An improved SqueezeNet model for the diagnosis of lung cancer in CT scans

Article · August 2022

DOI: 10.1016/j.mlwa.2022.100399

Thomas Papastergiou
Université de Montpellier

Contents lists available at ScienceDirect

Machine Learning with Applications

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

An improved SqueezeNet model for the diagnosis of lung cancer in CT scans

Michail Tsivgoulis [a], Thomas Papastergiou [a,b,*], Vasilis Megalooikonomou [a]

[a] Computer Engineering and Informatics Department, University of Patras, Rio, Patras, 26504, Achaia, Greece
[b] LIRMM, University of Montpellier, IBMM, CNRS, Montpelier, France

ARTICLE INFO

Keywords:
Deep learning
Compact convolutional neural networks
Lung nodules classification
CT classification

ABSTRACT

Lung cancer is the leading cause of cancer deaths nowadays and its early detection and treatment plays an important role in survival of patients. The main challenge is to acquire an accurate diagnosis in a limited time and without the need of massive computing power. Here, we propose SqueezeNodule-Net, a light and accurate convolutional neural network (CNN) that can rapidly classify nodules into malignant and benign, requiring only a mid-range computing system. It is based on the compact CNN model SqueezeNet and its Fire Module, whose structure we modified in two different ways and compared them with state-of-the-art models. We used 888 CT scans from the public dataset LUNA16 from which, after appropriate preprocessing, we generated 2D 50 × 50 images of benign and malignant nodules. We, also, produced 3D images in order to prove that our models can run successfully with more spatial information by using the same computing system. For 2D images, SqueezeNodule-Net V1 achieves 93.2% accuracy, 94.6% specificity and 89.2% sensitivity, while the SqueezeNodule-Net V2 achieves 94.3% accuracy, 95.3% specificity and 91.3% sensitivity. In 3D space, SqueezeNodule-Net V1 gives 94.3% accuracy, 96.0% specificity and 87.4% sensitivity, while SqueezeNodule-Net V2 gives 95.8% accuracy, 96.2% specificity and 90.2% sensitivity. Overall, compared to Squeeze-Net, SqueezeNodule-Net V1 is 1.2–1.06 times smaller, 1.31–1.5 times faster and has 0.8–2.5 better classification performance, while SqueezeNodule-Net V2 is 1.4–1.5 time larger, 0.04–1.5 times faster and has 0.1–2.7 times better classification performance.

1. Introduction

Lung Cancer is the leading cause of cancer deaths among both men and women worldwide (Siegel, Miller, Fuchs, & Jemal, 2021). Prognosis for patients is poor, with 5-year survival rate for all types of lung cancer being 21% (Siegel et al., 2021). However, early detection of lung nodules is currently one of the most effective ways to predict and treat the disease, reducing the mortality rate. Lung nodule is an early clinical sign of lung cancer and low dose Computed Tomography (CT) is widely considered to be the most used approach for early screening (Cui et al., 2020). Early detection of lung cancer significantly improves the chances for survival, but it is a challenging task to detect early stages of lung cancer as patients present a few, or even no symptoms (Choi & Choi, 2013).

Convolutional Neural Networks (CNN) have emerged in recent years as the state-of-the-art approach in computer vision tasks, such as image classification (Krizhevsky, Sutskever, & Hinton, 2012), object recognition (Su, Li, & Chen, 2021) and detection methods (Redmon, Divvala, Girshick, & Farhadi, 2015), achieving high classification performance (Krizhevsky et al., 2012; Redmon et al., 2015; Su et al., 2021). CNN is a neural network model, typically composed of three types of layers: convolution, pooling, and fully connected layers, with the first two types of layers performing feature extraction sequentially while the last maps the extracted features into the final classification output (Yamashita, Nishio, Do, & Togashi, 2018).

In the last decade, machine learning has been increasingly used for Computer-Assisted Diagnosis (CAD) to analyze medical images in various fields (Mathews, 2019). The architecture of a nodule CAD system typically consists of three stages: (1) nodule candidates' detection (2) false positive reduction and (3) nodule classification. In the nodule detection stage, the main goal is to recognize as many candidate nodules as possible, as this will contribute to an accurate diagnosis and possibly to an increase in the survival rate. This procedure results in many false positive nodules; therefore stage 2 aims at the significant reduction of false positive samples. The nodule classification stage, is the final step determining whether the nodule is malignant or benign, playing a major role in the performance of CAD systems.

There are several papers reporting various types of deep learning models aiming to improve the accuracy of CAD systems (Cao, Wu, Cao, & He, 2020). The major challenge remains the balance between classification performance, computational resources consumption and training times, with the two later being inversely proportional to classification performance. For this reason, our study, based on the deep learning model SqueezeNet (Iandola et al., 2016) that was proposed

[*] Corresponding author at: Computer Engineering and Informatics Department, University of Patras, Rio, Patras, 26504, Achaia, Greece.
E-mail addresses: up1048623@upnet.gr (M. Tsivgoulis), papastergiou@ceid.upatras.gr (T. Papastergiou), vasilis@ceid.upatras.gr (V. Megalooikonomou).

https://doi.org/10.1016/j.mlwa.2022.100399
Received 25 March 2022; Received in revised form 3 August 2022; Accepted 3 August 2022
Available online 9 August 2022

as a compact version of AlexNet, proposes a lighter model structure containing fewer trainable parameters, with comparable classification performance.

SqueezeNet is pre-trained for image classification on the ImageNet Dataset (Deng et al., 2009). A smaller CNN architecture provides advantages such as reduced communication costs in a distributed training setting and is more suitable for deployment on memory-constrained devices such as Field-Programmable Gate Arrays (FPGAs). Two variants (with a simple and a complex bypass) of SqueezeNet has been proposed (Iandola et al., 2016) with the former achieving the best classification performance. However, the purpose of our work is to make SqueezeNet even simpler, faster, and more efficient.

We introduce SqueezeNodule-Net, a compact and fast, regarding training time, model, which is a simplified version of SqueezeNet. Our model classifies 2D and 3D, preprocessed low-dose CT scans of lung nodules, in two classes: malignant and benign. SqueezeNodule-Net consists of a lighter, than SqueezNet's, Fire Module and comes in two versions, with the first one performing less "expand" and the second one less "squeeze" of the input. The first version was intended to be more compact than SqueezeNet, while the second focused mainly on achieving better classification performance. Despite that SqueezeNet was trained on the ImageNet database images (Deng et al., 2009; Iandola et al., 2016), we chose to train the networks from scratch without relying on the pre-trained weights as our classification task is targeted to lung nodules, a specific task different than the generic classification on ImageNet. The dataset used comes from LUNA16 Grand challenge (Setio et al., 2017), a publicly available dataset that contains 888 Low-dose CT scans in mhd+raw format. Data where, firstly, preprocessed using undersampling, segmentation of the lungs using threshold, region of interest centering, and augmentation. A 5-Fold Cross Validation technique was used to evaluate our models.

In the 2D setting, SqueezeNodule-Net V1 had 1.2 times fewer parameters, achieving better runtime performance (1.5 times faster) and better classification performance (0.8–2.5%) with respect to all investigated evaluation metrics, compared to SqueezeNet. SqueezeNodule-Net V2 was 1.5 times larger but also 1.5 times faster than our first version, while achieving improved classification results (0.7–2.1%). SqueezeNodule-Net V2 was compared to state-of-the-art CNNs, such as LeNet-5, DenseNet-121, ResNet-50 and VGG-11. As it was found, training times were shorter (2.1 to 7.8 times faster) and classification results were better (1.1–13.4%) in all evaluation metrics.

In the 3D data setting, SqueezeNodule-Net V1 is 1.06 times smaller, has similar classification performance, but is 1.31 times faster than SqueezeNet. SqueezeNodule-Net V2 is 1.4 times larger than SqueezeNodule-Net V1, almost similar in training times (0.04 times faster) but achieves better classification performance (0.1–2.7%). Once again, SqueezeNodule-Net V2 was evaluated against state-of-the-art models in the 3D setting. It was observed, that SqueezeNodule-Net V2 was much faster in training (from 4.5 to 21 times), having fewer trainable parameters (from 4.5 to 18.5 times), while classification results were better in all investigated evaluation metrics (0.4–6.6% more), with the exception of LeNet-5 that was 5.8 times faster and 7.8 times smaller compared to our model, but without achieving better accuracy.

For the sake of completeness, we tested our models also on benchmark datasets such as Cifar-10 and MNIST. SqueezeNodule-Net V2 had the best accuracy on Cifar-10 (82.5%) and MNIST (99.1%) datasets. SqueezeNodule-Net V1 was the next best performing model on Cifar-10 (81.5%) and third on MNIST (99.0%) as far as accuracy is concerned. It is also noteworthy that both models were approximately 1.04 to 2.62 times faster compared to the other models.

The contributions of this paper can be summarized as follows:

1. SqueezeNodule-Net V1 and V2 are proposed, by modifying SqueezeNet's (Iandola et al., 2016) core module, the Fire Module, that is more compact and faster than SqueezeNet in both 2D and 3D image models.
2. SqueezeNodule-Net V1 and V2 improve accuracy by 1.2–2.3%, specificity by 0.8–1.5% and sensitivity by 2.5–4.6% of the SqueezeNet model on 2D images of nodules.
3. SqueezeNodule-Net V1 has 15.8% fewer parameters and performs better in terms of classification and running time performance than SqueezeNet. Furthermore, despite the fact that SqueezeNodule-Net V2 has 23.0% more parameters than SqueezeNet, it outperforms SqueezeNet in terms of classification performance and requires significant less epochs to converge, performing better in terms of running time also.
4. The proposed models in the 3D images setting achieve the same or better accuracy (+1.4%), specificity (+0.1%) and sensitivity (+2.7%) than SqueezeNet3D. SqueezeNodule-Net3D V1 is 6.2% smaller in size than SqueezeNet3D as well as 1.31x faster. SqueezeNodule-Net3D V2 is 37.3% larger in size but due to fast convergence is 1.35x faster than SqueezeNet3D.
5. The proposed models outperform, in most of the cases, state-of-the-art models such as LeNet-5, VGG-11, ResNet-50 and DenseNet-121 in terms of classification and running time performance.

In Section 2 related work on the topic is covered while in Section 3 the methodology of the proposed SqueezeNodule-Net model and SqueezeNodule-Net's V1 and V2 architecture is described. The description of the dataset used, the preprocessing procedure on the specific dataset is covered in Section 4 and the results of our experiments in 2D and 3D image space are discussed in Section 5. Finally, Sections 6 and 7 conclude the paper.

2. Related work

In recent years, CNN-based methods have been used a lot to improve CAD systems. Most of the CAD systems for early lung cancer detection include three stages: (1) nodule candidate detection; (2) false positive reduction and (3) nodule classification.

Segmentation of lungs, (i.e. detecting and isolating the lung lobes), and lung nodules detection (i.e. the detection of presence and the determination of the region of interest of a nodule) is an important procedure in a CAD system when it comes to the diagnosis of malignant lung nodules (Lee, Kouzani, & Hu, 2012). Various methods have been proposed including traditional methods (Kamble, Sahu, & Doriya, 2020) as well as deep learning methods (Gite, Mishra, & Kotecha, 2022). Classical methods for lung segmentation include 3D Region Growing (Medeiros da Nobrega, Rodrigues, & Filho, 2017), Non-Negative Matrix Factorization with constraints (Hosseini-Asl, Zurada, Gimelfarb, & El-Baz, 2016), Adaptive Crisp Active Contours (Rebouças Filho, Cortez, da Silva Barros, Albuquerque, & Tavares, 2017), Automatic Lung Segmentation using thresholds and morphologic operations (Nery, Silva, Ferreira, & Caramelo, 2012) etc. Furthermore, lung segmentation methods can be classified in three major categories fowling (Shaukat, Raja, & Frangi, 2019): deformable boundary-based techniques, edge-based techniques and threshold-based techniques. For an elaborated comparison of classical techniques, the interested read can refer to (Shaukat et al., 2019).

In the frame of deep learning methods various approaches have been proposed (Gite et al., 2022; Krizhevsky et al., 2012). Deep neural networks architectures like 2D CNNs (employing a U-net architecture for segmenting lungs from X-rays) as in (Gordienko et al., 2019) or 3D CNN networks as in (Alakwaa, Nassef, & Badr, 2017; Hamidian, Sahiner, Petrick, & Pezeshk, 2017) where a 3D CNN is trained for

Table 1
Performance comparison of state-of-the-art models on the LUNA-16 dataset.

Method	Acc.	Spec.	Sens.	CPM	Remarks
(Yuan, Fan, Wu, & Cheng, 2021)	0.956	0.998	0.812	0.881	Training Time =∼48 h
(Zhang, Lin, & Wang, 2021)	0.924	0.96	0.87	–	–
(Zhang et al., 2020)	0.9167	–	–	–	–
(Mastouri, Khlifa, Neji, & Hantous-Zannad, 2021)	0.9199	0.9227	0.9185	–	–
(Liao, Liang, Li, Hu, & Song, 2019)	0.8142	–	–	–	AUC = 0.87
(Utkin, Meldo, Kovalev, & Kasimov, 2019)	0.918	–	0.908	–	Precision = 0.926
(Wang & Chakraborty, 2019)	–	–	–	–	Training Time per epoch = 4 h

detecting automatic lung nodules by using volumes of interest and then converted to a 3D Full convolutional Network that can generate efficiently in a single pass the scores for the entire volume, have be proven very efficient. U-net has been employed efficiently for lung segmentation in different contexts e.g. for tuberculosis detection (Rahman et al., 2020) or for quantification of COVID-19 infections from CTs (Diniz et al., 2021). Furthermore, a 3D model (V-net) combined with a Spatial Transformation Network has been employed for pulmonary parenchyma segmentation (Zhao et al., 2021). These segmentations where then employed for extracting radiomics features from the segments in order to distinguish between COVID-19 from community acquired pneumonia and healthy subjects. Finally, U-net combined with Variational Autoencoders have been employed for lung segmentation from X-rays images, with promising results (Cao & Zhao, 2021).

