A deep CNN of 15 layers, proposed by G. S. Tran [13], was trained with augmented 64x64 patches from LUNA16 database. Kernels of size 5x5 for the first and 3x3 in subsequent layers were used, as recommended in this study, to obtain an accuracy of 97.2% using focal loss function. Variations in pooling operations were introduced by W. Shen [14] applying a multi-crop strategy to feature maps. Trained with augmented LIDC-IDRI nodules, small filters and three convolutional layers, their MC-CNN achieved an accuracy of 87.1%.

Multi-Level CNNs created by J. Lyu [15] used a combination of networks with different kernels to fed a fully-connected layer and a softmax classifier, reaching an accuracy of 84.3% on LIDC-IDRI nodules. LUNA16 database was used for developing an ensemble of three variable depth 3D-CNN by W. Huang [16]. Using input sizes of (32, 64, 96) and 5x5 filters, reported accuracies reached 81.7% and 85.1% with 0.125 and 0.25 FPs/scan.

In this paper, a comprehensive study of the influence of adjustable hyperparameters on CNN performance is conducted. Several variations of a three convolutional layer model are trained, selecting different learning rates, number of kernels per layer, kernel sizes and pooling operations. The analysis is focused on training and testing processes, evaluating the evolution of classification accuracy, confusion matrix parameters and loss function values, among other parameters. To conduct this, nodule and non-nodule datasets are generated from scratch using images and information provided with an annotated database, and applying data pre-processing and augmentation techniques.

2. Material and methods

2.1. Image Database and Patch Cropping

A PC with an Intel-i7 processor, 12GB of DDR3-RAM and an NVIDIA GeForce 920M Graphical Card is used for CNN training, validation and testing. Code is compiled with *Theano* [17], a *Python* implementation for machine learning that works with symbolic calculations as graphs and runs in GPU [18]. Patch cropping and preprocessing are done using Matlab® platform.

The Cancer Imaging Archive (TCIA) is a public database for radiological cases [19]. In this work, annotated screening and diagnostic CT scans from the Lung Image Database Consortium (LIDC-IDRI) are used [20], [21]. *LIDC Toolbox*, developed by T. Lampert [22], is used to extract nodule locations that allow patch cropping (see Figure 1).

Figure 1. Two cropped nodules for different matrix sizes

As slice thickness is usually smaller than nodule size, visualized sections increase available patches. Besides, more nodules fulfil cropping requirements for bigger sizes (Table 1).

Table 1. Total number of nodule patches for each size

Matrix size	Nodule patches
16x16	5008
24x24	6373
32x32	7480
40x40	8045
48x48	8164

In Figure 2, the wide variability in nodule shape, morphology, location and surrounding structures can be seen.

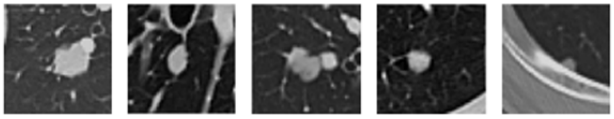

Figure 2. Example of five nodules cropped in 48x48

This holds for non-nodule patches, randomly selected from non-marked and healthy regions, as lung anatomy includes airways, blood vessels, lymph nodes and alveolar airspace, as shown in Figure 3.

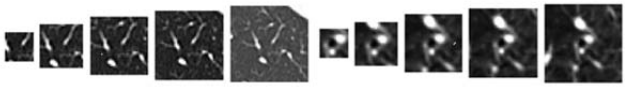

Figure 3. Examples of two non-nodule regions

2.2. Data Preprocessing and Augmentation

Pixel values are integers of 16 bits with sign. As neural activation functions perform better in the interval (-1, 1), cropped matrices are normalized by their maximum absolute value, using Eq.1 and Eq.2, and then stored with eight decimal digits.

$$x_{max} = max\{|max(x_{ij})|, |min(x_{ij})|\} \; \forall \; i,j \quad (1)$$

$$x_{ij}^{norm} = \frac{x_{ij}}{x_{max}} \quad (2)$$

To expand the above cited number of patches, data augmentation based on rigid geometrical transformations is applied to each cropped patch, as shown in Figure 4 for a 40x40 nodule. For nodule patches, four size-dependent random vector translations, four random rotations and four rotation+translation are applied. For non-lesions, due to their greater availability, only five rotations are

applied. To allow supervised learning, a numerical label is assigned to each patch, according to radiologists' annotations: "1" for nodules, and "0" for non-nodules.

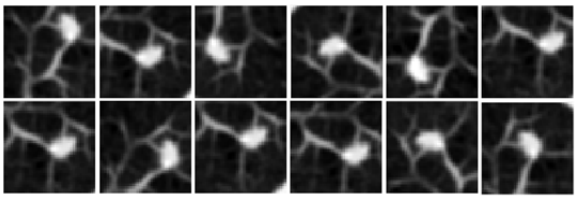

Figure 4. Patches generated from an original nodule

2.3. Neural Network Architecture and Training

All nodules and non-nodules for each matrix size, along with their numerical labels, are divided into three datasets: 75% for training and 12.5% for validation and testing. Each dataset contains the same number of patches of each type. Their distribution and patch grouping (batches), to make training more efficient, can be seen in Table 2.

