovering the informative representations in a self-taught manner [27]. Deep learning has shifted the burden of feature learning from humans to computers, resulting in improved performance [28]. The convolutional neural network (CNN) is a frequently used deep learning architecture that is based on the structure and function of the human brain and is suited for most computer vision tasks such as classification, detection and segmentation [29].

3.2 Convolutional layer

Convolutional layer is the key element in any CNN architecture which extracts hierarchical features from the input image. Convolution is a mathematical operation that takes two inputs such as image matrix and a filter or kernel. A convolutional layer has multiple number of kernels, and a feature map is produced by convolving each kernel with the input image [30]. In general, convolution is a dot product operation. Figure 3 shows a convolution operation of a 3 × 3 kernel on a 5 × 5 input matrix. The kernel is slided over the input image in both horizontal and vertical manner to produce output feature map. The dimension of the feature map produced by the convolutional layer is based on the given Eq. (1).

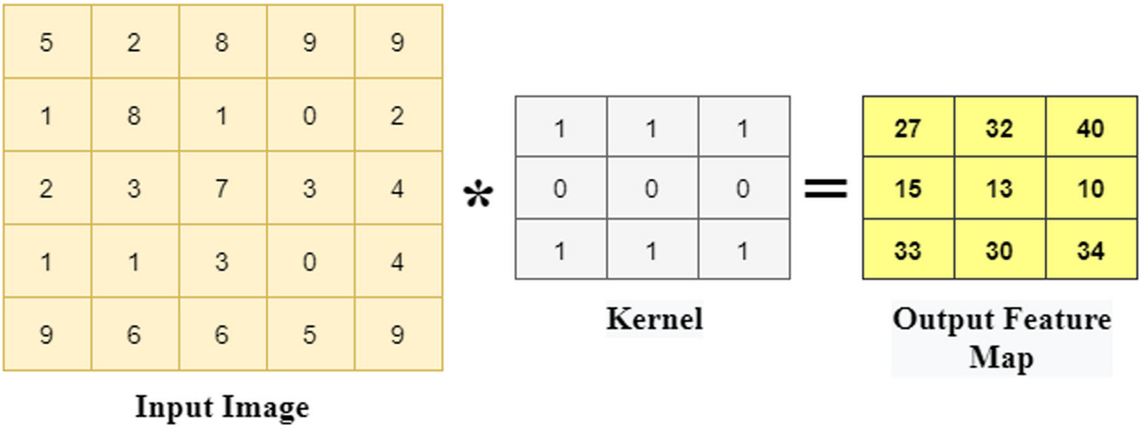
$$O = \frac{Ip - K + 2P}{S} + 1 \tag{1}$$

where O is dimension of output feature map; Ip is input matrix size; K is kernel size; P is padding value; and S is stride value.

The initial convolutional layers extract low level features such as edges, lines and deep convolutional layers learns global features like texture, boundary and shapes. The proposed CNN architecture has four convolutional layers of constant kernel size 5 × 5 and varying number of kernel layers 16, 32, 64, 128 and stride 1 and padding 1.

3.3 Pooling layer

Pooling layer is generally placed after the convolutional layer to reduce the spatial size of the feature maps. Pooling essentially reduces the computation and number of parameters in the network and also helps in extracting the dominant features from the input feature map. Max pooling, min pooling and average pooling are the types of pooling, where max pooling of window size 2 × 2 with a stride value of 2 is used in the proposed CNN architecture. A max pooling operation with window size 3 × 3 and



5	2	8	9	9
1	8	1	0	2
2	3	7	3	4
1	1	3	0	4
9	6	6	5	9

1	1	1
0	0	0
1	1	1

27	32	40
15	13	10
33	30	34

Fig. 3 Convolution operation of a (5 × 5) image and (3 × 3) kernel

stride 2 is explained in Fig. 4, where maximum value of each 3 × 3 window is the output. The max pooling layer reduces the size of the incoming feature matrix based on the given Eq. (2).

$$P = \frac{Ip - W + 2P}{S} - 1 \tag{2}$$

where P is output size of max pooling layer; Ip is input matrix size; W is max pooling window size; P is padding value; and S is stride value.

3.4 Batch normalization

During the training of the network, the distribution of the activations is constantly changing. This slows down the training process because each layer must learn to adapt itself to a new distribution in every training step. This problem is known as internal covariate shift. Batch normalization normalizes the inputs of each layer and so the problem of internal covariance shift is reduced [31]. The steps followed by the batch normalization layer during the training phase are given below.

1. Calculate the mean and variance of the layer's input.

$$\text{Batch mean:} \quad \mu_b = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{3}$$

$$\text{Batch variance:} \quad \sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_b)^2 \tag{4}$$

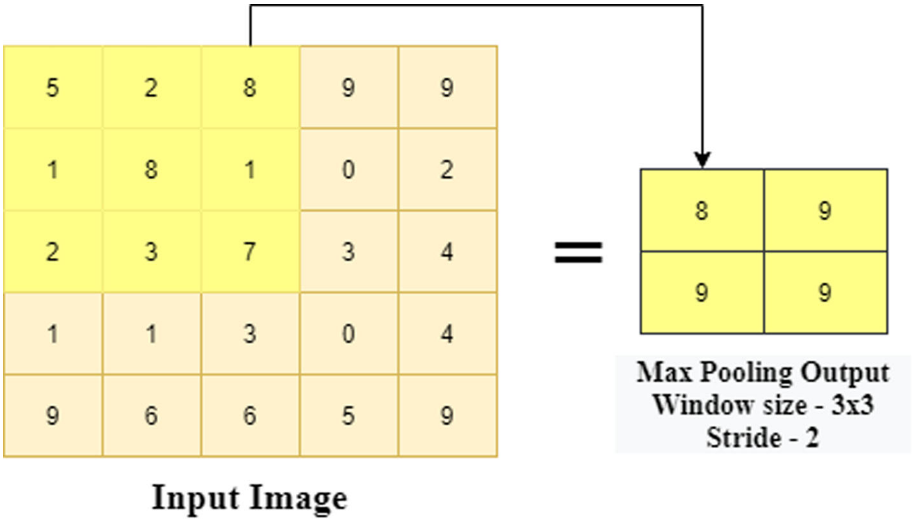
2. Normalize the layer inputs using the previously calculated batch statistics.

$$\overline{x}_i = \frac{x_i - \mu_b}{\sqrt{\sigma_B^2 + \epsilon}} \tag{5}$$

3. Scale and shift in order to obtain the output of the layer.

$$y_i = \gamma x_i + \beta \tag{6}$$

Fig. 4 Max pooling operation of window size 3 and stride 2



5	2	8	9	9
1	8	1	0	2
2	3	7	3	4
1	1	3	0	4
9	6	6	5	9

8	9
9	9

Parameters γ and β in Eq. (6) are learned during training phase and the past means and variances of each training batches are used for the testing phase. In this proposed CNN architecture, batch normalization is done after each convolutional layer.

