COMP 551 - Applied Machine Learning - Project IV: Lung Tumor Location and Identification using a Custom CNN: ICLR 2018 Reproducibility Challenge

Afuad Hossain
afuad.hossain@mail.mcgill.ca
260621073

Xin Tong Wang
xin.t.wang@mail.mcgill.ca
260640319

Qianyuan Sun
qianyuan.sun@mail.mcgill.ca
260769678

I. Preface

To access the paper that is subject to our reproducibility challenge, simply visit the the openreview.net link: https://openreview.net/forum?id=rJr4kfWCb. For more information on the dataset utilized, simply visit the LUNA2016 web page at https://luna16.grand-challenge.org. All information needed to download the data and visualize it are provided within the confines of the site. All relevant code developed in relation to the reproducibility challenge can be accessed via the following Github repository: https://github.com/ExTee/COMP551-Final

II. Introduction

Lung cancer is the leading cause of cancer deaths in the world, with roughly 1.67 million deaths a year (WHO, 2017). Early, efficient, effective and most importantly correct diagnoses of patients still remain the most important step in the treatment of cancer. However, false positive rates in clinical settings remains especially high (Rivera et al., 2013). Recent development of machine learning methods in medical imaging have provided doctors with a complementary set of tools to tackle these issues. However, past CNN architectures still remain not well suited for specialized medical tasks (cancer nodule classification). As such, we seek to justify the claims by the original authors, that we can in fact develop a CNN that best accurately identifies benign and cancerous growths in patient CT scans, whilst dramatically reducing false positives in the process.

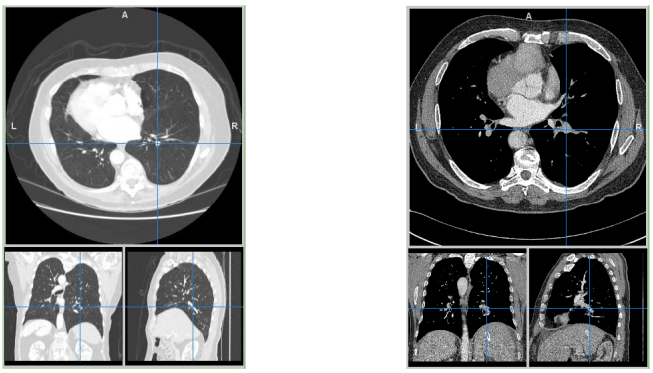
This paper highlights this process by which we sought to reproduce the claims of the original authors. Notable claims included 1) the proposed CNN architecture classifies cancer nodes at an accuracy of 99.79% and 2) false positive rates are reduced to less than 0.01%. While the entirety of the CNN architecture was able to be reproduced with comparable accuracy and false positive rates, we suggest that the nature of the dataset, ambiguities in the original authors preprocessing and testing methods, would present several hindrances to groups that solely base their reproducibility efforts on the information provided.

III. Related Work

Krizhevsky et al. developed one of the most seminal CNN architectures in 2012 (aptly named AlexNet). This CNN was able to classify 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into 1000 different classes while achieving top-1 and top-5 error rates of 37.5% and 17.0%, a marketed improvement from past infrastructures. In fact, AlexNet, due to its effectiveness, is chosen in the reproduced paper as a baseline CNN to compare their proposed CNN architecture's performance to.

Pereira et al. developed a CNN for the purpose of automatic segmentation in MRI scans of brain tumors, utilizing small 3x3 kernels, allowing for deeper architectures and limiting the effects of over-fitting. The developed CNN finished 1st in the Brain Tumor Segmentation Challenge 2013 database (BRATS 2013). A majority of the reproduced paper's proposed CNN architecture is based on the Pereira et al. CNN architecture.

IV. Problem Representation



(a) Cancer (b) Benign

Fig. 1. *Reproduced from the original authors, an example of PNG images obtained from the cross-section of CT scans via preprocessing methods

A. Dataset

The dataset utilized, as outlined by the original authors are derived from the LUNA2016 dataset. Much of this dataset was originally created from the publicly available LIDC/IDRI database, a database of lung CT scans. The LIDC/IDRI database contains nodule annotations which were collected during a two-phase annotation process using 4 experienced radiologists.

LUNA2016 has organized much of the LIDC/IDRI database so as to make it readily available to groups working on medical

imaging classifiers. Among some of the organization choices include dividing the complete dataset into 10 subsets, with LUNA 2016 recommending each subset be used in 10-fold cross validation accordingly. In total, the LUNA2016 dataset contains 888 total CT scans. Such scans may either be used to develop nodule detection algorithms to identify the location of nodules within each CT scan or be used to classify identified nodules as either benign or cancerous. The latter track was chosen in this paper and the reproduced paper. In total, LUNA2016 has provided 551,065 candidates to be classified within the provided CT scans. Each candidate has an x, y, and z position in world coordinates and a classification as either benign (0) or cancerous (1). Interestingly, the candidates listed represent 1120 out of 1186 nodules detected in all CT scans. It should also be noted that there can be multiple candidates per nodule.

