# Application of Deep Learning Neural Network for Classification of TB Lung CT Images Based on Patches

Xiaohong Gao [1], Yu Qian [2]

[1] Department of Computer Science, Middlesex University, London NW4 4BT, United Kingdom

x.gao@mdx.ac.uk

[2] Cortexcia Vision Systems, London SE1 8RT, United Kingdom.

yu.qian@cortexica.com

**Abstract.** In this work, convolutional neural network (CNN) is applied to classify the five types of Tuberculosis (TB) lung CT images. In doing so, each image has been segmented into rectangular patches with side width and high varying between 20 and 55 pixels, which are later normalised into 30x30 pixels. While classifying TB types, six instead of five categories are distinguished. Group 6 houses those patches/segments that are common to most of the other types, or background. In this way, while each 3D dataset only has less than 10% distinguishable volumes that are applied to perform the training, the rest remains part of the learning cycle by participating to the classification, leading to an automated process to differentiation of five types of TB. When tested against 300 datasets, the Kappa value is 0.2187, ranking 5 among 23 submissions. However, the accuracy value of ACC is 0.4067, the highest in this competition of classification of TB types.

**Keywords:** Deep learning, patch-based, classification, Tuberculosis disease.

## 1 Convolutional Neural Network (CNN

### 1.1 A Subsection Sample

Deep learning models refer to a class of computing machines that can learn a hierarchy of features by building high-level attributes from low-level ones [1, 2] , thereby automating the process of feature construction. One of these models is the well-known convolutional neural network (CNN) [2]. Consisted of a set of algorithms in machine learning, CNN comprises several (deep) layers of processing involving learnable operators (both linear and non-linear), and hence has the ability to learn and build high-level information from low-level features in an automatic fashion [3]. Stemming from biological vision processes, a CNN applies a feed-forward artificial neural network to simulate variations of multilayer perceptrons whereby the individual neurons are tiled in such a way that they respond to overlapping regions in the visual field [4]. As a direct result, these networks are widely applied to image and video recognition. Specifically, CNNs have demonstrated as an effective class of models for understanding image content, proffering state of the art results on image recognition, segmentation, detection and retrieval. For example, when trained with appropriate regularization, CNNs can achieve superior performance on visual object recognition tasks without relying on any hand-crafted features, e.g. SIFT, SURF. In addition, CNNs have been shown to be relatively insensitive to certain variations on the inputs [5]. Significantly, recent advances of computer hardware technology (e.g., Graphics Processing Unit (GPU)) have propitiated the implementation of CNNs in representing images.

Theoretically, CNN can be expressed in the following formulas. For example, for a set of training data $\left(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\right)$, where image $\boldsymbol{x}^{(i)}$ is in three-dimension (inclusive of RGB channel as the 3rd dimension) and $\boldsymbol{y}^{(i)}$ the indicator vector of affiliated class of $\boldsymbol{x}^{(i)},$ the feature maps of an image, namely, $w_1, \dots, w_{L,}$ will be learnt based on CNN by solving Eq. (1).

$$\underset{\boldsymbol{w}_1,\dots,\boldsymbol{w}_L}{argmin} \frac{1}{n} \sum_{i=1}^{n} \ell(f(\mathbf{x}^i; \boldsymbol{w}_1, \dots, \boldsymbol{w}_L), \boldsymbol{y}^i) \tag{1}$$

Where $\ell$ refers to a suitable loss function (e.g. the hinge or log loss) and $f$ the selected classifier.

To obtain these feature maps computationally, in a 2D CNN, convolution is conducted at convolutional layers to extract features from local neighbourhood on the feature maps acquired in the previous layer. Then an additive bias is applied and the result is passed through a sigmoid function as formulated in Eq. (2) mathematically in order to obtain a newly calculated feature value $\boldsymbol{v}_{ij}^{xy}$ at position $(x, y)$ on the $j_{th}$ feature map in the $i_{th}$ layer.

$$\boldsymbol{v}_{ij}^{xy} = tanh\left(\boldsymbol{b}_{ij} + \sum_{m} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \boldsymbol{w}_{ijm}^{pq} \boldsymbol{v}_{(i-1)m}^{(x+p)(y+q)}\right) \tag{2}$$

where the notations of those parameters in Eq. (2) are explained in Table 1.