3.5 Activation functions

Activation functions introduce nonlinearity in the network by applying a nonlinear transformation to the input. There are different activation functions such as tangent, sigmoid, rectified linear unit (ReLU) that makes the network to learn and perform more complex tasks. ReLU is the commonly used activation function where any negative input given into the ReLU becomes zero. In this work, ReLU activation function is used. The main advantage of using the ReLU function over other activation functions is that it is computationally efficient, since it does not activate all the neurons at the same time [32]. The ReLU activation function $R(i)$ can be defined as in Eq. (7).

$$R(i) = \begin{cases} i & i > 0 \\ 0 & i \leq 0 \end{cases} \tag{7}$$

Finally, fully connected (FC) layer is a simple feed forward neural network where the output from convolutional or pooling layer is flattened (i.e., matrix values are unrolled into a vector) and fed into the FC layer. The final layer is a softmax activation that is used to get the probabilities of the input image belonging to a particular class in a classification application and classification layer classifies the input image as liver or non-liver slice.

3.6 Loss function

Loss function measures the loss value that is used to predict the error rate of the network. In this work, binary cross-entropy loss function is used to evaluate the loss value. Cross-entropy loss function is given in Eq. (8).

$$\mathrm{CE} = -\sum_{i}^{c} t_i \log s_i \tag{8}$$

where t_i is the actual value and s_i is the predicted output of the CNN for each class i € {liver, non-liver}.

3.7 Learning algorithm

Training and testing are the two phases in developing any model for classification application. Training helps to optimize the parameters of the model such as weights, biases and testing evaluates the performance of the model. The initial weights are assigned based on Gaussian distribution with the mean and standard deviation values as zero and 0.01, respectively, and the initial bias values are set as zero. In CNN, the images are forward passed into the network where various layers in the architecture perform their respective functions and finally the loss function is calculated between the predicted output and actual output. Using the loss value, the network parameters such as weights and biases of the network are updated using a backpropagation technique called gradient descent algorithm [33]. This algorithm aims to minimize the loss function by updating the network parameters in the direction of the negative gradient of the loss function [34]. This process of weight updation is repeated for as many epochs as necessary to reach the desired level of accuracy. The weight updation is given in Eq. (9).

$$\theta_{i+1} = \theta_i - \alpha \nabla E(\theta_i) \tag{9}$$

where θ is the parameter vector (weights or biases), i is the iteration number, $E(\theta)$ is the loss function, $\nabla E(\theta)$ is the gradient of the loss function, and α is the learning rate. Learning rate refers to the number of weights that are updated during training with respect to the loss gradient. The training data are split into batches called mini-batch and full pass of the entire training data using mini batches into the training algorithm is one epoch. Mini batch stochastic gradient descent (MBSGD) is a variant of gradient descent algorithm, in which the parameter updates are computed for every mini-batch. A variant of MBSGD called Adam optimizer that is derived from adaptive moment estimation is used in this work [35].

4 Experimental setup

4.1 Dataset

The 'LITS-Liver Tumour Segmentation Challenge', 3DIRCADb (3D Image Reconstruction for Comparison of Algorithm Database) and COVID-19 CT scan datasets are used in this work, which are publically available [36–38]. The LITS and 3DIRCADb datasets contain the CT images of the liver organ, and the COVID-19 CT dataset contains the CT images of the lung organ. The LiTS data are collected from seven academic and clinical institutions around the world, and the image data are acquired with different CT scanners and acquisition protocols [39]. The 3DIRCADb is provided by the University Hospital in Strasbourg, France (Centre Hospitalier et Universitaire), and the COVID-19 CT scan dataset provided by Kaggle. The LITS dataset contains 130 CT scans and the 3DIRCADb and COVID-19 CT datasets contain 20 CT scans each. Each scan has varying number of 2-D slices ranging from 46 to 1026, and the image size is 512 × 512 pixels. The CT scans also contains abnormalities from primary to secondary level tumour and metastases. Some sample images from the datasets are shown in Fig. 5 including both organ and non-organ slices.

As seen in Fig. 5a, c, e, the organ slices maintain a constant intensity range over a larger area than the non-organ slices in Fig. 5b, d, f. From the LITS dataset, all the 71,131 axial 2-D slices from the 130 scans are used and it includes 19,907 liver slices and 51,224 non-liver slices. From the 3DIRCADb dataset, all the 2823 axial 2-D slices from the 20 CT scans are selected and it contain 1096 liver slices and 1727 non-liver slices. The COVID-19 CT dataset has 2142 CT slices, combining 1432 lung slices and 710 non-lung slices. The three datasets have all the different shapes and sizes of the organ. The total number of organ and non-organ slices of the three datasets is 22,435 and 53,661. All the three datasets have a huge difference between the number of organ and non-organ slices. To avoid data imbalance between organ and non-organ slices, data augmentation techniques like horizontal flipping, vertical flipping and horizontal vertical flipping are performed on the organ slices of the LITS and COVID-19 CT datasets. As a result of data augmentation, a total of 1,00,299 slices were acquired, including 46,638 organ slices and 53,661 non-organ slices.

4.2 Image pre-processing

The pixel values of each CT image are in Hounsfield units [40]. Hounsfield units (HU) are a dimensionless unit universally used in computed tomography (CT) scanning to express CT numbers in a standardized and convenient form. The Hounsfield density of tissues reflects their attenuation of X-ray and is proportional to their physical density. The Hounsfield density provides accurate density for the type of tissue. The HU for various regions like water, air and bone is 0 HU, air, − 1000 HU and + 1000 HU, respectively. The HU unit of the LITS, 3DIRCADb and COVID-19 datasets varies for different cases. In order to make all CT volumes convenient for visualization and further processing, the intensity values of the CT volumes are windowed to a particular range. A truncation range of (400, − 100) is applied to the CT slices of LITS and 3DIRCADb datasets, and a truncation range of (400, − 1200) is applied to the CT slices of COVID-19 dataset.

The difference between the raw CT image from the dataset and truncated image is shown in Fig. 6. The texture of the organs in the CT image can be clearly seen in the truncated image. Hence, it is more effective to carry out further processing using the processed image.