For the nodule classification problem, Yang, Yu, and Wang (2016) developed a 2D deep Convolutional Neural Network (CNN) using 5 different datasets. The datasets were a combination of real and artificial nodules they created in order to verify the role of enlarging the dataset and the relevance of artificial nodules. Furthermore, Zhang, Zhang, Zhao, Wei, and Zhang (2018) used a current ResNet and 2.5D view of CT Images. The 2.5D view images are constructed by the 2D images, resampled from the original 3D CT images, through three different planes, i.e., coronal plane, sagittal plane and axial plane, converting to a RGB three-channel image. Also, Xie et al. (2019) performed a 3D multiview knowledge-based collaborative (MV-KBC) deep model, which was built with three pre-trained ResNet-50 networks, for extracting multiple features from nine planes and diagnosing malignant nodules. Overall, in recent years, several algorithms have been developed for CAD system optimization. The interested reader is referred to the recent review papers (Cao et al., 2020; Li, Xiao, Huang, Hassan, & Huang, 2022).

The LUNA-16 dataset, used in this study, was lunched for the Lung Nodules Analysis 16 (LUNA16) challenge, consisting of two different tasks: (1) a complete system for nodule detection and (2) a false positive reduction task (i.e. a classification task on given Region of Interest (ROIs) (Setio et al., 2017). For the nodule classification task many recent studies have appeared using the LUNA-16 dataset. Yuan et al. (2021) proposed a 3D convolutional network, for nodule vs. non-nodule classification, that extract spatial information using three paths of different field sizes that are fused later in the model. Although they reported a Competitive Performance Metric (CPM) score of 0.881, Recall of 0.812 Specificity 0.891 and Accuracy 0.956, their model needed about 48 h of training time and used 10,855 MB of memory. Zhang et al. (2021) propose a DenseNet architecture comprising 3D filters and pooling kernels along with data augmentation, for the malignant vs. benign classification task. They report an accuracy of 92.4%, a specificity of 96.0% and a sensitivity of 87.0%. Zhang et al. (2020) proposed a benign vs. malignant nodules classification architecture using a squeeze-and-excitation network along with aggregated residual transformations (SE-ResNeXt) modules, for features recalibration and features reuse. They report an accuracy of 91.67% and an AUC of 0.9563. Mastouri et al. (2021) proposed Bilinear CNN (BCNN) structures combined with a linear SVM for nodules classification. They used [VGG16, VGG19] structures, fine-tuned on the dataset (feature extraction), to build the bilinear architectures and they report an accuracy of 91.99% a specificity of 92.27% and a sensitivity of 91.85% for the nodule vs. non-nodule classification task. Liao et al. (2019) proposed a two modules 3D deep U-net-like network architecture: the first module detects and indicates the suspicious nodules of a patient, while the second selects the top 5 nodules, according the detection confidence, calculates the probabilities that the nodules are cancerous and outputs the cancer probability of each subject in a Multiple Instance Learning frame, using a leaky noisy-OR gate. The reported classification accuracy and AUC are 81.42% and 0.87 respectively. In (Utkin et al., 2019) an ensemble of triple neural networks is proposed, for taking into account atypical cases of cancer, that operate on 5 histograms characterizing the shape, the inner and outer structures of the nodules, extracted from the nodules' segments. The reported classification results are an accuracy of 91.8%, a Sensitivity of 90.8% and a Precision of 92.6%. Finally, Wang and Chakraborty in (Wang & Chakraborty, 2019) propose a hierarchical (sliced) model of recurrent neural networks (RNN), where different layers can be trained hierarchically, with an addition of selective attention (alignment). They report in their results that the proposed network converges much faster in comparison to a 3-D CNN, with the training time per epoch being 4 h, in comparison of 12 h training per epoch that needs the 3D-CNN. They, finally, state that both networks need 10 epochs for full convergence. Table 1 summarizes the comparison of the results of State-of-the-art methods on the LUNA-16 dataset for the task of the classification of lung nodules.

An important factor in the overall efficiency of the models is the consumption of resources and time. A complex and large model has the limitation of requiring large computational effort. SqueezeNet was invoked to solve this problem so that it can also run on mobile systems. Initially, it was used to classify 227 × 227 × 3 images from the ImageNet dataset, with similar accuracy to the well-known AlexNet model (Iandola et al., 2016), but there exist some papers where it is used for other types of images, such as medical images. Some of these works present a modified version of Squeeze-Net. For instance, Ucar and Korkmaz (2020) developed a deep SqueezeNet for rapid diagnosis of Covid-19 using Bayesian optimization, for tuning hyperparameters, to improve accuracy. Polsinelli, Cinque, and Placidi (2020) also worked on Covid-19 using a modified SqueezeNet to improve convergence and overfitting by adding a Batch Normalization layer between the squeeze convolution layer and the ReLU layer. Furthermore, Özyurt, Sert, and Avcı (2020) used SqueezeNet to extract features in MRI images for brain tumor classification, in combination with the SR-FCM-CNN hybrid method. Mobeen-ur Rehman Khan, Abbas, and Danish Rizvi (2019) used pre-trained SqueezeNet and compared it to VGG19 and AlexNet for classification of diabetic retinopathy images. Finally, Santos et al. (2018) reduced the size of SqueezeNet by modifying the 3 × 3 convolutions in the SqueezeNet's Fire Modules and turning them into Depthwise Separable Convolution.

Besides the special design of neural networks aiming at optimizing the consumption of resources and time, like MobileNet (Howard et al., 2017), ShuffleNet V2 (Ma, Zhang, Zheng, & Sun, 2018) or in (Zhao, Sugiyama, Yuan, & Cichocki, 2019) by optimizing their microarchitecture (e.g., kernel types, low rank approximation etc.), or like Mnasnet (Tan et al., 2019) or FBnet (Wu et al., 2019) by optimizing their macro-architecture (e.g., module types) various other approaches have been proposed that can be categorized into four major classes:

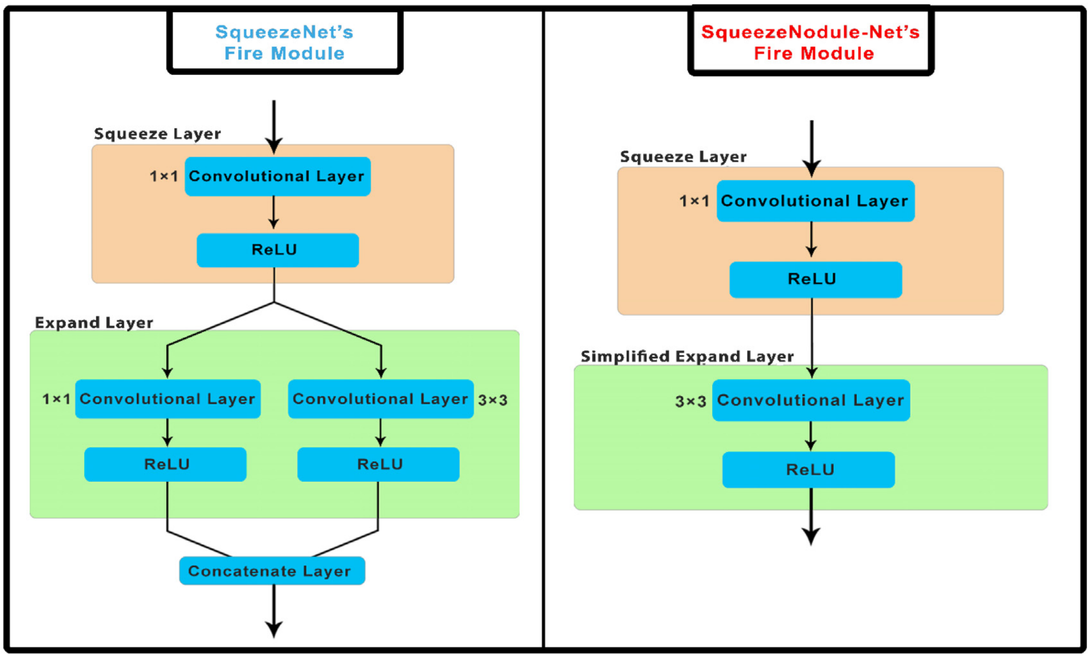


Fig. 1. The different internal structure of the Fire module in SqueezeNet and SqueezeNodule-Net model.

(1) parameter pruning and quantization, (2) low-rank factorization, (3) transferred/compact convolutional filters and (4) knowledge distillation (Cheng, Wang, Zhou, & Zhang, 2017). Pruning can help to tackle the overfitting problem, regularize and decrease the complexity of o a network (Gong, Liu, Yang, & Bourdev, 2014). Proposed methods include works for reducing the number of connections based on the Hessian of the loss functions (Hassibi & Stork, 1992), for pruning non-informative weights in pre-trained models (Han, Pool, Tran, & Dally, 2015) or in training compact models with sparsity constraints (Wen, Wu, Wang, Chen, & Li, 2016) where sparsity constrains are used on each layer for reducing redundant filters, channels or layers. Furthermore, network quantization refers in the compression of a network by reducing the amount of bits used for representing their weights. In (Wu, Leng, Wang, Hu, & Cheng, 2016) the authors used *k*-means scalar quantization for the parameter values, Vanhoucke et al. in (Vanhoucke, Senior, & Mao, 2011) show that quantizing the weight parameters to 8-bit is beneficial in terms of speeding up the training time of the network without a significant decrease in the classification performance. Furthermore, in (Choi, El-Khamy, & Lee, 2016) the authors use the Hessian weights for measuring the importance of the network parameters and optimize Hessian-weighted quantization errors for clustering the network parameters. Another quantization approach come from (Rastegari, Ordonez, Redmon, & Farhadi, 2016) where the authors propose XNOR-Net, a network that directly learn binary weights and activations. For a comprehensive review on the quantization methods for efficient neural network inferences the interested read may refer to (Gholami et al., 2021). Finally, we present some real-life applications where compact Deep Neural Networks are applied in various settings. In (Chen, Lee, Sritapan, & Lin, 2016) the authors trained a 16-layer CNN on iOS mobile devices on the ImageNet database without compromising the classification performance. Wu et al. (2016) and Yin et al. (2019) tackled the image classification problem using Deep Neural Networks on mobile devices using quantization, Nagel, Baalen, Blankevoort, and Welling (2019) applied quantized Deep Neural Networks for semantic segmentation and object detection while Mishchenko et al. (2019) used quantized CNNs for small-footprint keyword spotting. The interested reader may refer to (Nalepa, Antoniak, Myller, Lorenzo, & Marcinkiewicz, 2020) for an detailed review on quantized models and applications.

3. Material and methods

3.1. SqueezeNodule-Net architecture

Here we introduce the SqueezeNodule-Net architecture that comes in two versions. A lighter version SqueezeNodule-Net V1 and a heavier version SqueezeNodule-Net V2. The key features of the SqueezeNodule-Net architecture are described below. The key element to reduce the model's total parameters is the Fire Module. As shown in Fig. 1, the Fire Module of SqueezeNodule-Net (simpfire) and SqueezeNet consists of 1 × 1 Convolution instead of 3 × 3 Convolution that is employed by AlexNet, resulting in a 1/9 reduction of the parameters for both models. This layer is known as the Squeeze layer. In SqueezeNet this layer is feeding an Expand layer (Fig. 1, left Fire Module) of two convolution layers with 1 × 1 and 3 × 3 filters, respectively, that increases the number of channels of the Squeeze layer. This is called a bottleneck structure. However, in the proposed model the Expand Layer of the Fire Module has been modified by reducing the number of 1 × 1 Convolutional Layer of the Expand Layer to zero as illustrated in Fig. 1 (right Fire Module).

Intuitively, 1 × 1 convolutions are operating on the filters' space without altering the spatial dimensionality of the input layer, since practically they consist of a scalar (1 × 1) convolution, and they can be used either to compress (compute reductions (i.e. embeddings) on filters), if the number of the filters used is less than the dimensionality of the filters' space of the input or to expand an input (i.e. mapping the input to an output of higher dimensionality on the filters' space), if the number of filters used in the convolution is higher than the filters' space dimensionality of the input (Szegedy et al., 2014). In the case of SqueezeNet's Fire Module as well as in the proposed lighter Fire Module (simpfire), the first Squeeze Layer computes a projection of the input in the filter's space, in a lower dimensionality filter space, acting as computing embeddings in the filters' space. The second Expand Layer of SqueezeNet computes two different expansions: one the in spatial and in the filters' space with a 3 × 3 convolution (as in the proposed lighter Fire Module) and an expansion on the filters space, by 1 × 1 convolution, resulting, after the concatenation layer, to an output of twice the number of filters used in the 1 × 1 and 3 × 3 convolutions. In other words, the embedded input is both expanded in the filters' and in the spatial space by the 3 × 3 convolution and in the filters' space by the 1 × 1 convolution. In contrary, in the proposed simpfire module the expansion occurs only by the 3 × 3 convolution in both the spatial and the filters' space. In this way, none information is lost, since the embedded from the Squeeze layer information on the filters' space, as well as the spatial information is expanded by the 3 × 3 Convolution. As observed experimentally (see Section 5) SqueezeNet's second expansion (1 × 1 convolution of the expand layer) can be omitted, since this does not affect the classification performance of the model, but instead contributes to a significant reduction of the parameters' space and consequently to reduced running times.