**Table 1.** Notations of parameters in Eq. (4).

| Parameter | Notation | Parameter |
|---|---|---|

| | | |
|---|---|---|
| $tanh(.)$ | hyperbolic tangent function | $tanh(.)$ |
| $m$ | index over the set of feature maps in the $(i-1)th$ layer | $m$ |
| $\boldsymbol{b}_{ij}$ | bias for the feature map $f$ in Eq. (1). | $\boldsymbol{b}_{ij}$ |
| $\boldsymbol{w}_{ijk}^{pq}$ | value at the position (p, q) of the kernel connected to the $k_{th}$ feature map | $\boldsymbol{w}_{ijk}^{pq}$ |
| $(p,q)$ | 2D position of a kernel | $(p,q)$ |

As a result, CNN architecture can be constructed by stacking multiple layers of convolution and subsampling in an alternating fashion. The parameters of CNN, such as the bias $\boldsymbol{b}_{ij}$ and the kernel weight $\boldsymbol{w}_{ijk}^{pq}$ are trained using unsupervised approaches [6, 7].

## 2. Datasets

This work is to response to the challenge task of automatic detection of Tuberculosis (TB) types using datasets that is organised by ImageCLEF as put forward in [8, 9], part of CLEF conference to take place in Dublin [10, 11].

As explained in [8], data are collected for the evaluation of ImageCLEF tuberculosis competition with 500 datasets of 3D CT lung images (512x512xdepth) for training and further 300 for testing. Among 500 datasets there are 140, 120, 100, 80 and 60 respectively for five TB types of for Infiltrative (type 1), Focal (type 2), Tuberculoma (type 3), Miliary (type 4) and Fibro-cavernous (type 5).

For the training data, each 3D dataset firstly undergoes pre-processing stage, whereby those artefacts are removed, i.e., slices that contain little visual content will be excluded. In this way, each dataset includes slices between 100 and 170. Then upon each slice, patches of varying sizes are created based on the lung boundary is created from its mask file [12]. Figure 1 illustrates the pre-processing by depicting the montage of original dataset (top-left), segmented (top-right) and patches segmented from one slice (bottom) that contains at least 80% of lung contents, checked by its mask.