4.3 Performance evaluation metrics

The performance of the proposed CTSC-Net is statistically evaluated in terms of sensitivity, specificity, accuracy and AUC [41]. Sensitivity (also called as true positive rate (TPR)) measures the proportion of positive samples that are correctly identified as positives and specificity (also called as true negative rate (TNR)) measures the proportion of negative samples that are correctly identified as negatives as given in Eqs. (10) and (11). Accuracy is the number of correctly classified samples to the total number of samples as given in Eq. (12). False positive rate (FPR) and false negative rate (FNR) denote the misclassified samples, which in reality these samples belong to the other class. AUC stands for 'area under the ROC (receiver operating characteristics) curve' and is a performance measurement for classification problem which is plotted with TPR against the FPR [42]. AUC represents the model's degree of separability between the classes where AUC value of 1 indicates a good measure of separability and AUC value of 0 indicates worst measure of separability.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{10}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{11}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{12}$$

where TP refers to true positive, FP refers to false positive, TN refers to true negative, and FN refers to false negative.

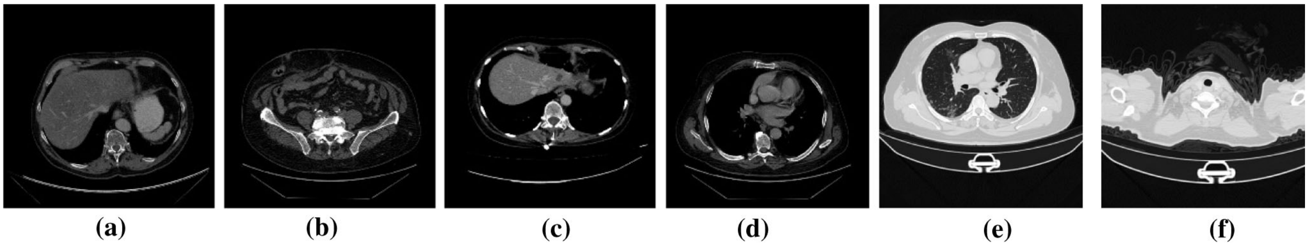


Fig. 5 **a**, **b** CT slices from LITS, **c**, **d** CT slices from 3DIRCADb, **e**, **f** CT slices from COVID-19 CT

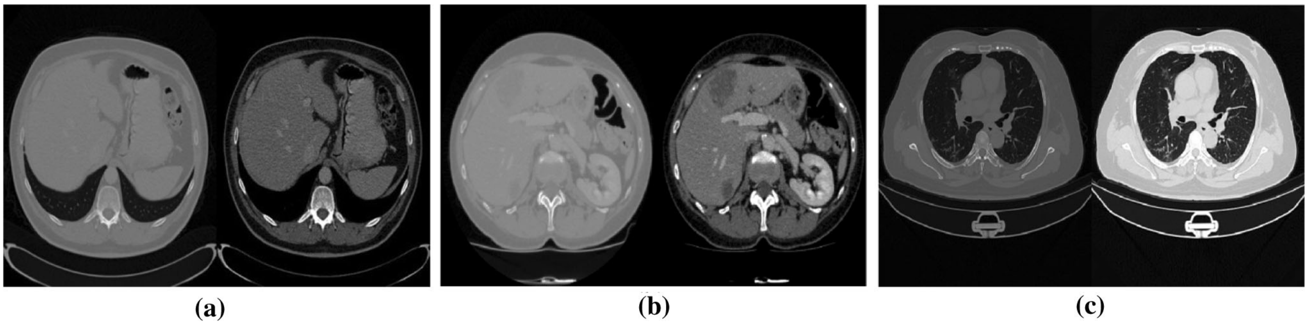


Fig. 6 Raw image (left) and truncated image (right) from: **a** LITS, **b** 3DIRCADb and **c** COVID-19 CT datasets

5 Experimental results and discussion

The experiments that were conducted to evaluate the proposed CTSC-Net for CT slice classification using LITS, 3DIRCADb and COVID-19 CT datasets are presented illustratively. The prime objective of the proposed CTSC-Net is to classify between organ and non-organ CT slices from a 3-D CT volume. The implementation of the proposed work is done using Matlab 2020a in a personal computer having Intel (R) core i5 and 64-bit processor with 8 GB RAM. The optimization algorithm used in this work is Adam optimizer, and the mini batch size is 32. The mini batch images are shuffled for every epoch. The weights of the network are initialized by Glorot initializer [43] which independently samples from a uniform distribution of zero mean and variance *2/(input size + output size)* and the initial bias values are zero. In this work, the network is trained for 10 epochs, initial learning rate is set as 0.01, and the learning rate drop factor is set to 0.001 for every 5. The learning rate for epochs 1 to 5 is 0.01, and the learning rate for epochs 6 to 10 is 10^{-5}.

5.1 Results of the proposed CTSC-Net

For training the network, 72,830 CT slices from the LITS dataset and 5150 slices from the COVID-19 CT dataset are used. For validation, 9104 slices from the LITS dataset and 644 slices from the COVID-19 CT dataset are used. For testing, 9104 slices from LITS, 644 slices from COVID-19 CT, and all 2823 slices from the 3DIRCADb dataset are used. The training set contains 77,980 slices, which includes 38,990 organ slices and 38,990 non-organ slices. The testing set of 12,571 slices includes 6285 organ slices and 6286 non-organ slices to test the proposed CTSC-Net. The remaining 9748 slices are allotted for validation. The collective dataset includes non-organ slices and organ slices of the liver and lungs. This training set is ensured such that an equal number of organ and non-organ slices are present in order to avoid overfitting. Since deep learning models can unintentionally memorize training data, it is important to make sure that there is no case overlap between the sets. The 2-D slices of a case only occur in one of the sets. When the training and testing sets are split in this way, all the slices of a case are in the same set. Hence, overfitting of the model is prevented when there are no case overlaps between the sets. All the images are pre-processed as discussed in Sect. 4.2 and then fed into the CTSC-Net model.

Nine CNN architectures with different layer assemblies are built for experimentation, and their performance is compared to arrive at the best CNN architecture and finally, the efficient CTSC-Net is proposed. Table 1 presents the accuracy obtained while training, validating and testing the nine CNN architectures for the task of organ and non-organ CT slice classification. From Table 1, it can be observed that, for the 1st architecture, with 100% training accuracy, the validation accuracy is only 58.82% and testing accuracy is 80.91%. This accuracy difference in training and validation is due to the overfitting of the model to the training data. Because of overfitting, the model is not able to generalize for unseen test data. Similarly, for 2nd to 7th architectures, this overfitting can be observed even after increasing additional layers, providing batch normalization and enabling max pooling. The architecture 8 seems to improve both validation and testing accuracy, but still the validation accuracy is lesser. Finally, the architecture 9 yields the maximum validation and testing accuracy and this model has perfectly overcome overfitting. Based on this inference, the architecture 9 is selected as the optimal network for the proposed CTSC-Net model. The proposed CTSC-Net took approximately 9 h for training the model using 77,980 images for 10 epochs.