To be more explicit, the necessary equations of SqueezeNet (Iandola et al., 2016) for the proposed models need to be modified. For the image information to be compressed the number of filters in the Squeeze Layer must be less than the total number of filters in the Expand Layer. This can be expressed by the inequality:

$$s_{i,1\times1} < e_{i,3\times3} \tag{1}$$

where, $s_{\mathrm{i},1\times1}$ and $e_{\mathrm{i},3\times3}$ denote the number of filters of the Squeeze and the Expand layers respectively, with i indicating the Fire Module to which we are referring to. Inequality (1) results in a limitation on the number of input channels. The ratio between the number of filters in Squeeze and Expand layers, named the Squeeze ratio (SR) is defined by:

$$SR = \frac{s_{i,1\times1}}{e_{i,3\times3}} \tag{2}$$

Eq. (2) suggests that smaller SR values result to larger bottleneck in the corresponding Fire Module.

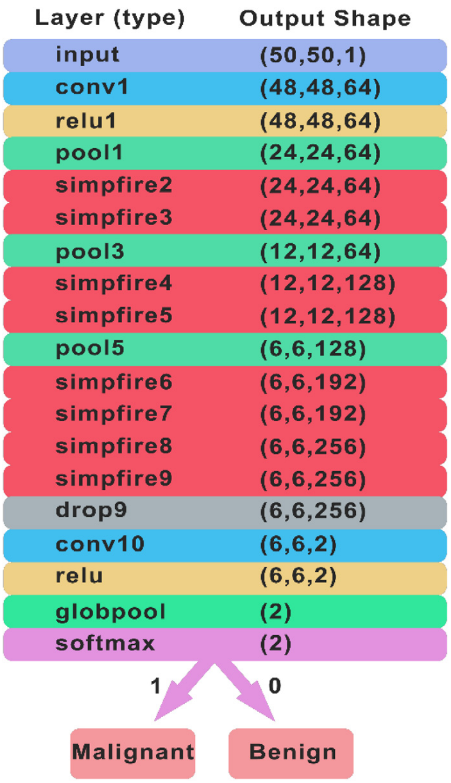
The number of the Expand filters of the Fire Module i is:

$$e_i = base_e + (incr_e * \left| \frac{i}{freq} \right|) \tag{3}$$

with $base_e$ indicating the number of the Expand filters in the first Fire Module and $freq$ the frequency of changing the number of the Expand filters in the subsequent Fire Modules (e.g., if $freq = 2$, the number of filters will be changed every 2 Fire Modules). Finally, $incr_e$ is the number of additional Expand filters added after a Fire Module (i.e., after every $freq$ Fire Modules, the number of Expand filters is increased by $incr_e$). In the Expand layer of the Fire Modules of the proposed model, there are only 3 × 3 filters. It is true that: $e_i = e_{i,3\times3}$

The structure of the proposed architecture is shown in Fig. 2. At the beginning of SqueezeNodule-Net V1 a convolution layer (conv1) is located, followed by 8 simpfire modules (simpfire2-simpfire9) resulting in a convolution layer (conv10). Each convolution layer is followed by a ReLU layer.

The number of filters in each simpfire module gradually increases, as a consequence of Eq. (3). SqueezeNodule-Net's downsampling, late in the network, results in convolution layers with large activation maps, unlike traditional architectures (e.g., VGG) that use early downsampling. Large activation maps result in higher classification accuracy given the same number of parameters. At the end of the model instead of using as usually, Fully Connected layers, we employ Convolutional Layers, since Fully Connected layers have a large number of parameters compared to convolutional layers and are prone to overfitting. For this reason, in the proposed model, like in the SqueezeNet model, a global average pooling layer is used, which computes a channel-wise average over the last convolutional layer. Furthermore, Max Pooling layers are used, as indicated in Fig. 2, and since they do not have any trainable weights, they do not contribute to the model's size. Furthermore, Max Pooling layers tend to reduce overfitting of the model (Nirthika, Manivannan, Ramanan, & Wang, 2022). Finally, we use a "softmax" activation function in the output layer to ensure that the output values are in the range 0 to 1. SqueezeNodule-Net's Fire module (simpfire) consists of 2 groups of hyperparameters, namely $s_{1\times1}$ and $e_{3\times3}$, and since there exist 8 simpfire modules, the proposed model results in a total of 24 groups of trainable parameters. Since it is a binary classification task the last Convolutional Layer (Conv10) consists of 2 filters.



Layer (type)	Output Shape
input	(50,50,1)
conv1	(48,48,64)
relu1	(48,48,64)
pool1	(24,24,64)
simpfire2	(24,24,64)
simpfire3	(24,24,64)
pool3	(12,12,64)
simpfire4	(12,12,128)
simpfire5	(12,12,128)
pool5	(6,6,128)
simpfire6	(6,6,192)
simpfire7	(6,6,192)
simpfire8	(6,6,256)
simpfire9	(6,6,256)
drop9	(6,6,256)
conv10	(6,6,2)
relu	(6,6,2)
globpool	(2)
softmax	(2)

Fig. 2. The SqueezeNodule-Net for 2D grayscale input (50 × 50 × 1).

In the 3D setting, the structure of the SqueezeNodule-Net3D is similar, with the only difference being the use of 3D Convolution and Max Pooling Layers in order to accept 3D input.

3.1.1. SqueezeNodule-Net V1

Having mentioned the above, in the proposed model SqueezeNodule-Net V1, we define the following: $base_e = 64$, $s_{\mathrm{i},1\times1} = 16$, $incr_e = 128$, $freq = 2$ and $SR = 0.25$.

Unlike SqueezeNet, $base_e = 64$ is used instead of 128, because, as it will be shown experimentally, less expanding the input contributes to better runtime and classification performance. The Expand Layer of the proposed Lighter Fire Module employs only 3 × 3 Convolutional Layers, which implies that the SR will be equal to 0.25, (e.g., in the first simpfire module the Squeeze Ratio is $\mathrm{SR} = \frac{\mathrm{s}_{\mathrm{i},1\times1}}{\mathrm{e}_{\mathrm{i},1\times1}+\mathrm{e}_{\mathrm{i},3\times3}} = \frac{16}{0+64} = 0.25$).

3.1.2. SqueezeNodule-Net V2

In order to further normalize the bottleneck, a second version is delivered, which employs a heavier Squeeze Layer ($s_{i,1\times1} = 32$). The incoming information in the simpfire module is increased, a fact that leads to even less information needed to be retrieved by the Expand Layer. This results in an increase in the SR, i.e., $SR = \frac{s_{i,1\times1}}{e_{i,1\times1}+e_{i,3\times3}} = \frac{32}{0+64} = 0.50$.

The last 2 simpfire modules have been set to have SR=0.25. By decreasing the SR of the last two simpfire modules we end up with a model employing 23% more parameters than SqueezeNet. On the contrary, by preserving the $SR = 0.50$ in the last two simpfire modules we end up with 68% more parameters than these of SqueezeNet. As will be shown in experiments, this choice did not affect the run time and classification performance, since less training epochs were needed to achieve better classification performance.

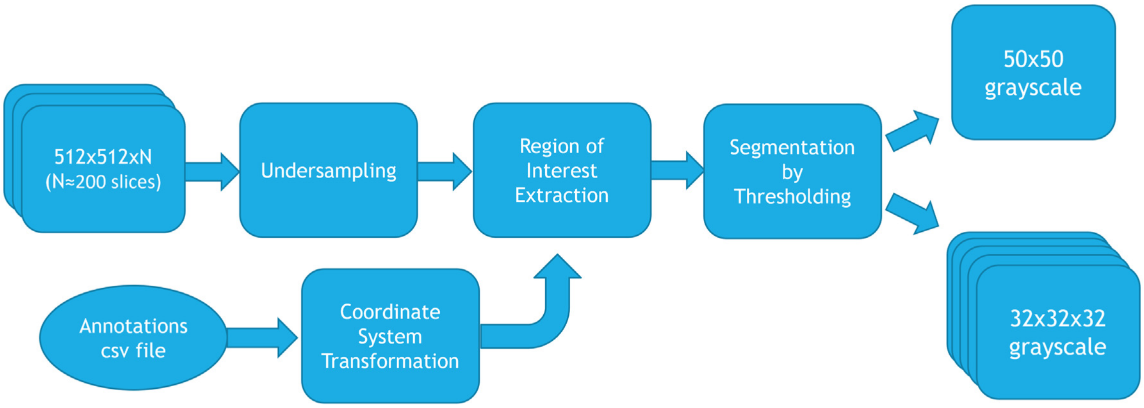

512x512xN
(N≈200 slices)
Undersampling
Region of Interest Extraction
Segmentation by Thresholding
50x50 grayscale
32x32x32 grayscale
Annotations csv file
Coordinate System Transformation

Fig. 3. Workflow of the preprocessing phase before the augmentation procedure.

4. Experiments

In this section, the dataset, the experimental setup, and the evaluation of the proposed SqueezeNodule-Net (V1 and V2), versus state-of-the-art classical and "light" architectures (e.g., the "light" CNN proposed by Polsinelli et al. (2020) for the diagnosis of Covid-19 disease) are presented. All experiments were conducted using the Jupyter Notebook employing Keras integrated in Python 3.7.9 (Chollet et al., 2015) for implementing the deep learning frameworks using Tensorflow as backend engine. The experiments were executed in an Intel® Core™ i7 4770 CPU, NVIDIA GeForce GTX 1650 4 GB GPU and 16,0 GB of RAM workstation.

The rest of the section includes the presentation and preprocessing of the LUNA16 dataset (as shown in Fig. 3), the definition of the evaluation metrics for the proposed models in the 2D and 3D setting.

4.1. LUNA16 dataset

The dataset used for the binary classification task (benign vs malignant nodules) was LUNA16 (Setio et al., 2017), a subset of the publicly available LIDC/IDRI dataset from the Cancer Imaging Archive, where CT scans with slice thickness greater than 2.5 mm have been excluded. More specifically, the LUNA16 dataset consists of 888 CT Scans, with about 200 slices in each CT scan. The annotations of the nodules were delivered by 4 experienced radiologists. Each radiologist marked lesions identified as non-nodule, and nodules < 3 mm and ≥ 3 mm. All nodules greater than 3 mm accepted by 3 out of 4 radiologists have been included, while nodules smaller than 3 mm have been excluded from the dataset. The annotations were saved in a CSV file, with each line indicating the 3D cartesian coordinates and the corresponding diameter in mm of the nodule. From a total of 551,065 annotations 1351 (0.24%) were labeled as positive samples (malignant nodules), while the rest were labeled as negative samples (benign nodule or non-nodule).

4.2. Undersampling

The LUNA16 dataset is highly unbalanced therefore, the positive and negative samples are balanced by employing a undersampling procedure: negative samples that constitute 25% of the total dataset have been selected by uniform distribution randomly, resulting in 4053 negative samples. Even with the undersampling used, the positive-negative samples ratio is not close to 1, a fact that will be addressed by data augmentation.

4.3. Segmentation of the Lungs by thresholding

After undersampling, there is a need to remove the parts of the CT scan that are not necessary for the training of the model. This is done by removing from the CT scan the redundant parts (e.g., bones) and maintain parts such as air, lung tissue, water, blood etc. By using this procedure, the contrast between the nodule and its surrounding area is increased, which is expected to increase the performance of the model, since the nodule will be highlighted against the surrounding tissue. Since, the intensity of the CT scan image is set in Hounsfield Units and each intensity interval corresponds to specific parts of the body (such as lung, liver, bone, water, etc.), using a thresholding procedure (Alakwaa et al., 2017; Khumancha, Barai, & Rao, 2019), the redundant parts of the CT scan can be removed. The radiodensity for the lung tissue is -500. Anything over 400 is redundant since these regions correspond to bones under different radiation (Alakwaa et al., 2017). Using the thresholding process (Alakwaa et al., 2017; Khumancha et al., 2019) intensity interval of interest is kept from −1000 to 400. In Fig. 4, we can observe a slice of the lung CT scan having a nodule, the nodule before thresholding and the nodule after thresholding. The upper row corresponds to a benign nodule and the lower row to a malignant nodule.

4.4. Detection of region of interest

The dimensions of each CT scan in the dataset were $512 \times 512 \times n$, with n being the number of axial scans. The ROIs were identified according to the coordinates in the annotation's file. The annotations were in Cartesian coordinates and were converted to Voxel coordinates, using SimpleITK library. The final grayscale images centered around the nodule, were of size 50 × 50. We also generated 3D images of 32 × 32 × 32 voxels, for feeding the 3D models, by including 16 slices before and 16 slices after the annotation of the radiologists. 32 × 32 × 32 voxels were used, since 95.33% of the nodules could fit in such a bounding box (Li, Cao, Zhao, & Wang, 2016).

4.5. Data augmentation

After undersampling, positive samples represent about 25.0% of the dataset. Hence, a data augmentation procedure was necessary. Different data augmentation methods have been employed: (1) augmentation of the positive class by transformation, (2) symmetrical augmentation on both classes and weighted training, (3) augmentation of the positive samples by artificial produced positive nodules by Generative Adversarial Networks (GAN) and finally (4) progressive image resizing.

4.5.1. Augmentation of the positive class by transformation

We augmented the positive (malignant) class by rotating the positive samples by 90 degrees. After the augmentation the positive samples represented about 40% of the training and validation set.