Table 2. Distribution of patches (nodules + non-nodules) for each matrix size

Matrix size	Training patches (batches)	Validation patches (batches)	Testing patches (batches)	Total patches (batches)
16x16	97656 (156)	16276 (26)	16276 (26)	130208 (208)
24x24	123948 (198)	20658 (33)	20658 (33)	165264 (264)
32x32	145232 (232)	24414 (39)	24414 (39)	194060 (310)
40x40	156500 (250)	25666 (41)	25666 (41)	207832 (332)
48x48	159004 (254)	26292 (42)	26292 (42)	211588 (338)

A tutorial on Deep Learning algorithm implementation with *Theano* was developed by LISA lab [23]. The network configuration selected for this research is based on *LeNet* model, initially applied to character recognition in documents [24]. A third convolutional layer has been added to the original model to increase depth, as shown in Figure 5. They are followed by a fully-connected layer and a logistic regression classifier.

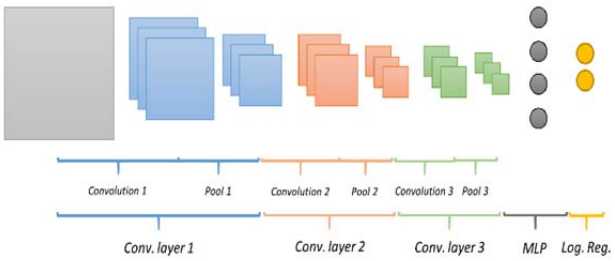

Figure 5. Shape of CNN architecture used in this work

To update model parameters, stochastic gradient descent (SGD) is modified to use groups of samples (batches) for estimating gradient values. Weights and offsets are randomly initialized and updated several times on every epoch. Loss function optimized during learning is negative log-likelihood, with no regularization. Neural activation function is the hyperbolic tangent and pooling is average of size 2. For early stopping, training iterations initially set (patience parameter) are only extended if validation improves beyond an improvement threshold.

Tunable hyperparameters that impact CNN behavior include learning rate, number of kernels per layer, their sizes and pooling operations. To test their influence on performance, different combinations are used to train several models. For each of the five sizes investigated, the CNN with best generalization capability is selected for further analysis.

2.4. Evolution of Confusion Matrix During Training

According to the relation between actual and predicted class, true positives (TP), true negatives (TN), false positives (FP), false negatives (FN) and cost function losses are computed during training. Logistic Regression class of *LeNet* model is modified to calculate them by comparing assigned labels with network results. Graphs are constructed to evaluate training evolution and performance on validation and test sets. As patch number varies with size, parameter rates are calculated according to Eq.3:

$$T\begin{Bmatrix}P\\N\end{Bmatrix}R=\frac{T\{_N^P\}}{T\{_N^P\}+F\{_P^N\}};F\begin{Bmatrix}P\\N\end{Bmatrix}R=\frac{F\{_N^P\}}{F\{_N^P\}+T\{_P^N\}} \quad (3)$$

Accuracy and precision are determined to monitor CNN performance as in Eq. 4:

$$Accuracy=\frac{TP+TN}{TP+FN+FP+FN};Precision=\frac{TP}{TP+FP} \quad (4)$$

To evaluate CNN abilities for differentiating classes, Receiver Operating Characteristic (ROC) is built varying classification threshold in 0.05 steps, and computing TPR (sensitivity) and FPR (1-specificity). Area under ROC (AUC) of trained models, is integrated numerically from these curves. For each size best network, a Matlab® routine is programmed to build CNN models by loading tuned hyperparameters and trained weights and biases. All test nodules are loaded iteratively, and after convolving, applying the fully-connected layer and logistic regression classifier, output class values are computed for all test patches. Finally, TPR and FPR are calculated modifying classification thresholds.

2.5. Validation With Independent Data

Nodules extracted from the SPIE-AAPM Lung CT Challenge database [25] are used to test trained CNNs sensitivity on an independent dataset. It contains 70 CT scans, with the coordinates of 83 marked nodules. Multiple sections can be visualized depending on their size resulting in the number of patches shown in Table 3:

Table 3. Patches extracted from SPIE-AAPM database

Matrix size	Nodule patches
16x16	149
24x24	338
32x32	561
40x40	634
48x48	634

3. Results

3.1. Training CNN Architectures with Variable Patch Sizes

The objectives of this section are tuning hyperparameters to find networks with best generalization capacity (less test score error) and study their influence on performance.

Tuning starts with 16x16 patches, 40 filters of size (3, 5, 3) per layer, and second layer pooling. Learning rate adjustment is important not only to control training time, but also to find the adequate minimum of the cost function, by setting SGD parameter update steps. Results after modifying its value are summarized in Table 4, showing longer times for small rates. The best validation epoch occurs before for higher rates, while test accuracy reaches a maximum for 0.1, and worsens below. Therefore, this is the value selected for training in this research.

Table 4. Results for different learning rates

Learning rate	Best valid. Epoch	Best valid. Error (%)	Test ccuracy (%)	Simulation time (min)
0.05	85	13.71	86.1	510.1
0.10	54	12.93	87.0	399.0
0.15	31	14.70	85.1	246.7
0.20	6	18.87	80.7	117.1

In Table 5, results after training 32 different CNN architectures for five patch sizes are presented.

Columns 2-8 contain the number of filters per convolutional layer and filter and pool sizes. The next columns show best validation epoch, score errors for validation and test (percentage of misclassified nodules and non-nodules with respect to total patch number), and simulation time.

To analyze these results, the influence of the number of kernels per layer on output is studied firstly. When kernels increase with depth (Ids.1, 3, 5), optimal test performance is achieved for (20, 40, 60), with an accuracy of 87.1% and FPR of 13.8%. Rising the number of filters (Id.5) reduces accuracy in 0.5% and increases FPR in 3.9%, while accuracy drops for less filters (Id.3).