All the architectures listed in Table 1 are initially trained using a constant learning rate of 0.01, but the results are not convincing. Hence, adaptive learning rate is adopted. Adaptive learning rate changes the learning rate during the training process where the initial learning rate is specified for some epochs and the learning rate is multiplied with the learning rate drop factor for every certain number of epochs. Constant learning rate of 0.01 produced 92.64%

Table 1 Experimentation with different set of layers to find an optimal CNN architecture

Architecture No	Layers	Kernel size	Number of kernels	Training accuracy in %	Validation accuracy in %	Testing accuracy in %
1	Conv	5	16	100	58.82	80.91
2	Conv	5	16	100	66.38	75.19
	Max pooling with (2 × 2)	–	–			
3	Conv	5	16	100	75.97	73.74
	BN					
4	Conv	5	16	100	72.77	76.36
	BN	–	–			
	Max pooling with (2 × 2)	–	–			
5	Conv1					
	Conv	5	16	100	80.14	87.40
	BN	–	–			
	ReLU	–	–			
	Max poolingwith (2 × 2)	–	–			
6	Conv	5	16	100	79.26	85.56
	Conv	5	32			
	BN	–	–			
	ReLU	–	–			
	Max pooling with (2 × 2)	–	–			
7	Conv1	5	16	100	79.53	85.56
	Conv1	5	32			
8	Conv1	5	16	100	89.85	94.74
	Conv1	5	32			
	Conv1	5	64			
9	Conv1	5	16	100	95.04	99.96
	Conv1	5	32			
	Conv1	5	64			
	Conv1	5	128			

*Conv, Convolutional layer; BN, Batch normalization; ReLU, Rectified linear unit; Conv1, a block of four layers

Table 2 Confusion matrix of the test images

Actual class	Predicted class	
	Organ slice	Non-organ slice
Organ slice	6285	0
Non-organ slice	5	6281

accuracy and adaptive learning rate improved the accuracy of CTSC-Net to 99.96% for the task of classification between organ and non-organ CT slices.

The confusion matrix is used to describe the performance of a classification model on a test set. The confusion matrix for the testing set of images is shown in Table 2. Thus, using the Eqs. (10), (11), (12) and the confusion matrix, the performance measures are evaluated. The proposed method results in 99.93% sensitivity, 100% specificity, 99.96% accuracy.

5.1.1 Experimentation with different normalization techniques

The proposed network is experimented with different normalization techniques, and the results are presented in Table 3. From Table 3, it is inferred that batch normalization results in maximum test accuracy and avoids overfitting.

Table 3 Summary of the effect of different normalization techniques in the CTSC-Net

Type	Validation accuracy in %	Testing accuracy in %
Without normalization	50	50
Cross channel normalization	89.06	92.64
Batch normalization	95.04	99.96

5.1.2 Experimentation with different activation functions

Even though ReLU is the most commonly used activation function for any CNN architecture, the proposed architecture is also experimented with hyperbolic tangent, leaky ReLU and clipped ReLU activation functions and their performance is shown in Fig. 7. As inferred from Fig. 7, ReLU activation function in the CTSC-Net outperforms the tanh, leaky ReLU and clipped ReLU activation functions for the task of CT slices classification.

5.1.3 Visualization of the feature activation maps

A richer grasp of what the CNN model learns can be achieved by visualizing the feature maps that are obtained as a result of applying the kernels to the input. A sample liver slice from the LiTS dataset is shown in Fig. 8a, and it is fed into the trained CTSC-Net to visualize the activations of different layers of the proposed network. Mostly, CNNs learn to detect features like colour and edges in their first convolutional layer and deep layers build up their features by combining features from earlier layers. Figure 8b shows the activations of 128 kernels of the ReLU layer in the conv4 block of the proposed architecture, where white pixels represent strong positive activations and black pixels represent strong negative activations. As depicted in Fig. 8b, the white pixels at some channels indicate that the channel is strongly activated at that position. As evident in Fig. 8b, the whole liver organ is strongly activated at some of the channels of the conv4 block. A non-liver slice shown in Fig. 9a is fed into the proposed architecture, and the activations of 128 kernels of the ReLU layer in the conv4 block are shown in Fig. 9b. The activations of the ReLU layer in the conv4 block show that the vertebral canal region is strongly activated and these strong activations in the non-liver region decide the slice as a non-liver slice. Some kernels of conv4 block learn the contour of the actual organ, as shown in Fig. 8b. Hence, the robustness of the proposed CTSC-Net is proved by visualizing the feature activation maps and the proposed network proves to be an effective model to classify between organ and non-organ slices.

5.2 Comparison of the proposed CTSC-Net with conventional machine learning techniques

For the purpose of comparison with conventional techniques, the machine learning techniques [44–46] are used and features are extracted from the same datasets and are trained using different classifiers. Conventional feature extraction techniques like grey level run length matrix (GLRLM), grey level co-occurrence matrix (GLCM), wavelet-based feature extraction techniques and classifiers like support vector machine (SVM) and AdaBoost are used. Different wavelets like db1, db2, db3, db4 are applied on the images, and features were extracted from all the four components of the wavelet decomposition such as