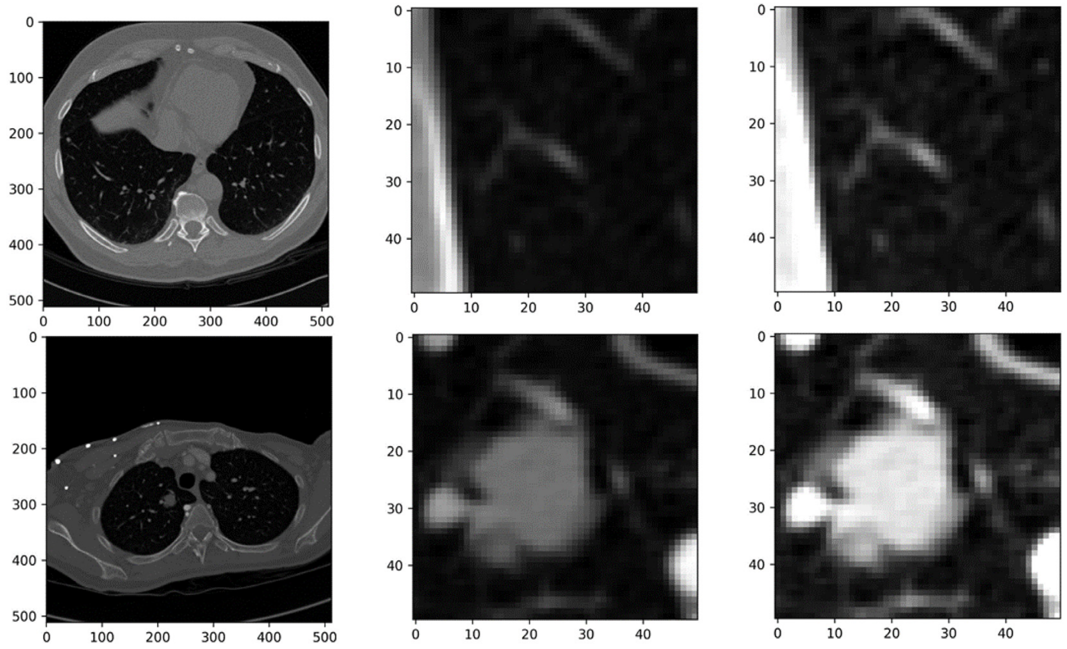


Fig. 4. Upper row: (1) slice of lung with benign nodule, (2) the benign nodule before thresholding, (3) the benign nodule after thresholding, Lower row: (1) slice of lung with malignant nodule, (2) the malignant nodule before thresholding, (3) the malignant nodule after thresholding.

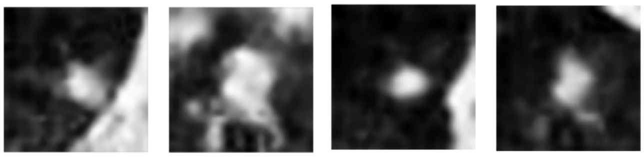


Fig. 5. Malignant nodules generated by the DCGANN.

4.5.2. *Symmetric augmentation of both classes and weighted training*

We augmented both classes symmetrically by generating new data by rotating each sample by 90 degrees. This results in an unbalanced data set with a ratio of 25:75. In particular benign nodules are overrepresented compared to malignant nodules. A way to address the class imbalance is to use the weighted training. Class weighting adjusts the cost function of the model so that misclassifying an observation from the minority class is penalized more heavily than misclassifying an observation from the majority class. The calculation of the optimal weight coefficients is done using the sklearn library tools. This procedure was implemented for all the models shown in Tables 3 and 4.

4.5.3. *Augmentation by artificially created samples using GANs*

Another technique to deal with imbalanced data is to generate new data artifacts based on the real data. Inspired by Chaudhari, Agrawal, and Kotecha (2020), we used a data augmentation technique for populating the underrepresented positive class. We used a deep learning model as proposed by Radford, Metz, and Chintala (2015), known as DCGAN (Deep Convolutional Generative Adversarial Networks). The model consists of two models: a generator and a discriminator. The generator produces synthetic images that resemble the training images (in our case malignant nodules). It takes a random noise vector in the latent space and maps it to the data space (i.e. gray scale image). The generator starts with a dense (i.e. fully-connected layer), followed by a series of transpose convolution, batch normalization and the leaky relu activation function to end up to a convolution layer with one filter and a tanh activation function for generating grayscale images of 32x32x1 pixels. The discriminator is a simple binary classification network that takes both the real and the fake images and outputs a probability of whether the given image is real or fake. For this purpose, a series of strided-convolution layers are used with leaky relu and the dropout of 0.25. Next, the feature maps are flattened and a dense layer with 1 unit is used. Finally, the sigmoid activation function is applied to the fully connected layer. The model was trained using 1081 malignant nodules for 20k epochs in order to produce artificially malignant nodules for augmenting the positive (malignant) class. At the end the positive class represented 40% of the dataset, as in all augmentation techniques that we used. Fig. 5 displays some artificially generated malignant nodules by DCGAN.

4.5.4. *Progressive image resizing*

Progressive Image Resizing is a transfer learning technique to sequentially resize all the images while training the CNN models on smaller to bigger image sizes (Bhatt, Ganatra, & Kotecha, 2021). In fact, we choose to train our model for 25 × 25 × 1 nodule images and use its weights in the next model that was fed with 40 × 40 × 1 images. Finally, the new weights are used for the model with input 50x50x1 images. In this manner, each larger scale model incorporates the previous layers and weights learned in the smaller scale model, and thus results in refining the final model as well as increasing the classification performance (Bhatt et al., 2021). We used this technique on the SqueezeNet, SqueezeNodule-Net V1 and SqueezeNodule-Net V2 models and not in models that due to their architecture design was not possible to be applied.

4.6. *Evaluation metrics*

For the evaluation of the models, 5-fold Cross Validation (CV) was used. For each fold a validation set was extracted from the Training Set (TS) (10% of the TS), for applying an early stopping procedure. Four statistical measures were used for evaluating the binary classification performance of the models: accuracy, sensitivity, specificity, and G-mean (i.e., the geometric mean of sensitivity and specificity).

4.7. *Evaluation setting*

In all models, the Adam (Kingma & Ba, 2017) gradient descent optimization algorithm was used with a learning rate of 10^{-4}, and the

Table 2
SqueezeNet, SqueezeNodule-Net V1 and V2 performance.

CNN	Acc.	Spec.	Sens.	G-mean	Total Params	Inf. Time (s)	FLOPS	Tr. Time (min.)	Mean epochs
SqueezeNet	92.0% (0.023)	93.8% (0.021)	86.7% (0.034)	90.1% (0.027)	772 K	0.28	50.4M	2.5	50.4
Light CNN [16]	93.7% (0.026)	96.4% (0.013)	85.5% (0.065)	89.5% (0.042)	1804 K	0.74	107.2M	1.8	28
ShuffleNet	85.9% (0.003)	90.6% (0.053)	78.4% (0.069)	84.27% (0.032)	957 K	1.02	**9.6M**	1.75	**21**
Mobile Net v2	90.3% (0.002)	93.8% (0.038)	79.9% (0.016)	86.3% (0.002)	2259 K	0.32	20.9M	1.4	**21**
SqueezeNodule-Net V1	93.2% (0.014)	94.6% (0.012)	89.2% (0.027)	91.8% (0.017)	**607 K**	**0.17**	43M	1.7	51.4
SqueezeNodule-Net V2	**94.3%** (0.019)	**95.3%** (0.016)	**91.3%** (0.030)	**93.2%** (0.022)	888 K	0.18	68.6M	**1.2**	38.4

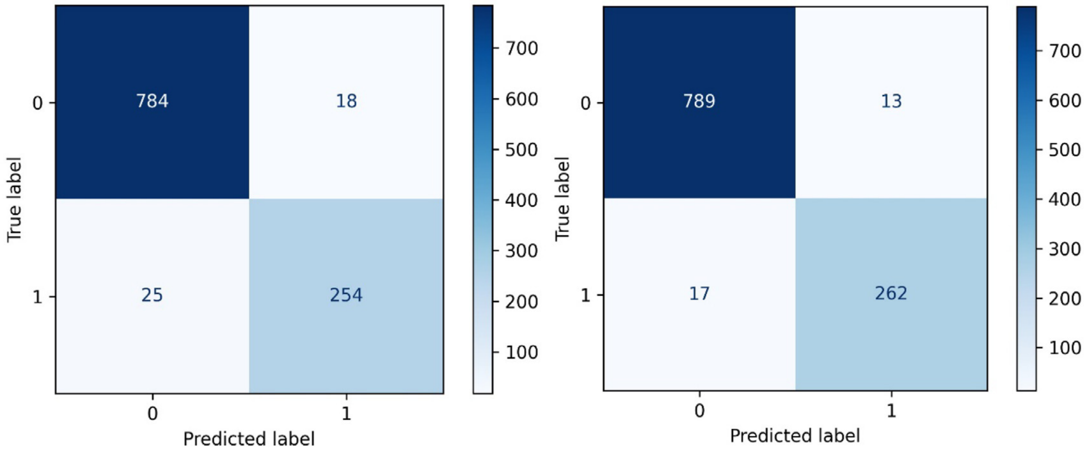


Fig. 6. Confusion matrices of the 2D models of SqueezeNodule-Net V1 (left) and SqueezeNodule-Net V2 (right).

Binary cross entropy loss function:

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log\left(p\left(y_i\right)\right) + (1 - y_i) \cdot \log(1 - p\left(y_i\right)) \qquad (4)$$

with y_i representing the label of sample i and $p\left(y_i\right)$ the predicted probability of sample i being positive. Unless otherwise stated, the batch size was set to 96 for all experiments. An early stopping procedure monitoring the validation loss was used, with patience parameter of 7 epochs. Also, a "reduce learning rate in plateau" technique was used with a patience factor of 5 epochs and a reducing learning rate factor of 0.1, so that if there is no improvement of the model for 5 consecutive epochs the learning rate is reduced by a factor of 0.1. Finally, it should be noted that all models were trained from scratch without using any pre-trained weights.

5. Results

5.1.

5.1.1. Results for the 2D models

In the first experiment SqueezeNodule-Net V1 and V2 were evaluated against SqueezeNet, and a Light CNN based on SqueezeNet (Polsinelli et al., 2020) on the 2D dataset. Table 2 presents the average values of the 5-fold CV along with their standard deviation value of the classification performance metrics. Furthermore, Table 2 presents the Total Parameters, the Training and Inference time, the Flops and the mean value of epochs needed for each model to converge. Fig. 6 presents the confusion matrices of the test set for SqueezeNodule-Net V1 (left) and V2 (right) for the 2D setting.

For the 2D dataset, the proposed models perform better than SqueezeNet in all metrics. In fact, SqueezeNodule-Net V2 model has the highest accuracy (94.3%), specificity (95.3%) and sensitivity (91.3%). In terms of evaluation metrics SqueezeNodule-Net V2 performs better than V1 and SqueezeNet models. The Light CNN (Polsinelli et al., 2020) is quite competitive and situated between SqueezeNet and SqueezeNodule-Net V1 but inferior to SqueezeNodule-Net V2 in terms of classification performance.

Table 2 shows that the proposed models are more compact and faster than SqueezeNet and Light CNN (Polsinelli et al., 2020) model. In particular, SqueezeNodule-Net V1 has 15.8% fewer parameters and is about 1.5× faster than SqueezeNet. While V2 model has 23.0% more parameters and yet it is still 2.1× faster than SqueezeNet, as it takes 12 epochs fewer to converge. Light CNN is a larger model than the proposed (SqueezeNodule-Net V1 has about 66% and V2 about 51% fewer parameters). Light CNN seems to converge faster but takes twice as long running time for each epoch, resulting in almost the same running time compared to SqueezeNodule-Net V1 while V2 is 1.5× faster the Light CNN. Finally, Figs. 7 and 8 present misclassified (False Positive and False Negative) nodules of SqueezeNodule-Net V1 and V2.

In Fig. 9 we can observe where the SqueezeNodule-Net V1 model focuses for malignant and benign samples respectively. On the left are the saliency maps and on the right are the samples which are input to the model and in the middle are both, one on top of the other, for comparison. The intensity of saliency is everywhere, however the saliency intensity concentrates around the nodule, which we want to identify. Finally, we note that because the benign nodule is not distinct, the intensity is more diffuse than in the malignant one but remains concentrated around the nodule.

5.1.2. Comparison with other 2D CNNs

In the next experiment, a comparison of the performance of the best version, SqueezeNodule-Net V2, with state-of-the-art CNN models

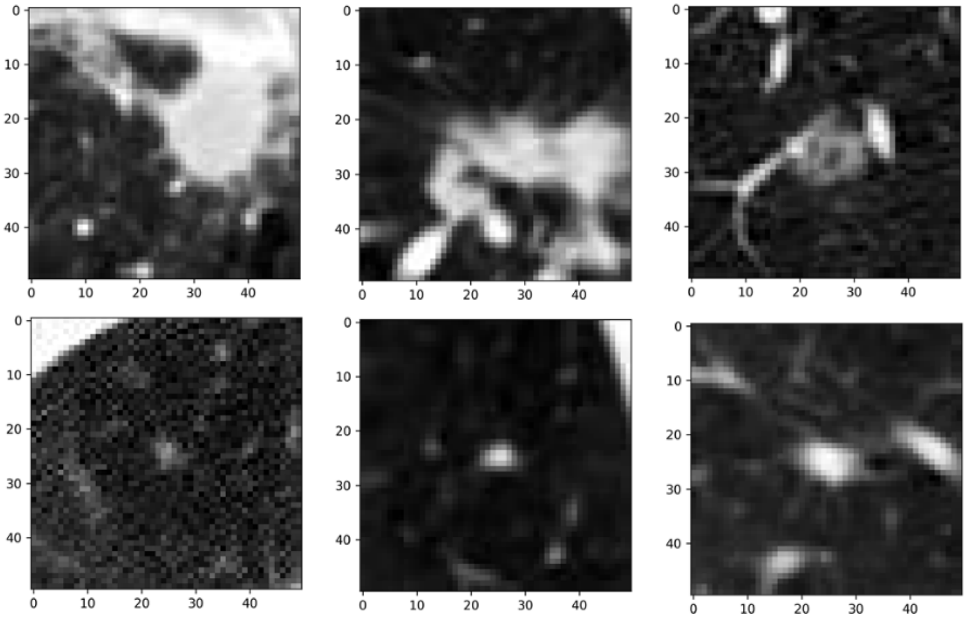


Fig. 7. False Negatives (upper row) and False Positives (lower row) predicted by SqueezNodule-Net V1.