When filter number decreases (Ids.2, 4), accuracy reduces in 1.3% and 3.3% respectively. Using (60, 40, 20) filters instead of (20, 40, 60) gives less accuracy and increases FPR in 5.8%. For (30, 20, 10), FPR rise since epoch 24.

For the five cases considered so far, best test score (12.88%) was obtained for Id.1. To continue with hyperparameter tuning, different filter sizes and pooling combinations were assayed in Ids.6-10 without improvements.

To complete 16x16 patches study, CNNs where the number of kernels is kept constant with depth are trained. Figure 6 represents changes in test accuracy and FPR for three cases (Ids.11-13) with equal number of kernels per layer (maintaining Id.1 structure). The best epoch (vertical line) is obtained before for less kernels. There is a significant improvement in test accuracy from 20 kernels (84.0%) to 40 (87.1%), but not from 40 to 60 (just 0.08%). Although Id.11 and Id.13 produce low errors, test score is slightly better for Id.1.

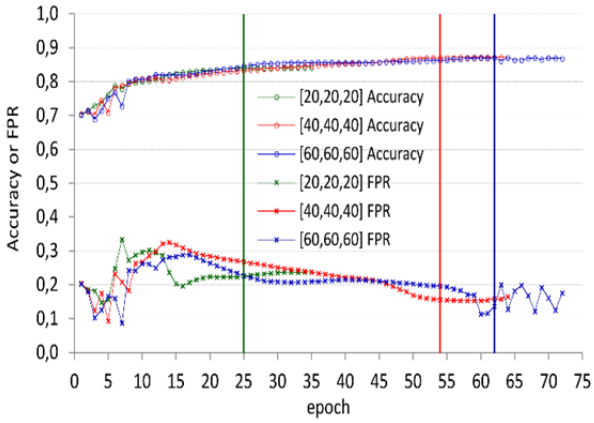

Figure 6. Test performance with equal number of filters per layer (Ids.11, 12, 13)

Table 5. CNN architectures and training results for 16x16 (Ids.1-13), 24x24 (Ids.14-19), 32x32 (Ids.20-25), 40x40 (Ids.26-30) and 48x48 (Ids.31-32)

CNN Id.	Number of filters	Sizes						Best validation epoch	Best validation error	Test score error	Simulation time (min)
		F1	P1	F2	P2	F3	P3				
1[1]	**[20,40,60]**	**3**	**-**	**5**	**2**	**3**	**-**	**63**	**13.06**	**12.88**	**490.2**
2	[60,40,20]	3	-	5	2	3	-	54	14.04	14.19	833.8
3	[10,20,30]	3	-	5	2	3	-	52	14.65	15.09	181.4
4	[30,20,10]	3	-	5	2	3	-	24	15.99	16.17	193.1
5	[30,50,70]	3	-	5	2	3	-	58	13.17	13.40	670.1
6	[20,40,60]	3	2	3	-	3	-	67	13.10	13.82	157.8
7	[20,40,60]	5	-	3	-	3	-	21	21.46	21.76	214.7
8	[20,40,60]	5	2	3	-	3	-	29	16.73	16.66	48.83
9	[20,40,60]	5	-	3	2	3	-	30	16.63	16.75	83.33
10	[20,40,60]	5	-	3	-	3	2	40	14.96	12.04	106.8
11	[40,40,40]	3	-	3	2	3	-	54	12.93	12.98	399.0
12	[20,20,20]	3	-	3	2	3	-	25	16.00	16.05	98.42
13	[60,60,60]	3	-	3	2	3	-	62	12.66	12.90	744.5
14	[20,40,60]	5	2	3	-	3	-	52	16.80	17.02	289.9
15	[20,40,60]	5	-	3	-	3	2	43	16.18	16.24	1008
16	**[40,40,40]**	**5**	**2**	**3**	**-**	**3**	**2**	**86**	**11.58**	**11.68**	**521.2**
17	[20,40,60]	5	-	5	2	5	-	56	14.85	15.02	898.9
18	[20,40,60]	7	-	5	2	5	-	56	13.50	13.59	530.3
19	[20,40,60]	7	-	7	-	5	2	79	13.99	14.10	1282
20	**[20,40,60]**	**7**	**2**	**5**	**-**	**3**	**2**	**59**	**11.70**	**11.86**	**600.6**
21	[40,40,40]	7	2	5	-	3	2	26	13.62	14.26	396.1
22	[20,40,60]	7	-	7	2	5	2	61	12.68	13.11	2236
23	[20,40,60]	3	-	3	2	3	2	32	12.89	13.69	1060
24	[40,40,40]	3	-	3	2	3	2	33	12.80	13.33	1753
25	[20,40,60]	5	2	3	2	3	-	23	12.39	12.83	445.8
26	[20,40,60]	5	2	5	2	3	-	35	13.67	13.38	943.6
27	**[40,40,40]**	**5**	**2**	**5**	**2**	**3**	**-**	**64**	**10.80**	**10.72**	**1419**
28	[40,40,40]	7	2	5	2	5	-	76	12.76	12.42	2042
29	[40,40,40]	9	2	7	-	5	2	72	12.28	12.20	1368
30	[20,40,60]	11	-	11	-	9	2	68	16.98	16.52	5519
31	[20,40,60]	5	2	3	2	3	2	49	11.91	12.50	958.5
32	**[40,40,40]**	**7**	**2**	**7**	**2**	**5**	**-**	**77**	**10.72**	**11.11**	**2452**

[1] *In bold, CNN with less test score error for each size*

To determine how pooling operation affects performance, four examples (Ids.7-10) are compared in Figure 7. Pooling on the third layer produces better accuracy, precision and less FPR. Pooling does not give bad results because matrix size after three convolutions is high, complicating classification in subsequent layers.