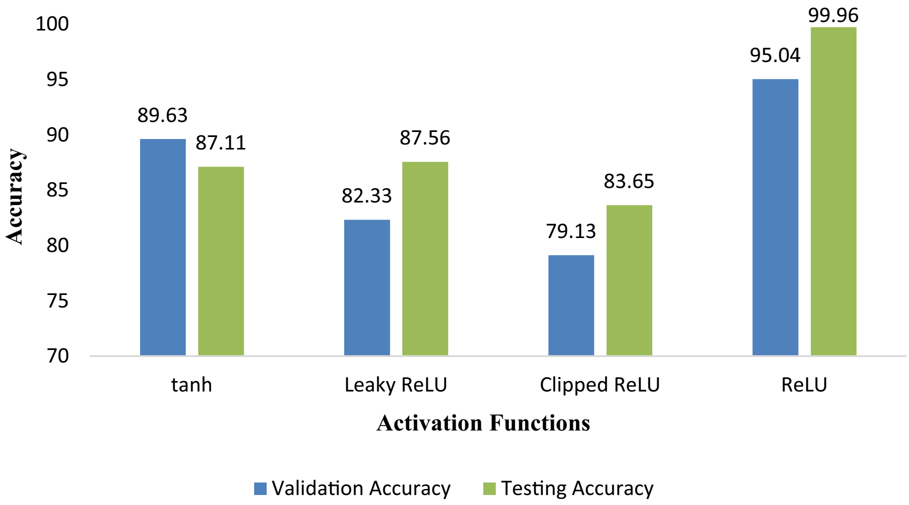


Fig. 7 Response of the proposed CTSC-Net to different activation functions

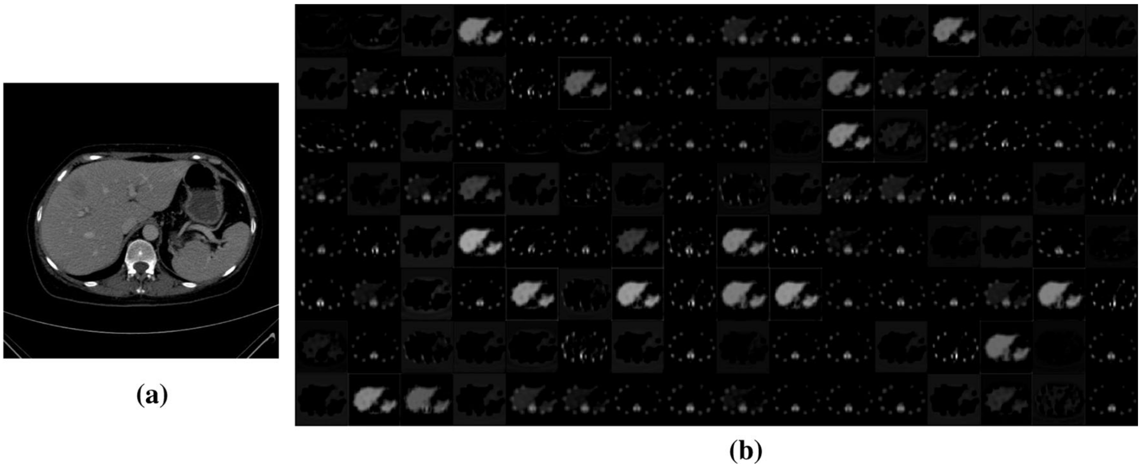


Fig. 8 **a** Sample organ slice, **b** visualization of activation maps corresponding to 128 kernels of 'conv4' ReLU layer

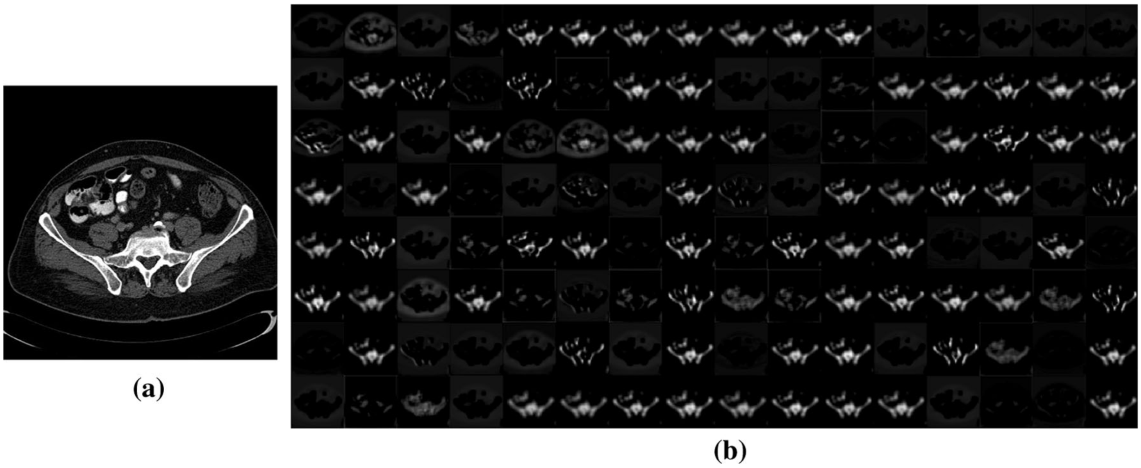


Fig. 9 **a** Sample non-organ image, **b** visualization of activation maps corresponding to 128 kernels of 'conv4' ReLU layer

Table 4 Comparison of the proposed CNN model with the conventional machine learning techniques

S.No	Input image	Features extracted	SVM accuracy (%)	AdaBoost accuracy (%)
1	Raw image	Histogram based features	64	72
2	Raw image	GLRLM features	80	89
3	Raw image	GLCM features	75	89
4	Wavelet transformed image. Wavelet: Db1 Wavelet, decomposition level: level 2	GLRLM features from approximation + details (combining all the four coefficients)	65	88
5	Wavelet transformed image. Wavelet: Db1 wavelet, decomposition level: level 2	GLCM features from approximation + details (combining all the four coefficients)	68	79
6	Proposed CTSC-Net model		CNN accuracy 99.96	

approximation (LL), horizontal detail, vertical detail, diagonal detail and the discrete wavelet transform (DWT) decomposition level is done up to 3rd level. SVM and AdaBoost classifiers [47, 48] are used to classify the liver slices from non-liver slices. From Table 4, it is inferred that, among db1, db2, db3, db4, a maximum accuracy of 88% is achieved using the GLRLM features obtained from db1 wavelet for the two level of DWT decomposition while GLCM and GLRM features extracted from raw images results in an accuracy of 89% using AdaBoost classifier.

But such conventional techniques using hand crafted features need a lot of manual intervention. Since CNN automates the task of feature learning from the input data, the time taken for experimenting and finding an effective conventional machine learning-based technique for an application is not needed. Hence, the proposed CTSC-Net model outperforms the conventional techniques with an accuracy of 99.96% and the proposed method proves to be an effective and accurate system which could assist the radiologists in diagnosing a 3-D CT volume.

5.3 Comparison of the proposed CTSC-Net with existing Deep Learning methods

CNN architectures [37] like AlexNet, SqueezeNet, Vgg-16, ResNet18, GoogleNet, MobileNetV2, ShuffleNet and DarkNet19 are the pre-trained deep models that were trained on more than a million images from the ImageNet database, and these architectures were trained to classify over a thousand different object categories. Transfer learning is a way of training the deep learning models which transfers the learned features from the pre-trained CNN architectures to perform a new task [38]. The optimal weights of the pre-trained architectures are set as the initial weights for the new task and then these models learn and update the weights during training.