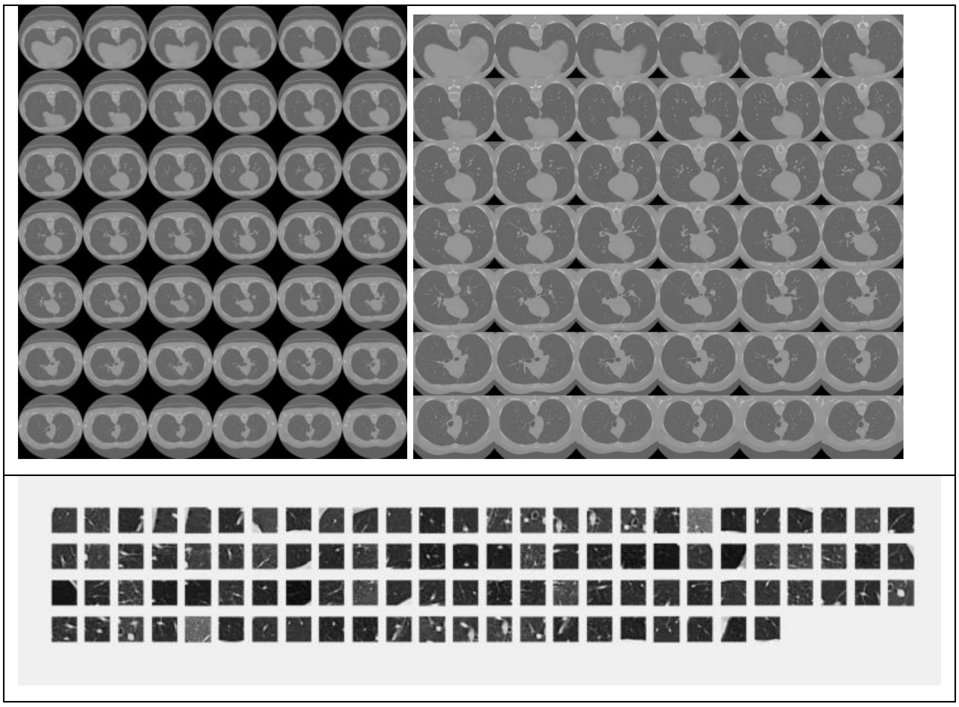


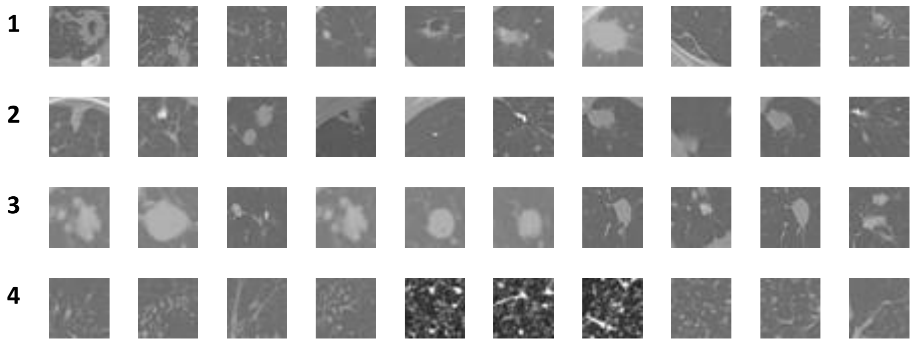
**Fig. 1.** Data pre-processing stage. Top-left: original datasets; top-right: Segmented; bottom: patches from segmented slices.

At present, two sizes of patches are fixed, including 30×30 and 50×50 pixels in an attempt to cover both small features of nodules and big characteristics of cavity, which are overlapping each other. As a result, each slice entails around 150 patches, leading to more than 1500 patches for each dataset being generated with 100 slices. Since more than 90% of patches are common among all five TB types, for each TB type, around 1000 to 2000 patches are selected, which demonstrate distinguishable visual features and are elaborated below.

For classes 4 (Miliary) and 5 (Cavity), the visual features are apparent with wide-spread dissemination of small spots (i.e. Mycobacterium tuberculosis) and dominating holes (cavities) respectively. However, for classes 1 to 3, visual features are not easily distinguishable. Therefore, the selection starts from background patches. In another words, those patches with very similar visual appearances (background) are pulled together to form Class 6 for the training, leaving remaining ones assuming to be representative. It is very likely that those background patches contain information from their individual classes, which does not appear to pose huge problems since the results from classes 1 to 5 are considered when it comes to classification. class 6 do not contribute to the final classification results. The interesting fact is that about 20% individual patches in Class 6 have been classified into their own individual groups, i.e. Types 1 or 2 or 3 instead of Class 6. In summary, in addition to 5 classes to be categorised, class 6 is created to include those patches that appear to be common among the first 5 classes, which again contains about 1000 patches, which is illustrated in Table 2. Figure 2 exemplifies the five types of patches that are applied in the training, which are all normalised into 30x30 pixels. As discussed above, type 6 remains an extra class to contain those common features shared between types 1 to 5 with patches randomly selected from type 1 to 3, and 5 without type 4, this is because the outstanding features in type 4 with miliary TB spread nearly every slice. Since the patches in each class are randomly generated by computers and selected manually, it is more likely that many patches belong to both of the five classes and class 6, which should not cause too much concerns as the final decision making is based upon the probabilities obtained for the first 5 classes. In addition, the patches belong to each subject will stay together when dividing between training and test data.