For bigger patch sizes, the number of simulations is reduced based on information extracted so far. For 24x24, smaller filters of 3x3 and 5x5 along with two pooling and 40 filters per layer (Id.16) provide the lowest test score error (11.68%).

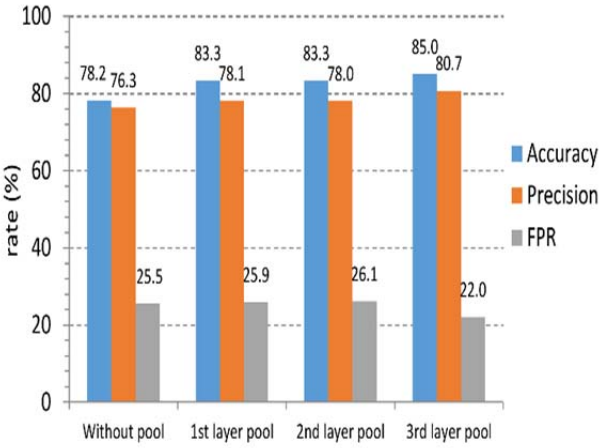

Figure 7. Effect of average pooling acting on different layers

A unique pooling in the first (Id.14) or the last layer (Id.15) and bigger kernels don not enhance CNN performance.

The best CNN for 32x32 is Id.20, with filter sizes (7, 5, 3) and two pooling (11.70% and 11.86% validation and test errors). A constant number of filters worsens results (Id.21), just as no pooling in the first layer (Id.22) and using smaller filters (Ids.23-25).

Regarding 40x40, (40, 40, 40) small filters (5, 5, 3) with two pooling provide this work's best result: 10.80% validation error and 10.72% for testing, with 9.0% FPR (Id.27). To study the influence of filter sizes, testing evolution on four 40x40 CNNs is compared in Figure 8: sizes 5-7-9 result in 2% worse accuracy and FPR. For Id.30, 11x11 and 9x9, complicate learning despite training time (83.5% accuracy). Eventually, two 48x48 CNNs were trained, resulting in validation and test errors of 10.72% and 11.11% for 40 kernels per layer.

In summary, accuracy of lower test error CNNs are: 40x40 (89.2%), 48x48 (88.9%), 24x24 (88.3%), 32x32 (88.1%) and 16x16 (87.1%). The best performance occurs for bigger sizes, although results are comparable.

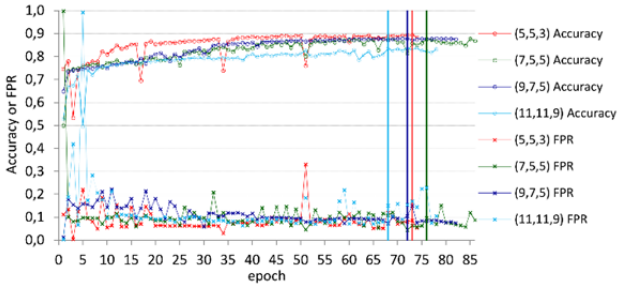

Figure 8. Influence of filter size on test accuracy and FPR in 40x40 CNNs

3.2. Training Evolution of Best Performing CNNs

The evolution of accuracy, precision and FPR on training dataset for top performing CNNs is represented in Figure 9, where vertical lines mark best validation epoch. Changes are smoother for smaller matrix size (particularly 16x16), and become more abrupt as size increases (multiple peaks for 48x48).