Table 5 presents the comparison of the proposed model with the existing state-of-the-art deep learning models for the classification of organ and non-organ slices using the LITS, 3DIRCADb and COVID-19 CT datasets. For any pre-trained CNN architecture, the input layer requires the input image size to be the same as per its own architecture [49]. Since changing the input image size in the input layer would affect all internal dimensions of the corresponding architecture, the input image size for all the existing deep models in Table 5 is changed according to the actual input size of the corresponding architectures. The implementation of the existing models in Table 5 used the same dataset and same hyper-parameter settings (epochs, weights initialization, loss function, learning rate and optimization algorithm) as CTSC-Net.

Transfer learning [50] is performed in all the methods 1 to 8 in Table 5. In each method 1 to 8, the optimal weights of the respective pre-trained network are considered as the initial weights for the task of CT slice classification and the weight updates are performed over the initial weights. In contrast, the proposed CTSC-Net is built from scratch based on various experiments as discussed in Table 1. As inferred from Table 5, the CTSC-Net has outperformed all the other existing methods for the task of CT slice classification between organ and non-organ slices. Still, the existing architectures could not provide the maximum accuracy where accurate results are the most essential in medical imaging applications. The proposed CNN architecture has only 20 layers, which is very little compared to the pre-trained architectures discussed in Table 5. The proposed model contains only 0.5 million network parameters, which is less compared to the other existing models. The memory consumption of the trained CTSC-Net model is 5.39 MB, which outperforms the other comparable models. In the proposed work, the original dimension of the dataset images is used where the CTSC-Net has effectively learnt the discriminative features from the input slices and has produced the maximum results compared to the state-of-the-art deep learning methods. Hence, the proposed CTSC-Net model has produced the maximum accuracy as well as the highest AUC score compared to the existing deep models for the task of CT slice classification.

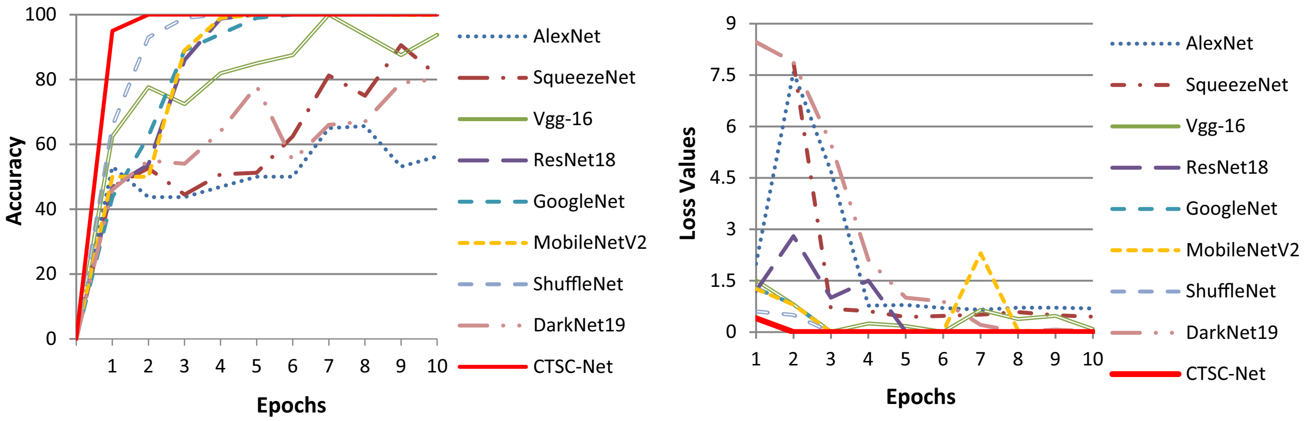
Table 6 presents the network complexity of different models in terms of training time of an epoch, prediction time per image, number of network parameters and memory consumption of the trained model. As evident in Table 6, the network complexity of CTSC-Net is very minimal than the other existing models. The training time of a complete epoch is considered as the unit training time. The training time and prediction time of the proposed model are less than the other models. The training time of CTSC-Net is obviously less due to the lower number of

Table 5 Comparison of the proposed CTSC-Net with the state-of-the-art deep learning models

S. No	Model name	Input image size	No of layers	Testing accuracy (%)	AUC
1	AlexNet [17]	227 × 227 × 3	25	50	0.50
2	SqueezeNet [18]	227 × 227 × 3	68	76.02	0.887
3	Vgg-16 [19]	224 × 224 × 3	41	91.21	0.984
4	ResNet18 [20]	224 × 224 × 3	71	93.80	0.976
5	GoogleNet [21]	224 × 224 × 3	144	91.6	0.965
6	MobileNetV2 [22]	224 × 224 × 3	154	83.04	0.830
7	ShuffleNet [23]	224 × 224 × 3	172	89.10	0.922
8	DarkNet19 [24]	256 × 256 × 3	64	75.10	0.792
9	Proposed CTSC-Net	512 × 512 × 1	20	99.96	0.985

Table 6 Comparison of network complexity of CTSC-Net with other models

S. No	Model name	Training time of an epoch (hours)	Prediction time per image (seconds)	Parameter quantity (M—million)	Memory consumption (MB—megabyte)
1	AlexNet [17]	4.7	5.8	60 M	990 MB
2	SqueezeNet [18]	4.5	5.2	51.2 M	780 MB
3	Vgg-16 [19]	6.2	9.1	138 M	2309 MB
4	ResNet18 [20]	2.9	3.6	11.7 M	112 MB
5	GoogleNet [21]	2.5	2.9	7 M	73 MB
6	MobileNetV2 [22]	1.9	2.2	3.4 M	37.2 MB
7	ShuffleNet [23]	1.4	1.8	2.3 M	29 MB
8	DarkNet19 [24]	3.4	4.2	20 M	197 MB
9	Proposed CTSC-Net	0.9	1.15	0.5 M	5.39 MB



(a) (b)

Fig. 10 **a** Training accuracy of different models at each epoch, **b** loss values of different models at each epoch

network parameters. The proposed model contains only 0.5 million network parameters, which is less compared to the other existing models. The memory consumption of the trained CTSC-Net model is 5.39 MB, which outperforms the other comparable models. Hence, the computational burden of the CTSC-Net is reduced to a great extent compared to the existing models.

For all the models in Table 5, the training accuracy and loss values at each epoch are depicted in Fig. 10a, b.

The performance of the proposed CTSC-Net is indicated by the red straight lines in both the training accuracy and training loss plots. The accuracy and loss values for different models are shown by different types of lines and colours. As seen in Fig. 10a, the training accuracies of VGG-16, DarkNet19, SqueezeNet and AlexNet models have not reached their maximum at the end of 10 epochs. Despite the fact that all of the remaining models have reached maximum training accuracy, the CTSC-Net achieves maximum training accuracy faster than the other models. As evident in Fig. 10b, the proposed CTSC-Net shows faster convergence than all the other existing models. The training loss value of the CTSC-Net reached and settled around the stable-point final solution faster than all the other models. Hence, the CTSC-Net proves to be a more efficient model than the existing models in terms of faster convergence.