**Table 2.** The lists the number of patches in the training process.

| | **1** | **2** | **3** | **4** | **5** | **6** |
|---|---|---|---|---|---|---|
| **Training** | 2000 | 1000 | 1000 | 1500 | 1400 | 1000 |
| **Testing** | 267 | 161 | 33 | 98 | 91 | 64 |



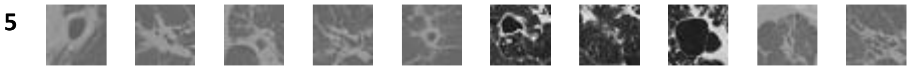


**Fig. 2.** The examples of patches in each class, which are applied for training. From top to bottom, Types 1, 2, 3, 4, 5, and 6.

## 3. The Architecture of Deep Learning Convolution Neural Network (CNN) with SVM

The CNN network that has been designed for this task is built upon matConvNet package written using Matlab software[1]. Seven layers of CNN have been designed with input data of 30x30 pixels. The filter sizes for each layer are of (6,6), (4,4), (3,3), (2,2), (2,2), (3,3), and (1,1) respectively. At layer 6, instead of scoring features into one of the six classes using *Softmax* approach that applies cross-entropy loss interpreting the scores as (unnormalised) log probabilities for each class, this study applies support vector machines (SVM) that adapts hinge loss to encourage the correct class to have a score higher by a margin than the other class scores. In this way, each class has a distinguishing boundary. For example, if a feature belongs to one of the two classes with probabilities of 0.51 and 0.49 respectively, the classification is not quite convincing. However, a score with a higher margin, e.g., 0.65, is more acceptable. Figure 3 illustrates the CNN architecture that is applied in this study. When those patches are stitched together for each individual subject, class 6 is ignored, i.e., only scores of first five classes are taken into consideration. In this way, computerised selection of patches is made possible to ensure that any kind of patch belongs to a group .

[1] MatConvNet: http://www.vlfeat.org/matconvnet/. Retrieved in May 2017.

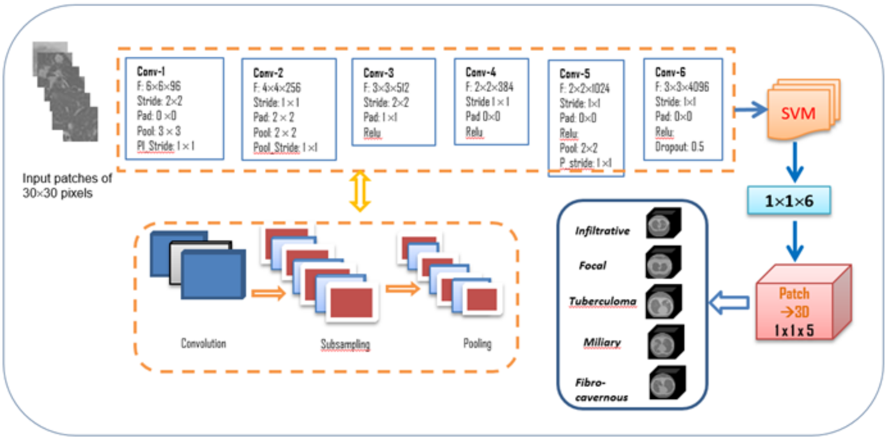


**Fig. 3.** The CNN network applied in this study**.**

## 4. Results

Within the existing of 500 datasets that have known ground truth, with reference of patch-wise, the classification results are given in Table 3 in percentage with an overall classification rate of 96%.

**Table 3.** Confusion matrix for Patch-wise accuracy.

| **Class** | **1** | **2** | **3** | **4** | **5** | **6** |
|---|---|---|---|---|---|---|
| **1** | **0.87** | 0.09 | 0.03 | 0 | 0.01 | 0 |
| **2** | 0.02 | **0.96** | 0 | 0 | 0.01 | 0.01 |
| **3** | 0 | 0.03 | **0.97** | 0 | 0 | 0 |
| **4** | 0.04 | 0 | 0 | **0.96** | 0 | 0 |
| **5** | 0.03 | 0 | 0 | 0.01 | **0.96** | 0 |
| **6** | 0 | 0.04 | 0 | 0 | 0 | **0.96** |

During the competition, 300 testing data were supplied at [2] and are ranked based on Kappa value [2]. This work was ranked 5 out of 23 submissions with 0.2187 Kappa value whereas ACC value was the highest (0.4067), implying that the proposed patch-based deep learning network has achieved averaged best accuracy results in the competition.