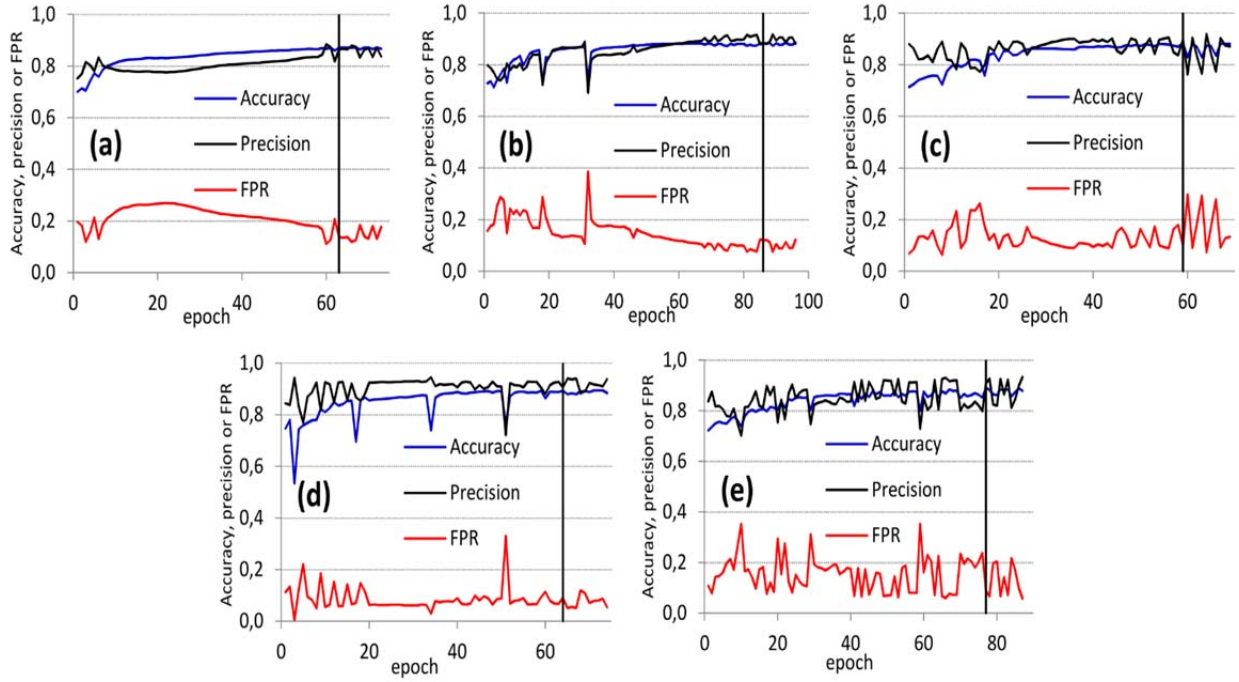

Figure 9. Monitoring of training process: (a) 16x16 (Id.1), (b) 24x24 (Id.16), (c) 32x32 (Id.20), (d) 40x40 (Id.27) and (e) 48x48 (Id.32)

To evaluate their generalization capabilities, a comparison between training and testing is shown in Table 6. As testing patches have not been used to adjust network parameters, the small reduction in accuracy and precision (from -1.8% to -3.3%) demonstrates that overfitting is not affecting training, so CNNs can extrapolate to unknown data. Likewise, the increment in FPR ranges from +1.6% to +3.4%. The amount of test patches, ranging from 16276 (16x16) to 26292 (48x48), reflected in Table 2, enforces data reliability.

Moreover, cost function losses show small differences between datasets in all cases, as represented in Figure 10. This demonstrates that underfitting is not present, so CNNs have enough number of parameters to explain data. Besides, when training and validation curves start to separate, early stopping is applied, avoiding overfitting.

Table 6. % Change in accuracy, precision and FPR from training to testing

Matrix size	Train accuracy (%)	Test change (%)	Train precision (%)	Test change (%)	Train FPR (%)	Test change (%)
16x16	90.4	-3.3	89.8	-3.3	10.4	+3.4
24x24	91.5	-3.2	91.2	-3.1	8.84	+3.2
32x32	90.0	-1.9	91.5	-1.8	8.25	+1.6
40x40	92.4	-3.1	93.4	-2.8	6.42	+2.6
48x48	91.8	-2.9	93.6	-2.9	6.10	+2.7

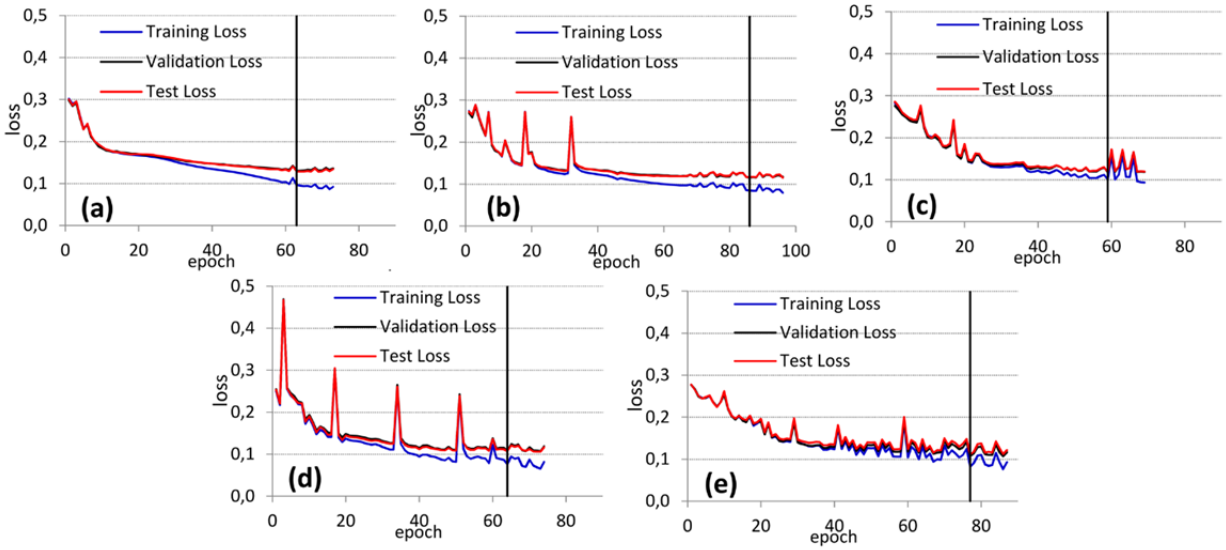

Figure 10. Monitoring of cost function losses during training process for three datasets: training, validation and test datasets for best CNNs

3.3. Evolution of Confusion Matrix During Training

Graphs in Figure 11 show the evolution of TPR, TNR, FPR and FNR on training dataset for best performing CNNs (highlighted in Table 5).

For 16x16, rates vary smoothly from epoch 8, but in all other cases bigger changes can be observed. Data tendency is similar for training, validation and testing datasets. The difference arises in the slight reduction of TPR and TNR and the slight increase in FPR, indicating good CNN extrapolation capacities.