The proposed task of classification of organ and non-organ slices can also be accomplished using a segmentation model. To affirm the presence of an organ in the CT slice, the remaining pixels from the segmented organ region can be analysed. The segmentation networks consist of double the number of network layers and require nearly double the number of training parameters as the classification networks. The higher number of training parameters leads to high computational complexity and high memory usage. Any deep segmentation model should be provided with the ground truth or label images along with the actual input images. At each iteration during training of the model, the segmentation model deals with the input image along with its corresponding ground truth and it needs more computational power than a classification model. Any small margin of error during segmentation will result in the incorrect classification of organ slices. Thus, segmenting the organ and analysing the segmented results for the presence of an organ is a two-step computationally complex and time-consuming process. Consequently, the classification strategy is an efficient and straightforward approach for distinguishing organ slices from non-organ slices in a 3-D CT image.

There are some advantages to the proposed CTSC-Net when compared with the existing deep models. First, the CTSC-Net has performed very well on three different datasets, such as LiTS, 3DIRCADb and COVID-19 CT. The proposed model is not trained and tested on the three datasets individually; indeed, all the 2-D slices from the three datasets are mixed together. The model is collectively trained and tested on the 2-D slices of the three datasets. None of the slices of 3DIRCADb dataset are included in the training and validation sets, but only in the test set. Even though the CTSC-Net has been trained only on the LITS and COVID-19 CT datasets, the model has classified the slices of the 3DIRCADb dataset precisely. Second, the proposed model has attained faster convergence during training than the other existing deep models. Third, the 20-layer CTSC-Net has resulted in 0.5 million network parameters and the memory consumption of the trained model is 5.39 MB. Low memory consumption and high computational efficiency are achieved by using a lower number of layers in the architecture. Fourth, many authors have down-sampled the original image size to compensate for the memory requirements and network performance in the literature, but the original image of size 512 × 512 pixels is used in this work. Even though down-sampling the image produces faster computation, it might result in some loss of information from the images. Thus, the constant intensity range of the organ in the CT slice makes the CNN kernels learn discriminating and effective features for classification between organ and non-organ slices. Fifth, the organ in any shape and size can be recognized by the proposed method so that tumour detection on the recognized organ slices can be done effectively. Finally, the CTSC-Net focuses on reducing the time taken by radiologists for differentiating the organ and non-organ slices in a 3-D CT volume. The proposed method can automatically recognize organ slices from a 3-D CT volume and this will fasten the initial stages of analysing a CT scan image. The application of the proposed work is that the organ slices recognized by the CTSC-Net can be used as inputs to the organ segmentation or tumour detection algorithms where the input organ slices are automatically selected by the proposed method instead of manual process. Hence, the proposed CTSC-Net can automatically select the appropriate organ slices from the 3-D volume where those slices contain the organ of interest, and from those selected slices, the diagnosis can be done easily and faster.

6 Conclusion

In this paper, a novel CNN called the computed tomography slice classification network (CTSC-Net) is proposed for the automatic classification of organ and non-organ slices from a 3-D CT volume. The proposed system is validated on three different datasets such as LITS, 3DIRCADb and COVID-19 CT, including the liver and lung organs. Nine different CNN architectures are developed and experimented with the collective dataset to arrive at the optimal CTSC-Net. The CTSC-Net achieved the highest accuracy of 99.96%, sensitivity of 99.93% and specificity of 100% for a test set of 12,571 CT slices. The main advantage of the proposed work is that organ slices are classified accurately irrespective of the organ's shape and size, since medical image diagnosis should be maximally accurate in order to prevent any misdiagnosis. For better comparison, the same task of CT slice classification is also experimented with using different conventional machine learning techniques as well as pre-trained deep models. The proposed CTSC-Net is a 20-layer architecture that has outperformed all the comparable models in the literature for CT slice classification between organ and non-organ slices. The visualization of the feature activation maps of the trained model also ensures that the CTSC-Net has learned discriminative features and the proposed network proves to be an effective model to classify between organ and non-organ slices. The proposed model achieved only 0.5 M network parameters, which is a significant outcome for this work. Hence, the proposed CTSC-Net is an effective model to recognize the organ slices from a 3-D CT volume that will be helpful in clinical diagnosis and reduce the time of diagnosis. In the future, the proposed CTSC-Net can be tested on other medical imaging modalities such as MRI, PET and for other classification tasks.

Data availability Available online [36–38].