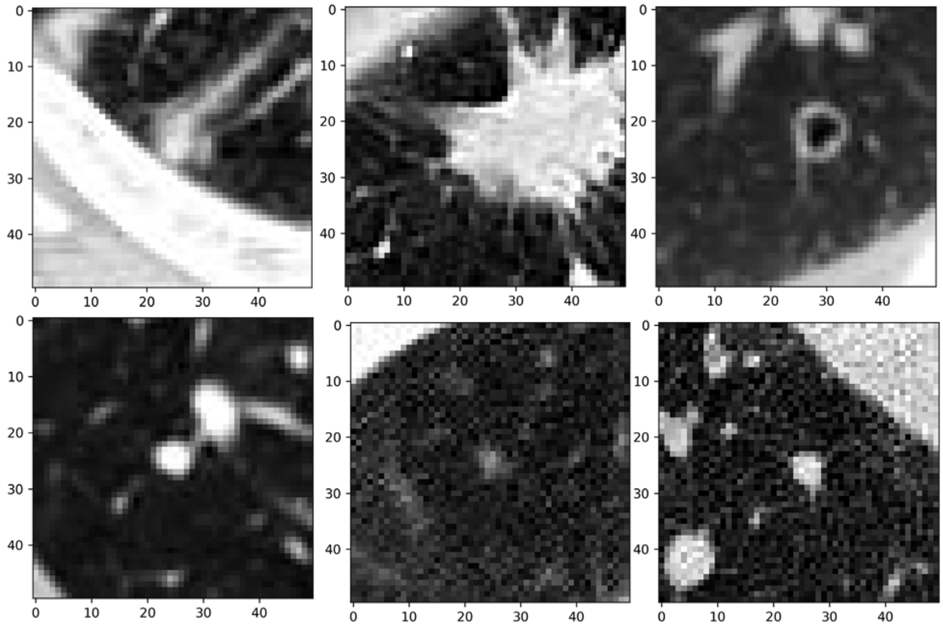


Fig. 8. False Negatives (upper row) and False Positives (lower row) predicted by SqueezNodule-Net V2.

LeNet-5, DenseNet-121, ResNet-50 and VGG-11 is performed (Fig. 10). The models are trained from scratch without using their pre-trained weights, in order to have a fair comparison. The default hypermeters were used as mentioned in Section 4.7. Fig. 10 presents the classification performance metrics of the compared models. Table 3 presents the total trainable parameters, the average training time, the training time per epoch as well as the average number of epochs needed for convergence.

In general, SqueezeNodule-Net V2 performs better than all other investigated models. It is observed (Fig. 10) that SqueezeNodule-Net V2 compared to the aforementioned state-of-art models has 1.1–5.8% better accuracy. Also, its specificity is better by +0.4% to +2.4%, with the exception of ResNet-50 where it is slightly lower by 1.3%. In sensitivity, however, SqueezeNodule-Net V2 is much better, achieving from +8.3% to 13.4%. According to Table 3, the SqueezeNodule-Net V2 is from 8 to 40 times smaller than the known models, and it also is 2.8 to 7.8 times faster. The exception is LeNet-5 where it is a very simple model being 3.7 times smaller and 3 times faster than SqueezeNodule-Net V2, but having poor performance in the evaluation metrics, as is the case with VGG.

In order to compare the mean CV evaluation metrics obtained from the proposed models against the other investigated models we performed pairwise Kruskal–Wallis test by ranks. Table 4 resumes the p-values obtained by comparing SqueezeNodule-Net V1 and V2 against the investigated methods.

Although the size of the samples (i.e. the 5-fold CV performance metrics) is not big, 5 over 7 architectures have at least one performance metric that is statistically significant different from the proposed models, in terms of classification performance metrics. Even though, all investigated architectures do not have statistically significant differences, all investigated models have inferior training and inference

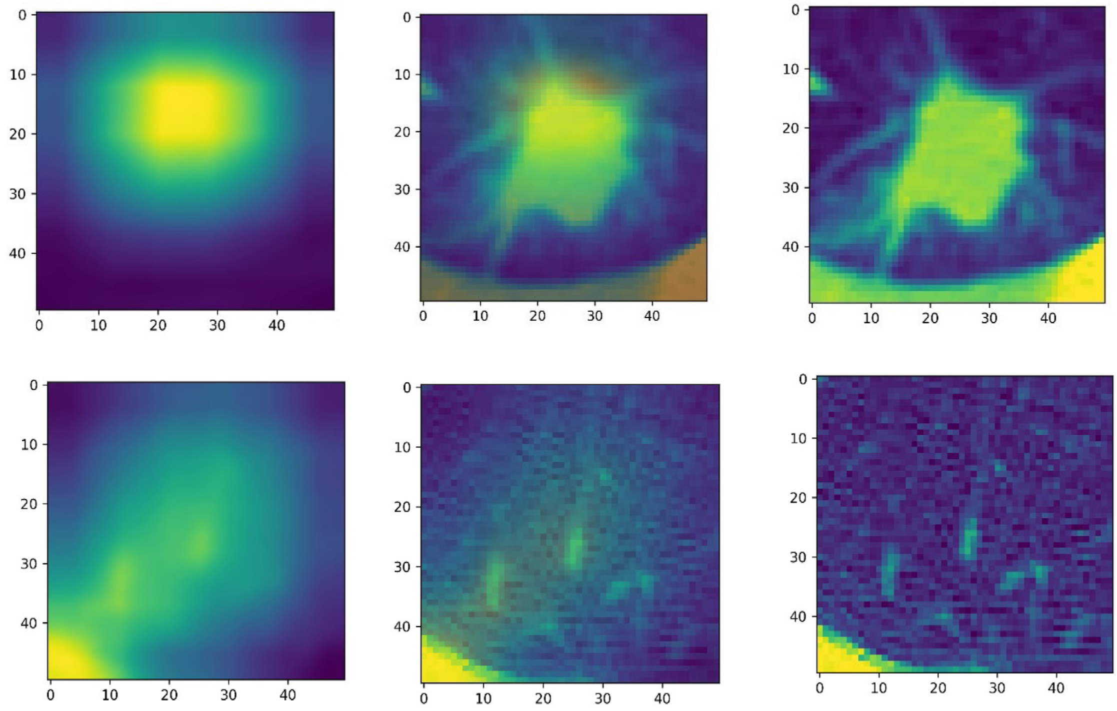


Fig. 9. On the left side are saliency maps of malignant (up) and benign (down). On the right side are the malignant (up) and benign sample (down) as input of the model. In the middle are combinations of saliency map and input, one on top of the other for comparison.

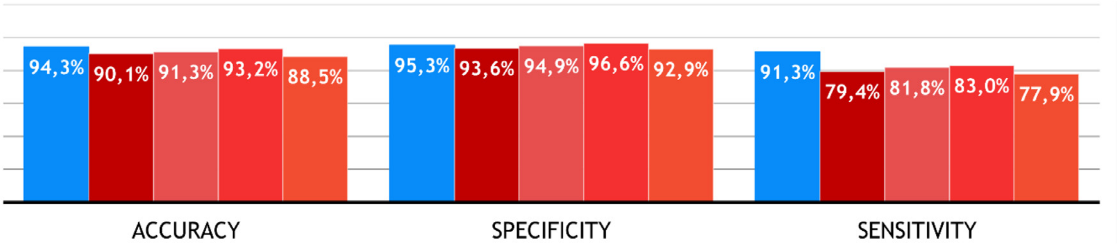


Fig. 10. Histogram comparison among SqueezeNodule-Net V2, LeNet-5, DenseNet-121, ResNet-50 and VGG-11.

Table 3
Comparison with well-known models.

CNN	Total # of Params	Avg. Tr. Time (min)	Tr. Time/epoch (s)	Inf. Time (s)	FLOPS	Avg.# of epochs
SqueezeNodule-Net V2	888.674	1,2	2	0.18	68.6M	38,4
LeNet-5	**234.714**	**0,4**	**0,0065**	**0.05**	**0.8M**	66,6
DenseNet-121	7.033.282	3,3	7	0.90	133.1M	29
ResNet-50	23.535.418	9,3	8	0.81	200.0M	70
VGG-11	28.110.082	0,9	3	0.22	30.8M	**19,8**

times, with the exception of LeNet-5, which classification performance is poor in comparison to the proposed models.

5.1.3. Results for the 3D models

In this section the performance of the same models was evaluated on the 3D dataset. In this set of experiments, the RMSprop optimization algorithm was used, which performs better in this setting (Valova, Harris, Mai, & Gueorguieva, 2020). As the 3D images consumes more memory resources, the batch size needs to be modified for this set of experiments. Specifically, a batch size of 64 was used for SqueezeNodule3D-Net V1 and V2, while SqueezeNet3D needs a batch size of 48 in order to run in our machine. Table 5 shows the average 5-fold CV classification metrics along with the total parameter, training and inference time, flops and mean epochs needed for each model to converge. Fig. 11 presents the confusion matrices of the testing set for SqueezeNodule-Net V1 (left) and V2 (right) for the 3D setting.

In the 3D dataset with images of 32 × 32 × 32 voxels, SqueezeNodule-Net3D V1 has almost the same classification performance compared to SqueezeNet3D, while SqueezeNodule-Net3D V2 has the best classification performance. SqueezeNodule-Net3D V1 compared to the SqueezeNet has about 6.2% fewer parameters. However, SqueezeNodule-Net V1 needs less memory consumption, and as a result, due to the larger batch size, it is about 1.3× faster than SqueezeNet3D. SqueezeNodule-Net3D V2 performs better than SqueezeNet3D, as it has +1.4% more accuracy, +0.1% specificity and +2.7% sensitivity; but

Table 4
Kruskal–Wallis rank tests of SqueezeNodule-Net V1 and SqueezeNodule-Net V2 vs. the investigated methods.

CNN	Accuracy		Specificity		Sensitivity	
	SqueezeNodule-Net V1	SqueezeNodule-Net V2	SqueezeNodule-Net V1	SqueezeNodule-Net V2	SqueezeNodule-Net V1	SqueezeNodule-Net V2
SqueezeNodule-Net V1	–	p=0.3472	–	p=0.3472	–	p=0.2948
SqueezeNodule-Net V2	p=0.3472	–	p=0.3472	–	p=0.2948	–
SqueezeNet	p=0.91681	p=0.5296	p=0.3472	p=0.7540	p=0.1172	p=0.0758
Light CNN	p=0.2506	p=0.9168	**p=0.0472**	p=0.6015	p=0.1172	p=0.0758
LeNet-5	**p=0.0283**	**p=0.0163**	p=0.8340	p=0.1425	**p=0.0163**	**p=0.0163**
DenseNet-121	p=0.1745	p=0.91681	p=0.1172	p=0.4020	p=0.0758	p=0.0758
ResNet-50	p=0.6015	p=0.91681	**p=0.0283**	p=0.3472	**p=0.0472**	p=0.0758
VGG-11	**p=0.0472**	**p=0.0283**	p=0.9168	p=0.3472	**p=0.0090**	**p=0.0090**
ShuffleNet	**p=0.0090**	**p=0.0090**	p=09168	p=0.4647	**p=0.0163**	**p=0.0163**

Table 5
SqueezeNet3D, SqueezeNodule-Net3D V1 and V2 performance.

CNN	Acc.	Spec.	Sens.	G-mean	Total Params	Inf. Time (s)	FLOPS	Tr. Time (min.)	Mean epochs
SqueezeNet3D	94.4% (0.009)	96.1% (0.010)	87.5% (0.012)	91.6% (0.004)	1829K	13.4	5581.0M	11.8	28.4
SqueezeNodule-Net3D V1	94.3% (0.008)	96.0% (0.005)	87.4% (0.024)	91.5% (0.014)	**1715K**	**2.16**	**419.7M**	9.0	25.8
SqueezeNodule-Net3D V2	**95.8%** (0.006)	**96.2%** (0.009)	**90.2%** (0.017)	**93.1%** (0.009)	2511K	2.40	740.9M	**8.8**	**21.2**

Table 6
Comparison with well-known models.

CNN	Total # of Params	Avg. Tr. Time (min)	Tr. Time/epoch (s)	Inf. Time (s)	FLOPS	Avg.# of epochs	Batch
SqueezeNodule3D-Net V2	2,5M	8,8	25	2.40	740.9M	21,2	**64**
LeNet-50 (3D)	**0,2M**	**1,5**	**4**	**0.54**	**28.7M**	22,6	**64**
DenseNet-121 (3D)	11,3M	61,5	90	3.26	671.3M	41,0	8
ResNet-50 (3D)	46,1M	183,6	249	12.3	4136.8M	44,3	8
VGG-11 (3D)	50,7M	48,3	232	13.69	9760.7M	**12,5**	8

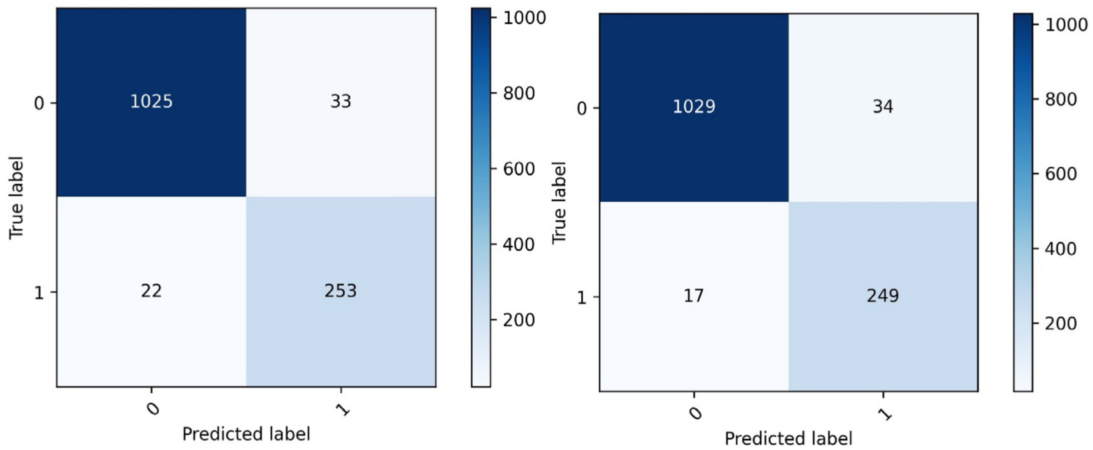


Fig. 11. Confusion matrices of the 3D models of SqueezeNodule-Net V1 (left) and SqueezeNodule-Net V2 (right).

the size of V2 model is 37% larger than the original. Nevertheless, the simpfire module is clearly requiring less computational power and according to the average number of epochs it helps to converge faster. As a result, SqueezeNodule-Net V2 is 1.35× faster than SqueezeNet3D.