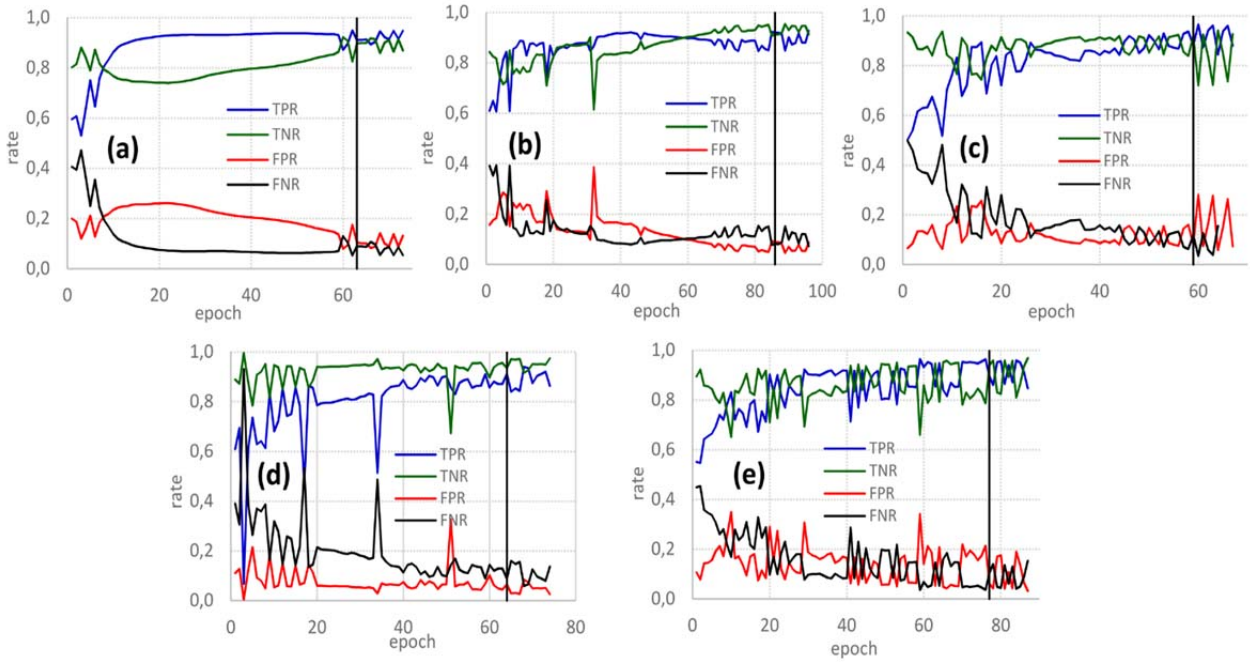

Figure 11. Influence of filter size on test accuracy and FPR in 40x40 CNNs TPR, TNR, FPR and FNR evolution: (a) 16x16 (Id.1), (b) 24x24 (Id.16), (c) 32x32 (Id.20), (d) 40x40 (Id.27), (e) 48x48 (Id.32)

The trade-off between TNR-FPR is more accused than for TPR-FNR when learning algorithm adjusts CNN parameters (peaks in graphs). For 16x16, Figure 11(a), CNNs have a different behavior: TPR is always higher than TNR from epoch 9 until early stopping. For other sizes TPR and TNR evolution curves cross several times except for 40x40, Figure 11(d): TNR are higher than TPR, performing better on non-nodules with lower FPR.

These variations are expressed in Table 7. Opposed changes occur for FPR and FNR. Accuracy and precision loss during testing, explained in Table 6, has more impact on TPR (nodule identification), except for 16x16, because the reduction affects more to TNR (non-nodule detection).

Table 7. True rate changes from training to testing

Matrix size	TPR (% change)	FNR (% change)
16x16	-3.1	-3.4
24x24	-1.9	-0.3
32x32	-2.2	-1.7
40x40	-3.6	-2.7
48x48	-3.0	-2.7

ROC curves, shown in Figure 12, present similar shapes in all cases. For FPR in the range 0-25%, sensitivity achieved by bigger patch CNNs is better. For higher FPR rates all curves have the same behavior.

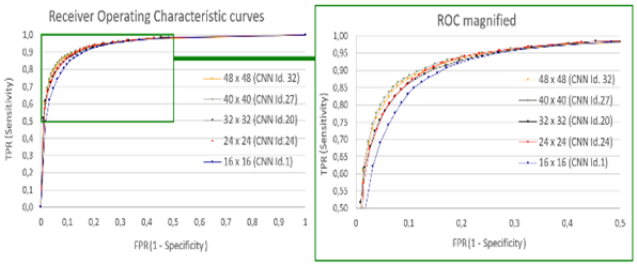

Figure 12. Left: complete ROCs. Right: shoulder part enlarged

Table 8 reflects FPR value to provide a sensitivity of 90%, showing no important differences except for 16x16. Sensitivity differences are below 1% for all cases for FPRs above 0.24. AUC values are close to unity, proving that trained models solve well the problem of class separation (nodule detection), especially for bigger patches.

Table 8. FPR to achieve 90% sensitivity and AUC for best performing CNNs

Matrix size	CNN Id.	FPR (%)	AUC
16x16	1	16.3	0.936
24x24	16	13.2	0.946
32x32	20	14.2	0.947
40x40	27	12.1	0.951
48x48	37	13.3	0.950

3.4. Result Validation with Independent Data

Each of the patches extracted from SPIE-AAPM database is fed into CNN models with learnt parameters. Based on logistic regression results, TPR is calculated (Table 9). High sensitivities, above 90%, are achieved with size 32x32 and bigger. For smaller sizes, sensitivity gets lower, so CNNs have more difficulties learning distinctive nodule features because their morphology is closer to non-nodules. Neuron activation values show that nodules are classified correctly with a high certainty degree, rejecting those misclassified by a small margin.