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Doi K (2007) Computer-aided diagnosis in medical imaging: historical review, current status and future potential. Comput Med Imaging Graph 31:198–211. https://doi.org/10.1016/j.compmedimag.2007.02.002
2. Van Beers BE, Daire J-L, Garteiser P (2015) New imaging techniques for liver diseases. J Hepatol 62:690–700. https://doi.org/10.1016/j.jhep.2014.10.014
3. Computed Tomography (CT or CAT) Scan of the Liver and Biliary Tract | Johns Hopkins Medicine. https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/computed-tomography-ct-or-cat-scan-of-the-liver-and-biliary-tract. Accessed 3 Feb 2021
4. Fujita H (2020) AI-based computer-aided diagnosis (AI-CAD): the latest review to read first. Radiol Phys Technol 13:6–19. https://doi.org/10.1007/s12194-019-00552-4
5. Yousef R, Gupta G, Yousef N, Khari M (2022) A holistic overview of deep learning approach in medical imaging. Multimed Syst 28:881–914. https://doi.org/10.1007/s00530-021-00884-5
6. Pradhan N, Dhaka VS, Rani G, Chaudhary H (2020) Transforming view of medical images using deep learning. Neural Comput Appl 32:15043–15054. https://doi.org/10.1007/s00521-020-04857-z
7. Abdou MA (2022) Literature review: efficient deep neural networks techniques for medical image analysis. Neural Comput Appl. https://doi.org/10.1007/s00521-022-06960-9
8. Nithiyaraj EE, Arivazhagan S (2020) Survey on recent works in computed tomography based computer—aided diagnosis of liver using deep learning techniques. Int J Innov Sci Res Technol 5:9. https://doi.org/10.38124/IJISRT20JUL058
9. Chen X, Wang X, Zhang K et al (2022) Recent advances and clinical applications of deep learning in medical image analysis. Med Image Anal 79:102444. https://doi.org/10.1016/j.media.2022.102444
10. Sharma AK, Nandal A, Dhaka A, Dixit R (2021) Medical image classification techniques and analysis using deep learning networks: a review. In: Patgiri R, Biswas A, Roy P (eds) Health informatics: a computational perspective in healthcare. Springer Singapore, Singapore, pp 233–258
11. Wang Y-W, Chen C-J, Wang T-C et al (2022) Multi-energy level fusion for nodal metastasis classification of primary lung tumor on dual energy CT using deep learning. Comput Biol Med 141:105185. https://doi.org/10.1016/j.compbiomed.2021.105185
12. Zhang J, Wang Y, Liu J et al (2022) Multiple organ-specific cancers classification from PET/CT images using deep learning. Multimed Tools Appl 81:16133–16154. https://doi.org/10.1007/s11042-022-12055-3
13. Yasaka K, Akai H, Abe O, Kiryu S (2018) Deep learning with convolutional neural network for differentiation of liver masses at dynamic contrast-enhanced CT: a preliminary study. Radiology 286:887–896. https://doi.org/10.1148/radiol.2017170706
14. Choi KJ, Jang JK, Lee SS et al (2018) Development and validation of a deep learning system for staging liver fibrosis by using contrast agent–enhanced CT images in the liver. Radiology 289:688–697. https://doi.org/10.1148/radiol.2018180763
15. Chaunzwa TL, Hosny A, Xu Y et al (2021) Deep learning classification of lung cancer histology using CT images. Sci Rep 11:5471. https://doi.org/10.1038/s41598-021-84630-x
16. Mohammad AA, Sakhaei SM (2021) A fully automated deep learning-based network for detecting COVID-19 from a new and large lung CT scan dataset. Biomed Signal Process Control 68:102588. https://doi.org/10.1016/j.bspc.2021.102588
17. Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. Commun ACM 60:84–90. https://doi.org/10.1145/3065386
18. Iandola FN, Han S, Moskewicz MW, et al (2016) SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. arXiv:160207360 Cs
19. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. arXiv:14091556 Cs
20. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. arXiv:151203385 Cs
21. Szegedy C, Liu W, Jia Y, et al (2014) Going deeper with convolutions. arXiv:14094842 Cs
22. Sandler M, Howard A, Zhu M, et al (2018) MobileNetV2: inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF conference on computer vision and pattern recognition. IEEE, Salt Lake City, UT, pp 4510–4520
23. Zhang X, Zhou X, Lin M, Sun J (2017) ShuffleNet: an extremely efficient convolutional neural network for mobile devices. arXiv:170701083 Cs
24. Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, Honolulu, HI, pp 6517–6525
25. Dwivedi AK Performance evaluation of different machine learning techniques for prediction of heart disease. Neural Comput 9
26. Singh A, Pandey B Intelligent techniques and applications in liver disorders: a survey. 44
27. Schmidhuber J (2015) Deep learning in neural networks: an overview. Neural Netw 33
28. Bengio Y (2009) Learning deep architectures for AI. Found Trends® Mach Learn 2:1–127. https://doi.org/10.1561/2200000006
29. Sathananthavathi V, Indumathi G, Swetha Ranjani A (2020) Parallel Architecture of fully convolved neural network for retinal vessel segmentation. J Digit Imaging 33:168–180. https://doi.org/10.1007/s10278-019-00250-y
30. A Comprehensive Guide to Convolutional Neural Networks—the ELI5 way | by Sumit Saha | Towards Data Science. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. Accessed 3 Feb 2021
31. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv:150203167 Cs
32. Vinod N, Hinton GE (2010) Rectified Linear Units Improve Restricted Boltzmann Machines. ICML
33. Stochastic Gradient Descent—Clearly Explained!! | by Aishwarya V Srinivasan | Towards Data Science. https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31. Accessed 4 Feb 2021
34. Bishop CM (2006) Pattern recognition and machine learning. Springer, New York
35. Kingma DP, Ba J (2017) Adam: a method for stochastic optimization. arXiv:14126980 Cs
36. LiTS – Liver Tumor Segmentation Challenge (LiTS17)—Academic Torrents. https://academictorrents.com/details/27772adef6f563a1ecc0ae19a528b956e6c803ce. Accessed 3 Feb 2021
37. 3D-IRCADb 01 | IRCAD France. https://www.ircad.fr/research/3d-ircadb-01/. Accessed 29 Apr 2021
38. COVID-19 CT scans. https://kaggle.com/andrewmvd/covid19-ct-scans. Accessed 25 Nov 2021
39. Bilic P, Christ PF, Vorontsov E et al (2019) The Liver Tumor Segmentation Benchmark (LiTS). ArXiv190104056 Cs

40. Hounsfield Scale - an overview | ScienceDirect Topics. https://www.sciencedirect.com/topics/medicine-and-dentistry/hounsfield-scale. Accessed 4 Feb 2021
41. Zhao J, Zhang C, Li D et al (2020) Combining multi-scale feature fusion with multi-attribute grading, a CNN model for benign and malignant classification of pulmonary nodules. J Digit Imaging 33:869–878. https://doi.org/10.1007/s10278-020-00333-1
42. James G, Witten D, Hastie T, Tibshirani R (2013) An introduction to statistical learning. Springer, New York
43. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. AISTATS
44. Li Y, Xu X, Weng S et al (2020) CT image-based texture analysis to predict microvascular invasion in primary hepatocellular carcinoma. J Digit Imaging 33:1365–1375. https://doi.org/10.1007/s10278-020-00386-2
45. Varish N, Pal AK (2018) A novel image retrieval scheme using gray level co-occurrence matrix descriptors of discrete cosine transform based residual image. Appl Intell 48:2930–2953. https://doi.org/10.1007/s10489-017-1125-7
46. Öztürk Ş, Akdemir B (2018) Application of feature extraction and classification methods for histopathological image using GLCM, LBP, LBGLCM, GLRLM and SFTA. Procedia Comput Sci 132:40–46. https://doi.org/10.1016/j.procs.2018.05.057
47. Cortes C, Vapnik V (1995) Supportvector networks. Mach Learn 20:273–297. https://doi.org/10.1007/BF00994018
48. Freund Y, Schapire RE (1996) Experiments with a New Boosting Algorithm. ICML
49. Transfer Learning Using AlexNet - MATLAB & Simulink - MathWorks India. https://in.mathworks.com/help/deeplearning/ug/transfer-learning-using-alexnet.html. Accessed 4 Feb 2021
50. Hussain M, Bird JJ, Faria DR (2019) A Study on CNN Transfer Learning for Image Classification. In: Lotfi A, Bouchachia H, Gegov A et al (eds) Advances in computational intelligence systems. Springer International Publishing, Cham, pp 191–202

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.