## 5. Conclusion and Future directions

While the overall classification appears to be reasonable, with 86.49% accuracy rate, the result for Type 1 only sustains 50%. Since the testing datasets only include 8 subject samples in this type, the remaining work is to evaluate the results from real testing datasets when the ground truth is obtained. In order to obtain higher accuracy, it is recommended that medical knowledge should be embedded. Additionally, 3D segments should be also included to further enhance the characteristics that 3D datasets entail.

## References

1. Dicente Cid Y., Kalinovsky A., Liauchuk V., Kovalev V., Müller H., Overview of ImageCLEFtuberculosis 2017, Predicting Tuberculosis Type and Drug Resistances, in CLEF 2017 Labs Working Notes of CEUR Workshop Proceedings, CEUR-WS.org (http://ceur-ws.org), Dublin, Ireland, September 11-14, 2017.
2. Dicente Cid Y., Jiménez-del-Toro O. A., Depeursinge A., and Müller H., Efficient and fully automatic segmentation of the lungs in CT volumes. In: Goksel, O., et al. (eds.) Proceedings of the VISCERAL Challenge at ISBI. No. 1390 in CEUR Workshop Proceedings, Apr 2015.
3. Fukushima K., Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, *Biol. Cyb.*, 36: 193–202, 1980.
4. LeCun Y., Bottou L., Bengio Y., and Haffner P., Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, 86(11): 2278–2324, 1998.

[2] imageCLEF: http://www.imageclef.org/2017/tuberculosis. Retrieved in May, 2017.

5. LeCun Y., Huang F.J. , Bottou L., Learning methods for generic object recognition with invariance to pose and lighting, *Processings of Computer Vision and Pattern Recognition (CVPR)*, 2: II-97-104, 2004.
6. *LeCun* Y.*, Bengio Y., Hinton G., Deep Learning, Nature, 521: 436-444, 2015.*
7. Ranzato M., Huang F.J., Boureau Y., LeCun Y., Unsupervised learning of invariant feature hierarchies with applications to object recognition, *Processings of CVPR* 2007, pp1-8, 2007,
8. Ji S., Xu W., Yang M., Yu K., 3D convolutional neural networks for human action recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35 (1): 221-231, 2015.
9. Bogdan I., Müller H., Villegas, M., Arenas H., Boato G., Dang-Nguyen D., Dicente C., Eickhoff C., Garcia Seco de Herrera A., Gurrin C., Islam, B., Kovalev V., Liauchuk V., Mothe J., Piras L., Riegler M., and Schwall, I., Overview of ImageCLEF 2017: Information extraction from images, in Experimental IR Meets Multilinguality, Multimodality, and Interaction 8th International Conference of the CLEF Association, CLEF 2017, LNCS 10456, 2017, Dublin, Ireland, September 11-14, 2017.
10. Jones G. J. F., Lawless S., Gonzalo J., Kelly L., Goeuriot L., Mandl T., Cappellato L., and Ferro N. (eds.), Proceedings of Experimental IR Meets Multilinguality, Multimodality, and Interaction 8th International Conference of the CLEF Association, CLEF 2017, LNCS 10456, Dublin, Ireland, September 11-14, 2017.
11. Cappellato L., Ferro N., Goeuriot L., and Mandl T. (eds.), CLEF 2017 Labs Working Notes, CEUR-WS Proceedings, 2017.
12. Vedaldi A., Lenc K., MatConvNet Convolutional Neural Networks for MATLAB , *Proceedings of the 23rd ACM international conference on Multimedia*, pp 689-692, 2015.