Table 9. CNN sensitivity and average nodule neuron output

Matrix size	CNN Id.	TPR (%)	FNR (%)	Average nodule output
16x16	1	86.6	13.4	0.799
24x24	16	87.6	12.4	0.852
32x32	20	92.9	7.1	0.912
40x40	27	96.5	3.5	0.944
48x48	37	96.2	3.8	0.948

4. Discussion

In this study, datasets of variable patch size are constructed from LIDC-IDRI database. To test the influence of network hyperparameters and optimize performance, 35 different models are trained and results compared using different metrics.

The number of patches is high for matrix sizes considered (Table 2), compared to other papers. As matrix size increases, data complexity grows, but on the other hand training samples rise, so parameters have more chances to adapt. Best results obtained correspond to 40x40 (test accuracy of 89.3%) and 48x48 (88.9%), so the second effect overcomes the difficulties of learning more complex patterns.

The size of nodule and non-nodule datasets are equal to balance accuracy (includes all data) and precision (affects performance on nodules): less than 2% difference for the best validation epoch in all cases. With the exception of 16 x 16, where accuracy is higher than precision essentially during the whole training process (90.4% vs 89.8% respectively on the best epoch), in the rest of cases their evolution curves cross themselves several times during training.

The main objective of this kind of CAD schemes is to achieve a very high detection sensitivity (TPR), so no suspicious radiological finding is discarded, keeping the number of false positives low. To obtain the maximum number of nodule patches and increase sensitivity, all marked nodules from LIDC-IDRI are selected, regardless of marking radiologists.

One disadvantage inherent to CAD is the number of FPR, a fact that can be discouraging for

radiologists utilizing these systems. This has to be taken into account especially for lung CT screening programs, because of the negative consequences produced by patient recalls (for additional testing or biopsy) due to a FP. It is relevant to note that FPRs obtained are below 14% and even 10% for bigger sizes, as can be seen in Table 6. In training datasets FPRs around 6% are achieved.

The small gap between training and testing accuracy and precision (Table 6), along with the high number of test samples are good indicators of trained CNNs' generalization capabilities. Training is adequately stopped to avoid overfitting and underfitting, as reflected in the evolution of cost function losses. Tests with less restrictive stopping criteria have shown to improve results on training dataset, but without enhancing CNNs generalization capabilities.

The presence of underfitting and overfitting can be discarded evaluating the evolution of cost function losses presented in Figure 10. Learning is stopped when improvements in training data classification have no impact on validation dataset.

For three sizes, best results are obtained with (40,40,40) kernels per layer and for the others with an increasing number with depth (20,40,60), so it cannot be concluded whether is better to use the same number of filters per layer or an increasing number as CNN goes deeper. However, based on this article results, others conclusions can be established:

- A decreasing number of filters with depth compromises CNN performance.
- The same applies when reducing the number of filters below 40. Results were worse in cases with Id.3, Id.4, Id.12, where kernels used were (10, 20, 30), (30, 20, 10) and (20, 20, 20). Test errors obtained were of 15% or higher, resulting in accuracies below 85%.
- Rising the number of filters does not enhance output once a threshold has been passed, increasing training time without rewards.
- Tendency to use bigger filters when increasing input size has to be avoided. Using small filters (3x3 or 5x5) combined with pooling improves results. For small kernels, the lower reduction in input features size joined to the fact that neurons sweep more parts of the input allow the establishment of valuable data links.
- The influence of pooling operations on CNN performance is positive. When no pooling is done results are worse. In case of using a unique pool, better accuracy is obtained if it is applied in a deep layer. The combination of two of these operations enhance results, by controlling the number of neurons in the fully connected layer and easing classification task.

All ROCs have similar shapes, achieving sensitivities above 90% with relatively small FPRs. AUC values obtained are similar or superior to other reported values [11], [13], [14]. CNNs have been tested with nodule patches from an independent database, showing high sensitivities (up to 96%), better than for LIDC-IDRI.

The difficulty of having annotated datasets that cover the great variety of patient cases is partly overcome by LIDC-IDRI. Further tests to infer augmentation effects on generalization will be conducted. In future research related to this work, trained CNNs will be combined to reduce FPR and increase sensitivity, as part of a learning platform dedicated to radiology residents training.

5. Conclusions

Tuning CNN hyperparameters is crucial for learning complex patterns from CT images and obtaining high nodule classification accuracies. Performance of CNNs developed in this work is comparable to other published papers. Test accuracies range from 89.2% (48x48) to 87.1% (16x16). FPR on test dataset are below 9% for bigger patches, and do not exceed 15% for any size.

AUC values for the best five networks are in the range (0.936-0.951), demonstrating strong classification capacities. An independent sensitivity test with nodules from an alternative database provides TPRs above 86.6% in all cases, and superior to 96% for sizes 40x40 and 48x48.

The analysis of training and performance for different CNN configurations, show the importance of selecting the optimal arrangement of kernels for each layer, an adequate learning rate and using small size kernels combined with pooling. Matrix sizes of 40x40 or 48x48 are recommended based on the studied metrics, along with their capacity to detect bigger nodules.