5.1.4. *Comparison with other 3D CNNs*

In this section we evaluate our best 3D Model, SqueezeNodule-Net3D V2, against state-of-the-art CNN models: LeNet-5 (3D), ResNet-50 (3D) and VGG-11 (3D) (Fig. 12).

It is observed (Fig. 12) that SqueezeNodule-Net3D V2 compared to the aforementioned models achieves higher accuracy (+0.4% to +4.4%), specificity (+0.1% to 4.8%) as well as sensitivity (+0.4% to +6.6%). Also, according to Table 6 our model is from 4.5 to 20.2 times more compact as well as 4.5 to 21 times faster. It is worth mentioning that the VGG-11 (3D) which achieves better sensitivity by +1.4%, it is much worse in size (20.2× larger) and convergence speed (4.5× slower) while LeNet-5 (3D) which is ~ 7.8× smaller and 5.8× faster loses in metric evaluations.

In order to compare the mean CV evaluation metrics obtained from the proposed models against the other investigated models in the 3D setting, we performed pairwise Kruskal–Wallis test by ranks. Table 7 resumes the p-values obtained by comparing SqueezeNodule-Net V1 and V2 against the investigated methods.

Although the size of the samples (i.e. the 5-fold CV performance metrics) is not big, 2 over 5 architectures have at least one performance metric that is statistically significant different from the proposed

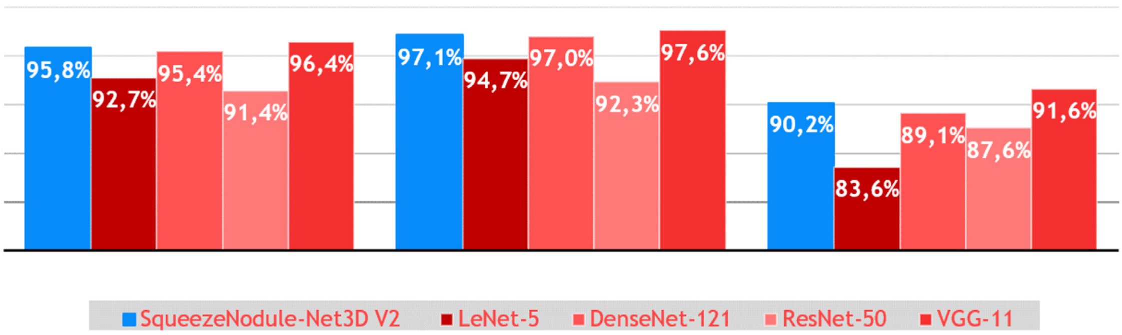


Fig. 12. Histogram comparison among SqueezeNodule-Net3D V2, LeNet-5, ResNet-50 and VGG-11.

Table 7
Kruskal–Wallis rank tests of SqueezeNodule-Net V1 and SqueezeNodule-Net V2 vs. the investigated methods.

CNN	Accuracy		Specificity		Sensitivity	
	SqueezeNodule-Net V1	SqueezeNodule-Net V2	SqueezeNodule-Net V1	SqueezeNodule-Net V2	SqueezeNodule-Net V1	SqueezeNodule-Net V2
SqueezeNodule-Net V1	–	p=0.25059	–	p=0.6015	–	p=0.1745
SqueezeNodule-Net V2	p=0.25059	–	p=0.6015	–	p=0.1745	–
SqueezeNet	p=0.4647	p=0.1172	p=0.34721	p=0.1745	p=1	p=0.0758
LeNet-5	**p=0.0163**	**p=0.0090**	**p=0.0472**	**p=0.0472**	**p=0.009**	**p=0.009**
DenseNet-121	p=0.3472	p=0.91681	p=0.1172	p=0.34721	p=0.6015	p=0.5296
ResNet-50	p=0.2506	p=0.0758	p=0.60151	p=0.6015	p=0.6015	p=0.0758
VGG-11	**p=0.0472**	p=0.1172	**p=0.0283**	p=0.1172	p=0.0758	p=0.17453

models, in terms of classification performance metrics. Even though, all investigated architectures do not have statistically significant differences, all investigated models have in general inferior training and inference times in comparison to our models.

5.2. Comparison of different augmentation methods

In this section we compare the results of the 2D models using three different augmentation methods: (1) Weighted data augmentation on both classes 25% for the benign and 75% for the malignant class; (2) Data augmentation on the malignant class using artificial produced malignant nodules employing DCGAN (Radford et al., 2015) by augmenting the malignant class by 40% of the training and validation set and (3) by enriching the dataset using progressive resizing on both classes starting from 25 × 25 pixels images and progressing to 40 × 40 and 50 × 50 using the procedure in described in (Bhatt et al., 2021).

Table 8 summarizes the results of the different augmentation procedures in comparison to the results obtained by the augmentation procedure used in the experiments in Section 5.1.1 (the percentages inside the parenthesis refer to the difference to the results obtained in Section 5.1.1).

We can observe in the results that the weighted augmentation procedure on the two classes influenced all investigated methods positively, by augmenting slightly their classification performance, with the exception of LeNet-5 network. The same positive effect had also the progressive resizing procedure. In contrast the augmentation by using artificial images produced by the DCGAN had a negative influence on all investigated methods. Furthermore, when using the progressive resizing procedure we achieved, in general, better classification performance in all the investigated metrics: sensitivity increased from 0.7% to 1.6% in all three models, placing thus SqueezeNodule-Net V2 (92.0%) in first place and SqueezeNet (88.3%) in last place. We did not choose to further enlarge the final images in order to make the results comparable to our initial method.

Finally, we have to mention that the proposed methods SqueezeNodule-Net V1 and V2, had better classification performance in comparison to all investigated methods. Furthermore, we can observe that the different augmentation methods do not have a big influence on the ranking of the methods with respect to the classification performance.

5.3. Squeeze ratio comparison

For completeness, the proposed architecture was evaluated for different values of the SR, to observe how much the model's performance varies as a function of SR, as shown in Fig. 13. At the same time, all Fire Modules were tested to have the same Squeeze Ratio as shown in Fig. 14, seeking to prove that the last 2 Fire Modules do not need to have a Squeeze Ratio above 0.25.

In Fig. 13 the Fire Modules of each model have the same Squeeze Ratio, even the last 2 Fire Modules. In Fig. 14 all models have the last 2 Fire Modules with a Squeeze Ratio equal to 0.25. That is, V3(0.60*) means that the first 6 Fire Modules have SR=0.60 and the last 2 Fire Modules have SR=0.25. The same for V4 and V5.

What it is observed is that when the model has the last 2 Fire Modules with $SR > 0.25$ then the model increases in size without achieving better evaluation metrics. In contrast, the models where the last 2 Fire Modules have the last 2 Fire Modules equal to 0.25 (Fig. 14) seem to achieve good evaluation metrics without increasing the model size as much as in Fig. 13.

5.4. Cifar-10 & MNIST

The SqueezeNodule-Net V1 and V2 models were evaluated on the benchmark datasets Cifar-10 and MNIST, and compared with SqueezeNet, LeNet-5, DenseNet-BC, ResNet-50 and VGG-11. In both datasets no pre-trained weights were used to be fair in comparison to our models. The results are shown in Tables 9 and 10.

On Cifar-10 dataset the models ran with a batch size of 64 with the exception of the VGG-11 which was able to run with a batch size of 32, due to a mid-range GPU. All models were run on 100 epochs with the Stochastic Gradient Descent optimizer with learning rate = 0.001 and momentum = 0.9.

On the MNIST dataset the models were run for 15 epochs with a batch size of 128. We used the Adam optimizer with a learning rate = 0.001 as it gave better results than SGD.

Table 10 summarizes the classification and run time performance of the investigated models. The proposed models in both Cifar-10 and MNIST are the most efficient if we consider the training time along with the classification accuracy. Again, SqueezeNodule-Net V2 is the preferred model in terms of both training time and accuracy.

Table 8
Classification results using different augmentation procedures in comparison with the results reported in Tables 2 and 3 (the percentage inside the parenthesis refers to the difference in relation to Tables 2 and 3).

CNN	Symmetric augmentation (25:75) on both classes			Augmentation using DCGANN			Progressive Resizing 25 × 25 → 40 × 40 → 50 × 50		
	Acc.	Spec.	Sens.	Acc.	Spec.	Sens.	Acc.	Spec.	Sens.
SqueezeNodule-Net V1	94.9% (+1.3%)	96.5% (+1.5%)	89.4% (+0.2%)	86.7% (−6.5%)	90.4% (−4.2%)	76.4% (−12.8%)	94.3% (+1.1%)	95.5% (+0.9%)	90.5% (+1.3%)
SqueezeNodule-Net V2	**96.1%** (+1.8%)	**97.7%** (+2.4%)	**91.3%** (+0)	**87.4%** (−6.9%)	**91.8%** (−3.5%)	**77.8%** (−13.5%)	**95.8%** (+1.5%)	**97.7%** (+2.4%)	**92.0%** (+0.7%)
SqueezeNet	93.1% (+1.1%)	96.4% (+2.6%)	85.6% (−1.1%)	87.1% (−4.9%)	91.7% (−2.1%)	74.2% (−12.5%)	95.3% (+1.3%)	97.5% (+3.7%)	88.3% (+1.6%)
Light CNN	93.1% (+1.3%)	96.4% (+1.0%)	85.6% (+2.4%)	[a]	[a]	[a]	[b]	[b]	[b]
LeNet-5	87.4% (−2.5%)	89.3% (−4.3%)	81.2% (+1.8%)	85.5% (−4.6%)	91.5% (−2.1%)	68.6% (−10.8%)	[c]	[c]	[c]
DenseNet-121	93.4% (+2.1%)	96.1% (+1.2%)	84.8% (+3%)	81.3% (−10.0%)	89.9% (−5.0%)	58.1% (−23.7%)	[d]	[d]	[d]
ResNet-50	94.5% (+1.3%)	96.4% (−0.2%)	88.5% (+5.5%)	74.7% (−18.5%)	84.9% (−11.7%)	47.1% (−36.5%)	[d]	[d]	[d]
VGG-11	92.0% (+3.5%)	95.3% (+2.4%)	82.2% (+4.3%)	85.7% (−2.8%)	91.1% (−1.8%)	70.4% (−9.5%)	[d]	[d]	[d]
ShuffleNet	90.1% (+2.7%)	92.7% (+2.1%)	82.3% (+3.9%)	78.3% (−11.8%)	88.3% (−2.3%)	48.9% (−29.5%)	93.2% (+7.3%)	96.4% (+5.8%)	82.1% (+3.7%)

[a] Light CNN cannot handle input images of sizes 32 × 32 as DCGANN demands.
[b] Light CNN cannot handle input images of sizes 40 × 40 as the Progressive Resizing procedure demands.
[c] LeNet-5 needs minimum size of input images 28 × 28, thus the Progressive Resizing Procedure cannot start from 25 × 25 as in our setting.
[d] The models need minimum size of input images 32 × 32, thus the Progressive Resizing Procedure cannot start from 25 × 25 as in our setting.

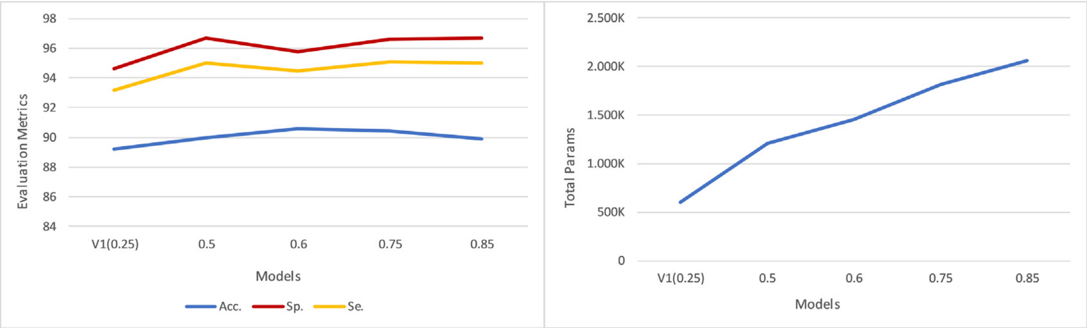


Fig. 13. Comparison of the evaluation metrics and total number of parameters between models with different values of Squeeze Ratio, following the SqueezeNodule-Net architecture. All Fire Modules of each model have the same SR.