References

[1]. Soffer, S., Ben-Cohen, A., Shimon, O., Amitai, M. M., Greenspan, H., & Klang, E. (2019). Convolutional neural networks for radiologic images: a radiologist's guide. Radiology, 290(3), 590-606. doi: 10.1148 /radiol.2018180547

[2]. Suzuki, K. (2012). A review of computer-aided diagnosis in thoracic and colonic imaging. *Quantitative Imaging in Medicine and Surgery*, *2*(3), 163-176. doi: 10.3978/j.issn.2223-4292.2012.09.02

[3]. Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A., & Bray, F. (2021). Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: a cancer journal for clinicians*, *71*(3), 209-249. doi: 10.3322/caac.21660

[4]. Zhang, G., Jiang, S., Yang, Z., Gong, L., Ma, X., Zhou, Z., ... & Liu, Q. (2018). Automatic nodule detection for lung cancer in CT images: A review. *Computers in biology and medicine*, *103*, 287-300. doi: 10.1016/j.compbiomed.2018.10.033

[5]. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444. doi: 10.1038/nature14539

[6]. Yasaka, K., Akai, H., Kunimatsu, A., Kiryu, S., & Abe, O. (2018). Deep learning with convolutional neural network in radiology. *Japanese journal of radiology*, *36*(4), 257-272. doi: 10.1007/s11604-018-0726-3.

[7]. McBee, M. P., Awan, O. A., Colucci, A. T., Ghobadi, C. W., Kadom, N., Kansagra, A. P., ... & Auffermann, W. F. (2018). Deep learning in radiology. *Academic radiology*, *25*(11), 1472-1480. doi: 10.1016/j.acra.2018.02.018

[8]. 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*. doi: 10.1155/2016/6215085

[9]. Shen, W., Zhou, M., Yang, F., Yang, C., & Tian, J. (2015, June). Multi-scale convolutional neural networks for lung nodule classification. In *International conference on information processing in medical imaging* (pp. 588-599). Springer, Cham. doi: 10.1007/978-3-319-19992-4_46

[10]. Wang, Q., Zheng, Y., Yang, G., Jin, W., Chen, X., & Yin, Y. (2017). Multiscale rotation-invariant convolutional neural networks for lung texture classification. *IEEE journal of biomedical and health informatics*, *22*(1), 184-195. doi: 10.1109/JBHI.2017.2685586

[11]. Alakwaa, W., Nassef, M., & Badr, A. (2017). Lung cancer detection and classification with 3D convolutional neural network (3D-CNN). *Lung Cancer*, *8*(8), 409. doi: 10.14569/IJACSA.2017.080853

[12]. Serj, M. F., Lavi, B., Hoff, G., & Valls, D. P. (2018). A deep convolutional neural network for lung cancer diagnostic. *arXiv preprint arXiv:1804.08170*.

[13]. Tran, G. S., Nghiem, T. P., Nguyen, V. T., Luong, C. M., & Burie, J. C. (2019). Improving Accuracy of Lung Nodule Classification Using Deep Learning with Focal Loss. *Journal of Healthcare Engineering*, *2019*, 5156416-5156416. doi: 10.1155/2019/5156416

[14]. Shen, W., Zhou, M., Yang, F., Yu, D., Dong, D., Yang, C., ... & Tian, J. (2017). Multi-crop convolutional neural networks for lung nodule malignancy suspiciousness classification. *Pattern Recognition*, *61*, 663-673. doi: 10.1016/j.patcog.2016.05.029

[15]. Lyu, J., & Ling, S. H. (2018, July). Using multi-level convolutional neural network for classification of lung nodules on CT images. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 686-689). IEEE. doi: 10.1109/ EMBC.2018.8512376

[16]. Huang, W., Xue, Y., & Wu, Y. (2019). A CAD system for pulmonary nodule prediction based on deep three-dimensional convolutional neural networks and ensemble learning. *Plos one*, *14*(7), e0219369. doi: 10.1371/journal.pone.0219369

[17]. Team, T. T. D., Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., ... & Zhang, Y. (2016). *Theano: A Python framework for fast computation of mathematical expressions*. arXiv preprint arXiv:1605.02688.

[18]. Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I., Bergeron, A., ... & Bengio, Y. (2012). Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*.

[19]. Clark, K., Vendt, B., Smith, K., Freymann, J., Kirby, J., Koppel, P., ... & Prior, F. (2013). The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository. *Journal of digital imaging*, *26*(6), 1045-1057. doi: 10.1007/s10278-013-9622-7

[20]. Armato III, S. G., McLennan, G., Bidaut, L., McNitt-Gray, M. F., Meyer, C. R., Reeves, A. P., & Clarke, L. P. (2015). Data from lidc-idri. *The cancer imaging archive*, *10*, K9. doi: 10.7937/K9/TCIA.2015.LO9QL9SX

[21]. Armato III, S. G., McLennan, G., Bidaut, L., McNitt-Gray, M. F., Meyer, C. R., Reeves, A. P., ... & Clarke, L. P. (2011). The lung image database consortium (LIDC) and image database resource initiative (IDRI): a completed reference database of lung nodules on CT scans. *Medical physics*, *38*(2), 915-931. doi: 10.1118/1.3528204

[22]. Lampert, T. A., Stumpf, A., & Gançarski, P. (2016). An empirical study into annotator agreement, ground truth estimation, and algorithm evaluation. *IEEE Transactions on Image Processing*, *25*(6), 2557-2572. doi: 10.1109/TIP.2016.2544703

[23]. LISA Lab. (2015). *Deep Learning Tutorial*. University of Montreal, Canada.

[24]. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324. doi:10.1109/5.726791

[25]. Armato III, S. G., Hadjiiski, L., Tourassi, G. D., Drukker, K., Giger, M. L., Li, F., ... & Clarke, L. P. (2015). Spie-aapm-nci lung nodule classification challenge dataset. the cancer imaging archive. doi: 10.7937/K9/TCIA.2015.UZLSU3FL