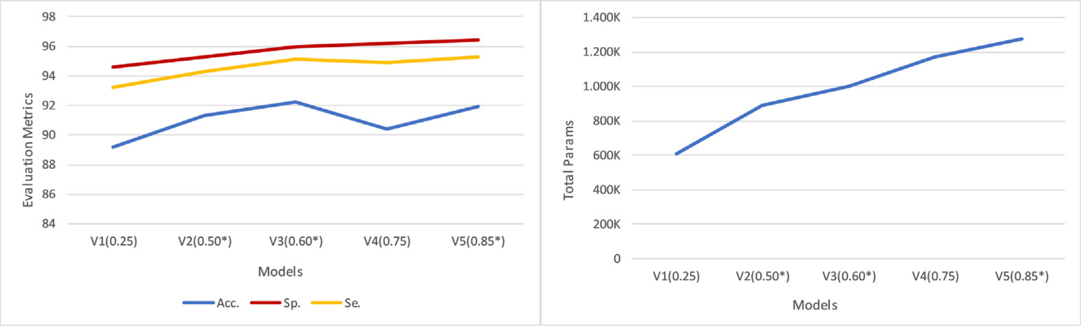


Fig. 14. Comparison of the evaluation metrics and total number of parameters among models with different values of Squeeze Ratio, following the SqueezeNodule-Net architecture. The last 2 Fire Modules in each model have SR=0.25.

6. Discussion

In this work we proposed a modified SqueezeNet model, called SqueezeNodule-Net, to tackle the task of lung nodules classification in two classes: malignant and benign. We modify the Fire Module of SqueezeNet, which performs in the first place a 1 × 1 convolution, for projecting the input into a lower dimensionality filters' space, compressing (i.e. squeezing) the input of the module and expanding afterwards the squeezed tensor by applying separately an 1 × 1 and a 3 × 3 expanding convolutions to result to an output having twice the size of the expand filters e_i after the concatenation layer, by omitting the 1 × 1 expand convolution. In this way, the output of

Table 9
Comparison with Cifar-10 dataset.

CNN	Accuracy	Total Tr. Time (min)
SqueezeNodule-Net V1	81.5%	38.3
SqueezeNodule-Net V2	**82.5%**	40.0
SqueezeNet	80.6%	41.6
LeNet-5	65.1%	**36.6**
DenseNet-BC	79.9%	46.6
ResNet-50	69.9%	105.0
VGG-11	68.7%	105.0

Table 10
Comparison with MNIST dataset.

CNN	Accuracy	Total Tr. Time (min)
SqueezeNodule-Net V1	99.0%	1.75
SqueezeNodule-Net V2	**99.1%**	2.00
SqueezeNet	98.9%	2.75
LeNet-5	98.7%	**0.50**
DenseNet-BC	98.6%	3.0
ResNet-50	99.1%	11.50
VGG-11	98.2%	8.75

the modified Fire Module (simpfire) will have exactly e_i filter, without loss of information, since the spatial as well as the filters' compressed (i.e. embedded) information expands through the 3 × 3 convolution. The results of our experiments regarding classification performance in the task of lung nodules classification showed that SqueezeNodule-Net V1 and V2 perform better or comparable to other compact architectures, as well as in comparison to state-of-the-art Deep Learning models. On the other hand, our proposed architecture has in general fewer parameters, and needs thus less training and inference time, without compromising the classification performance. For example, although LeNet-5 has fewer parameters, and requires less training and inference time, its performance on Sensitivity is poor in comparison to the proposed models.

Data augmentation can insert bias to the classification model, and this fact can sometimes lead to confounded training. One of the augmentation methods that we used was augmenting the positive (malignant) samples by performing a 90 degrees rotation of the existing positive samples. In order to ensure that this augmentation procedure does not insert bias in the training of our models we used three different techniques: (1) symmetric augmentation on both classes followed by weighted training, (2) augmentation by producing artificial positive samples by applying a generative model and (3) the progressive resizing technique (although this technique is rather a transfer learning paradigm, as we discuss it in the paragraph regarding of the data augmentation). As reported in Section 5.2, augmenting symmetrical on both classes, and then using a weighted training as well as the progressive resizing procedures, had a small positive impact on all classification evaluation metrics on all investigated methods, without impacting the ranking of the investigated methods. That could suggest that even the 90 degrees augmentation of the positive class had not inserted bias into the training procedure. On the other hand, the DCGANN augmentation technique had a negative influence on all investigating methods. This could be attributed to the fact that DCGAN produces 32 × 32 images, and as the images we used on our experiments are of size 50 × 50, we had to resize the images of the initial dataset and train the models on 32 × 32 images. In medical applications, like the lung nodules classification, down-scaling the input images can result to possible loss of crucial information and thus in poor classification results, as it may be the case, with the DCGAN augmentation technique used here. Further investigation of augmentation by artificial generated images is needed, something that is out of the scope of this work.

Hyperparameters play a significant role on all Machine Learning and Deep Learning models and the tuning of them is a long and elaborating procedure. In order to ensure that the results provided in Section 5 are not biased towards our models, because of the hyperparameters, we revaluated the classification metrics of the investigated base line methods, by optimizing their hyperparameters for the lung nodules classification task. For this purpose, we used a grid search on the hyperparameter's space consisting of (1) learning rate, (2) optimization algorithm and (3) batch size. We tuned the hyperparameters in a validation set and in each case, we used the best model to compute the test classification metrics. Only in two cases (LeNet-5 and ShuffleNet) the best hyperparameters were not the hyperparameters of the reported classification metrics in Section 5. Table 11 summarizes the classification metrics for these two models.

In the case of the LeNet-5 network, the hyperparameter optimization has augmented slightly (below 1%) all the classification metrics. In the case of ShuffleNet the classification metrics have augmented significantly (from 5.7%–9.3%). Even through the specificity of ShuffleNet is slightly better (1%) than SqueezeNodule-Net V2 when the other two metrics (accuracy and sensitivity) are considered, the proposed model has better classification performance (0.2% and 3.9% respectively). We need to note here, that this comparison is done against the proposed models without optimizing their corresponding hyperparameters. Also, in the results reported for the CIFAR-10 and MNIST datasets hyperparameters were not tuned. For the proposed models we utilized the parameters that were utilized in the LUNA-16 dataset, (a dataset completely different of CIFAR-10 and MNIST databases), while for the base models the default parameters were used.

7. Conclusion

This work focused on lung cancer nodule classification, relying on a fairly compact model, SqueezeNet. The main goal was to reduce the required computational power and runtime needed of classification systems while maintaining high accuracy. In this work two models were proposed, SqueezeNodule-Net V1 and SqueezeNodule-Net V2. Special emphasis was given on the latter, as it is the best performing model. These models are a simplified version of the original SqueezeNet and are capable of running on mid-range computer systems, at reasonably good times.

In the first part of the experiments, these models were trained for 2D images (50 × 50), giving much better results than classical models such as LeNet-5, DenseNet-121, ResNet-50, VGG-11, but also better than SqueezeNet itself. SqueezeNodule-Net V1 has 93.2% accuracy, 94.6% specificity and 89.2% sensitivity, while SqueezeNodule-Net V2 has 94.3% accuracy, 95.3% specificity and 91.3% sensitivity. Still, our 2D models V1 and V2 are about 1.5× and 2.1× faster respectively than SqueezeNet.

In the second part, these models were used for 3D images of size 32 × 32 × 32, as well, in order to observe their performance in 3D space. Thus, the SqueezeNet and the proposed models were converted to receive the 3D images as well as the LeNet-5 (3D), ResNet-50 (3D) and VGG-11 (3D) models, in order to perform the comparison. The results were quite satisfying, as both proposed models performed better than the SqueezeNet even in 3D space. In particular, SqueezeNodule-Net V1 (3D) has 94.3% accuracy, 96.0% specificity and 87.4% sensitivity, while SqueezeNodule-Net V2 (3D) has 95.8% accuracy, 96.2% specificity and 90.2% sensitivity. SqueezeNet (3D) required more computational power, resulting in both proposed models being about 1.3× faster. The only downside is that even though they require less computational power in 3D space the size of our models increases more than the SqueezeNet. However, they are still quite competitive considering time and accuracy.

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.

Table 11
The classification evaluation using hyperparameter tuning of the baseline investigated methods.

CNN	Acc.	Sp.	Se.	Optmizer	lr	Batch size
LeNet-5	89.1% (0.050)	92.0% (0.047)	80.5% (0.072)	Adam	0.01	128
ShuffleNet	94.1% (0.005)	96.3% (0.005)	87.7% (0.009)	SGD	0.001	64

References

Alakwaa, W., Nassef, M., & Badr, A. (2017). Lung cancer detection and classification with 3D convolutional neural network (3D-CNN). *International Journal of Advanced Computer Science and Applications*, *8*(8), http://dx.doi.org/10.14569/IJACSA.2017.080853.

Bhatt, A. R., Ganatra, A., & Kotecha, K. (2021). Cervical cancer detection in pap smear whole slide images using convnet with transfer learning and progressive resizing. *PeerJournal Computer Science*, *7*(e348), http://dx.doi.org/10.7717/peerj-cs.348.

Cao, W., Wu, R., Cao, G., & He, Z. (2020). A comprehensive review of computer-aided diagnosis of pulmonary nodules based on computed tomography scans. *IEEE Access*, *8*, 154007–154023. http://dx.doi.org/10.1109/ACCESS.2020.3018666.

Cao, F., & Zhao, H. (2021). Automatic lung segmentation algorithm on chest X-ray images based on fusion variational auto-encoder and three-terminal attention mechanism. *Symmetry*, *13*(5), 814. http://dx.doi.org/10.3390/sym13050814.

Chaudhari, P., Agrawal, H., & Kotecha, K. (2020). Data augmentation using MG-GAN for improved cancer classification on gene expression data. *Soft Computing*, *24*(15), 11381–11391. http://dx.doi.org/10.1007/s00500-019-04602-2.

Chen, C.-F., Lee, G. G., Sritapan, V., & Lin, C.-Y. (2016). Deep convolutional neural network on iOS mobile devices. In *2016 IEEE International workshop on signal processing systems* (pp. 130–135). http://dx.doi.org/10.1109/SiPS.2016.31.

Cheng, Y., Wang, D., Zhou, P., & Zhang, T. (2017). A survey of model compression and acceleration for deep neural networks. http://dx.doi.org/10.48550/ARXIV.1710.09282.

Choi, W., & Choi, T.-S. (2013). Automated pulmonary nodule detection system in computed tomography images: A hierarchical block classification approach. *Entropy*, *15*, 507–523. http://dx.doi.org/10.3390/e15020507.

Choi, Y., El-Khamy, M., & Lee, J. (2016). http://dx.doi.org/10.48550/ARXIV.1612.01543, Towards the Limit of Network Quantization.

Chollet, F., et al. (2015). Keras.

Cui, S., Ming, S., Lin, Y., Chen, F., Shen, Q., Li, H., et al. (2020). Development and clinical application of deep learning model for lung nodules screening on CT images. *Scientific Reports*, *10*(1), 13657. http://dx.doi.org/10.1038/s41598-020-70629-3.

Deng, J., Dong, W., Socher, R., Li, L., Li, Kai., & Fei-Fei, Li. (2009). ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on computer vision and pattern recognition* (pp. 248–255). http://dx.doi.org/10.1109/CVPR.2009.5206848.

Diniz, J. O. B., Quintanilha, D. B. P., Santos Neto, A. C., da Silva, G. L. F., Ferreira, J. L., Netto, S. M. B., et al. (2021). Segmentation and quantification of COVID-19 infections in CT using pulmonary vessels extraction and deep learning. *Multimedia Tools and Applications*, *80*(19), 29367–29399. http://dx.doi.org/10.1007/s11042-021-11153-y.

Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M. W., & Keutzer, K. (2021). A survey of quantization methods for efficient neural network inference. http://dx.doi.org/10.48550/ARXIV.2103.13630.

Gite, S., Mishra, A., & Kotecha, K. (2022). Enhanced lung image segmentation using deep learning. *Neural Computing and Applications*, http://dx.doi.org/10.1007/s00521-021-06719-8.

Gong, Y., Liu, L., Yang, M., & Bourdev, L. (2014). Compressing deep convolutional networks using vector quantization. http://dx.doi.org/10.48550/ARXIV.1412.6115.

Gordienko, Yu, Gang, P., Hui, J., Zeng, W., Yu, Kochura., Alienin, O., et al. (2019). Deep learning with lung segmentation and bone shadow exclusion techniques for chest X-ray analysis of lung cancer. In Z. Hu, S. Petoukhov, I. Dychka, M. He (Eds.), *Advances in computer science for engineering and education (Vol. 754, 638–647)*. Springer International Publishing, http://dx.doi.org/10.1007/978-3-319-91008-6_63.

Hamidian, S., Sahiner, B., Petrick, N., & Pezeshk, A. (2017). In S. G. Armato, & N. A. Petrick (Eds.), *3D Convolutional neural network for automatic detection of lung nodules in chest CT*. http://dx.doi.org/10.1117/12.2255795, 1013409.

Han, S., Pool, J., Tran, J., & Dally, W. (2015). Learning both weights and connections for efficient neural network. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in neural information processing systems (Vol. 28)*. Curran Associates, Inc., https://proceedings.neurips.cc/paper/2015/file/ae0eb3eed39d2bcef4622b2499a05fe6-Paper.pdf.

Hassibi, B., & Stork, D. (1992). Second order derivatives for network pruning: Optimal brain surgeon. In S. Hanson, J. Cowan, & C. Giles (Eds.), *Advances in neural information processing systems (Vol. 5)*. Morgan-Kaufmann, https://proceedings.neurips.cc/paper/1992/file/303ed4c69846ab36c2904d3ba8573050-Paper.pdf.

Hosseini-Asl, E., Zurada, J. M., Gimelfarb, G., & El-Baz, A. (2016). 3-D lung segmentation by incremental constrained nonnegative matrix factorization. *IEEE Transactions on Biomedical Engineering*, *63*(5), 952–963. http://dx.doi.org/10.1109/TBME.2015.2482387.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. http://dx.doi.org/10.48550/ARXIV.1704.04861.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and.

Kamble, B., Sahu, S. P., & Doriya, R. (2020). A review on lung and nodule segmentation techniques. In M. L. Kolhe, S. Tiwari, M. C. Trivedi, & K. K. Mishra (Eds.), *Advances in data and information sciences (vol. 94, 555–565)*. Springer Singapore, http://dx.doi.org/10.1007/978-981-15-0694-9_52.

Khumancha, M. B., Barai, A., & Rao, C. B. R. (2019). Lung cancer detection from computed tomography (CT) scans using convolutional neural network. In *2019 10th International conference on computing, communication and networking technologies* (pp. 1–7). http://dx.doi.org/10.1109/ICCCNT45670.2019.8944824.

Kingma, D. P., & Ba, J. (2017). Adam: A method for stochastic optimization.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems (Vol. 25)*. Curran Associates, Inc., https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Lee, S. L. A., Kouzani, A. Z., & Hu, E. J. (2012). Automated detection of lung nodules in computed tomography images: A review. *Machine Vision and Applications*, *23*(1), 151–163. http://dx.doi.org/10.1007/s00138-010-0271-2.

Li, W., Cao, P., Zhao, D., & Wang, J. (2016). Pulmonary nodule classification with deep convolutional neural networks on computed tomography images. *Computational and Mathematical Methods in Medicine*, *2016*, 1–7. http://dx.doi.org/10.1155/2016/6215085.

Li, R., Xiao, C., Huang, Y., Hassan, H., & Huang, B. (2022). Deep learning applications in computed tomography images for pulmonary nodule detection and diagnosis: A review. *Diagnostics*, *12*(2), 298. http://dx.doi.org/10.3390/diagnostics12020298.

Liao, F., Liang, M., Li, Z., Hu, X., & Song, S. (2019). Evaluate the malignancy of pulmonary nodules using the 3-D deep leaky noisy-OR network. *IEEE Transactions on Neural Networks and Learning Systems*, *30*(11), 3484–3495. http://dx.doi.org/10.1109/TNNLS.2019.2892409.

Ma, N., Zhang, X., Zheng, H.-T., & Sun, J. (2018). ShuffleNet V2: Practical guidelines for efficient CNN architecture design. In V. Ferrari, M. Hebert, C. Sminchisescu, & Y. Weiss (Eds.), *Computer vision – ECCV 2018 (Vol. 11218, 122–138)*. Springer International Publishing, http://dx.doi.org/10.1007/978-3-030-01264-9_8.

Mastouri, R., Khlifa, N., Neji, H., & Hantous-Zannad, S. (2021). A bilinear convolutional neural network for lung nodules classification on CT images. *International Journal of Computer Assisted Radiology and Surgery*, *16*(1), 91–101. http://dx.doi.org/10.1007/s11548-020-02283-z.

Mathews, R. P. (2019). CAD in medical imaging: A review of current trends and future directions. *Ijert*, *8*(06), 199–213.

Mishchenko, Y., Goren, Y., Sun, M., Beauchene, C., Matsoukas, S., Rybakov, O., et al. (2019). Low-bit quantization and quantization-aware training for small-footprint keyword spotting. In *2019 18th IEEE International conference on machine learning and applications* (pp. 706–711). http://dx.doi.org/10.1109/ICMLA.2019.00127.

Nagel, M., Baalen, M. V., Blankevoort, T., & Welling, M. (2019). Data-free quantization through weight equalization and bias correction. In *2019 IEEE/CVF International conference on computer vision* (pp. 1325–1334). http://dx.doi.org/10.1109/ICCV.2019.00141.

Nalepa, J., Antoniak, M., Myller, M., Lorenzo, P. Ribalta., & Marcinkiewicz, M. (2020). Towards resource-frugal deep convolutional neural networks for hyperspectral image segmentation. *Microprocessors and Microsystems*, *73*, Article 102994. http://dx.doi.org/10.1016/j.micpro.2020.102994.

Nery, F., Silva, J. S., Ferreira, N. C., & Caramelo, F. (2012). 3D automatic lung segmentation in low-dose CT. In *2012 IEEE 2nd Portuguese meeting in bioengineering* (pp. 1–4). http://dx.doi.org/10.1109/ENBENG.2012.6331360.

Nirthika, R., Manivannan, S., Ramanan, A., & Wang, R. (2022). Pooling in convolutional neural networks for medical image analysis: A survey and an empirical study. *Neural Computing and Applications*, *34*(7), 5321–5347. http://dx.doi.org/10.1007/s00521-022-06953-8.

Medeiros da Nobrega, R. V., Rodrigues, M. B., & Filho, P. P. R. (2017). Segmentation and visualization of the lungs in three dimensions using 3D region growing and visualization toolkit in CT examinations of the chest. In *2017 IEEE 30th International symposium on computer-based medical systems* (pp. 397–402). http://dx.doi.org/10.1109/CBMS.2017.23.

Özyurt, F., Sert, E., & Avcı, D. (2020). An expert system for brain tumor detection: Fuzzy C-means with super resolution and convolutional neural network with extreme learning machine. *Medical Hypotheses*, *134*, Article 109433. http://dx.doi.org/10.1016/j.mehy.2019.109433.

Polsinelli, M., Cinque, L., & Placidi, G. (2020). A light CNN for detecting COVID-19 from CT scans of the chest. *Pattern Recognition Letters*, *140*, 95–100. http://dx.doi.org/10.1016/j.patrec.2020.10.001.

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. http://dx.doi.org/10.48550/ARXIV.1511.06434.

Rahman, T., Khandakar, A., Kadir, M. A., Islam, K. R., Islam, K. F., Mazhar, R., et al. (2020). Reliable tuberculosis detection using chest X-ray with deep learning segmentation and visualization. *IEEE Access*, *8*, 191586–191601. http://dx.doi.org/10.1109/ACCESS.2020.3031384.

Rastegari, M., Ordonez, V., Redmon, J., & Farhadi, A. (2016). XNOR-Net: ImageNet classification using binary convolutional neural networks. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer Vision – ECCV 2016 (Vol. 9908, 525–542)*. Springer International Publishing.

Rebouças Filho, P. P., Cortez, P. C., da Silva Barros, A. C., Albuquerque, V. H. C., & Tavares, J. M. R. S. (2017). Novel and powerful 3D adaptive crisp active contour method applied in the segmentation of CT lung images. *Medical Image Analysis*, *35*, 503–516. http://dx.doi.org/10.1016/j.media.2016.09.002.

Redmon, J., Divvala, S. K., Girshick, R. B., & Farhadi, A. (2015). You only look once: Unified, real-time object detection. CoRR, abs/1506.0.

Mobeen-ur Rehman Khan, S. H., Abbas, Z., & Danish Rizvi, S. M. (2019). Classification of diabetic retinopathy images based on customised CNN architecture. In *2019 Amity international conference on artificial intelligence* (pp. 244–248). http://dx.doi.org/10.1109/AICAI.2019.8701231.

Santos, A. G., de Souza, C. O., Zanchettin, C., Macêdo, D., Oliveira, A. L. I., & Ludermir, T. (2018). Reducing SqueezeNet storage size with depthwise separable convolutions. In *2018 International joint conference on neural networks* (pp. 1–6). http://dx.doi.org/10.1109/IJCNN.2018.8489442.

Setio, A. A. A., Traverso, A., de Bel, T., Berens, M. S. N., van den Bogaard, C., Cerello, P., et al. (2017). Validation comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the LUNA16 challenge. *Medical Image Analysis*, *42*, 1–13. http://dx.doi.org/10.1016/j.media.2017.06.015.

Shaukat, F., Raja, G., & Frangi, A. F. (2019). Computer-aided detection of lung nodules: A review. *Journal of Medical Imaging*, *6*(02), 1. http://dx.doi.org/10.1117/1.JMI.6.2.020901.

Siegel, R. L., Miller, K. D., Fuchs, H. E., & Jemal, A. (2021). Cancer statistics 2021. CA: A cancer. *Journal for Clinicians*, *71*(1), 7–33. http://dx.doi.org/10.3322/caac.21654.

Su, Y., Li, D., & Chen, X. (2021). Lung nodule detection based on faster R-CNN framework. *Computer Methods and Programs in Biomedicine*, *200*, Article 105866. http://dx.doi.org/10.1016/j.cmpb.2020.105866.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2014). Going deeper with convolutions. http://dx.doi.org/10.48550/ARXIV.1409.4842.

Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., et al. (2019). MnasNet: Platform-aware neural architecture search for mobile. In *2019 IEEE/CVF Conference on computer vision and pattern recognition* (pp. 2815–2823). http://dx.doi.org/10.1109/CVPR.2019.00293.

Ucar, F., & Korkmaz, D. (2020). COVIDiagnosis-net: Deep Bayes-SqueezeNet based diagnosis of the coronavirus disease 2019 (COVID-19) from X-ray images. *Medical Hypotheses*, *140*, Article 109761. http://dx.doi.org/10.1016/j.mehy.2020.109761.

Utkin, L., Meldo, A., Kovalev, M., & Kasimov, E. (2019). An ensemble of triplet neural networks for differential diagnostics of lung cancer. In *2019 25th Conference of open innovations association* (pp. 346–352). http://dx.doi.org/10.23919/FRUCT48121.2019.8981542.

Valova, I., Harris, C., Mai, T., & Gueorguieva, N. (2020). Optimization of convolutional neural networks for imbalanced set classification. *Procedia Computer Science*, *176*, 660–669. http://dx.doi.org/10.1016/j.procs.2020.09.038.

Vanhoucke, V., Senior, A., & Mao, M. Z. (2011). Improving the speed of neural networks on CPUs. In *Deep learning and unsupervised feature learning workshop, NIPS 2011*.

Wang, W., & Chakraborty, G. (2019). Evaluation of malignancy of lung nodules from CT image using recurrent neural network. In *2019 IEEE International conference on systems, man and cybernetics* (pp. 2992–2997). http://dx.doi.org/10.1109/SMC.2019.8913885.

Wen, W., Wu, C., Wang, Y., Chen, Y., & Li, H. (2016). Learning structured sparsity in deep neural networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in neural information processing systems (vol. 29)*. Curran Associates, Inc., https://proceedings.neurips.cc/paper/2016/file/41bfd20a38bb1b0bec75acf0845530a7-Paper.pdf.

Wu, B., Keutzer, K., Dai, X., Zhang, P., Wang, Y., Sun, F., et al. (2019). FBNet: Hardware-aware efficient ConvNet design via differentiable neural architecture search. In *2019 IEEE/CVF Conference on computer vision and pattern recognition* (pp. 10726–10734). http://dx.doi.org/10.1109/CVPR.2019.01099.

Wu, J., Leng, C., Wang, Y., Hu, Q., & Cheng, J. (2016). Quantized convolutional neural networks for mobile devices. In *2016 IEEE Conference on computer vision and pattern recognition* (pp. 4820–4828). http://dx.doi.org/10.1109/CVPR.2016.521.

Xie, Y., Xia, Y., Zhang, J., Song, Y., Feng, D., Fulham, M., et al. (2019). Knowledge-based collaborative deep learning for benign-malignant lung nodule classification on chest CT. *IEEE Transactions on Medical Imaging*, *38*(4), 991–1004. http://dx.doi.org/10.1109/TMI.2018.2876510.

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: An overview and application in radiology. *Insights Into Imaging*, *9*(4), 611–629. http://dx.doi.org/10.1007/s13244-018-0639-9.

Yang, H., Yu, H., & Wang, G. (2016). Deep learning for the classification of lung nodules.

Yin, P., Zhang, S., Lyu, J., Osher, S., Qi, Y., & Xin, J. (2019). Blended coarse gradient descent for full quantization of deep neural networks. *Research in the Mathematical Sciences*, *6*(1), 14. http://dx.doi.org/10.1007/s40687-018-0177-6.

Yuan, H., Fan, Z., Wu, Y., & Cheng, J. (2021). An efficient multi-path 3D convolutional neural network for false-positive reduction of pulmonary nodule detection. *International Journal of Computer Assisted Radiology and Surgery*, *16*(12), 2269–2277. http://dx.doi.org/10.1007/s11548-021-02478-y.

Zhang, G., Lin, L., & Wang, J. (2021). Lung nodule classification in CT images using 3D DenseNet. *Journal of Physics: Conference Series*, *1827*(1), Article 012155. http://dx.doi.org/10.1088/1742-6596/1827/1/012155.

Zhang, G., Yang, Z., Gong, L., Jiang, S., Wang, L., & Zhang, H. (2020). Classification of lung nodules based on CT images using squeeze-and-excitation network and aggregated residual transformations. *La Radiologia Medica*, *125*(4), 374–383. http://dx.doi.org/10.1007/s11547-019-01130-9.

Zhang, Y., Zhang, J., Zhao, L., Wei, X., & Zhang, Q. (2018). Classification of benign and malignant pulmonary nodules based on deep learning. In *2018 5th International conference on information science and control engineering* (pp. 156–160). http://dx.doi.org/10.1109/ICISCE.2018.00042.

Zhao, Q., Sugiyama, M., Yuan, L., & Cichocki, A. (2019). Learning efficient tensor representations with ring-structured networks. In *ICASSP 2019-2019 IEEE International conference on acoustics, speech and signal processing* (pp. 8608–8612). http://dx.doi.org/10.1109/ICASSP.2019.8682231.

Zhao, C., Xu, Y., He, Z., Tang, J., Zhang, Y., Han, J., et al. (2021). Lung segmentation and automatic detection of COVID-19 using radiomic features from chest CT images. *Pattern Recognition*, *119*, Article 108071. http://dx.doi.org/10.1016/j.patcog.2021.108071.