CT images are stored in MetaImage (mhd/raw) format. Each .mhd file is stored with a separate .raw binary file for the pixel data. the dimensions of these images are 512x512xZ with Z varying in length depending on the height of patients scanned.

B. Data Preprocessing

To get the images of the cancer nodules suitable for training and testing, we first utilize SimpleITK to read a list of .mhd files (the native CT scan file format) and visualize each image they contain. Each image is stored into a numpy array. We then extracted two key information: 1) image origin and 2) pixel spacing of each numpy. In the .csv file containing candidate information, the coordinates of the candidates are given in world coordinates. As such, we also need to transform theses world coordinates into voxel coordinates so as to have the locations of the nodules in the extracted PNG images. To calculate the voxel coordinates, we used the following equation:

$$C_v = \frac{C_w - Origin}{Spacing} \tag{1}$$

where Cv are the voxel coordinates, Cw are the world coordinates, and Origin, Spacing are the information that we extracted before. According to the voxel coordinates, we can get the patch coinciding with the presence of the nodules for each candidate. Patches were classified and saved in different paths based on whether the label of images is 1 or 0. Images were then saved in PNG format, so that they could be inputted in to train the corresponding CNN.

V. Algorithm Implementation

A. Overview

In Machine Learning, a Convolutional Neural Network(CNN) is a class of neural network using a variation of multilayer perceptrons designed to require minimal preprocessing. It has been applied to image analysis successfully. Compared to standard neural networks, CNN makes the explicit assumption that the inputs are images, which allows us to encode certain properties into the architecture. In particular, the layers of a CNN have neurons arranged in 3 dimensions: width, height, depth. These then make the forward function more efficient to implement and vastly reduce the amount of parameters in the network.

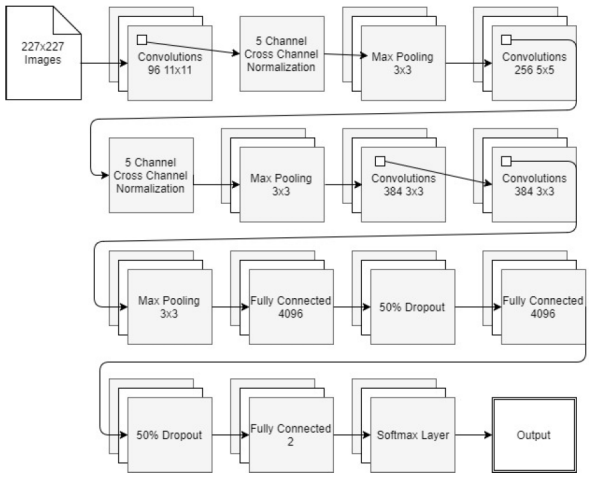


Fig. 2. *Reproduced from the original authors. Alex Krizhevsky's (2012) CNN architecture. Used by the original authors as their baseline CNN. Image above is from the original paper and actually does not reflect the actual architecture

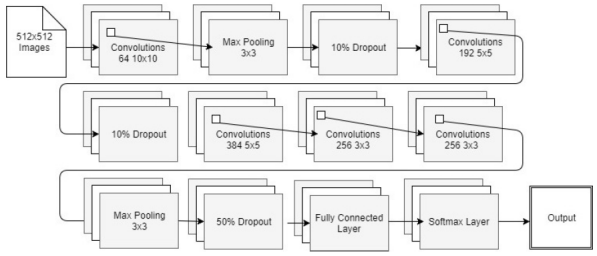


Fig. 3. *Reproduced from the original authors. The original authors' proposed CNN architecture (AlexNet). Image above is from the original paper and actually does not reflect the actual architecture

B. CNN Architecture

The authors propose a CNN with an architecture which is based on that of CNNs for Brain image segmentation from previous papers from Pereira et al. (2016) and Simonyan and Zisserman (2014). The proposed CNN comprises 7 Convolutional layers connected to a Dense layer for classification. The main differences between the proposed CNN and AlexNet are the two extra Convolutional layers, as well as the extra dropout layers. The authors propose that such a design allows them to construct deeper architectures while reducing over-fitting. The authors also mention that they obtained a reduced False Positive rate through their architecture.

In order to reproduce the authors' results, we constructed a network following the proposed architecture from their paper. The CNN was implemented using Keras with TensorFlow backend. As shown in fig. 4, every layer of our network follows the description of the authors' network, with the exception of the fully-connected layers due to the lack of specifications. Thus, our architecture contains 7 convolutional

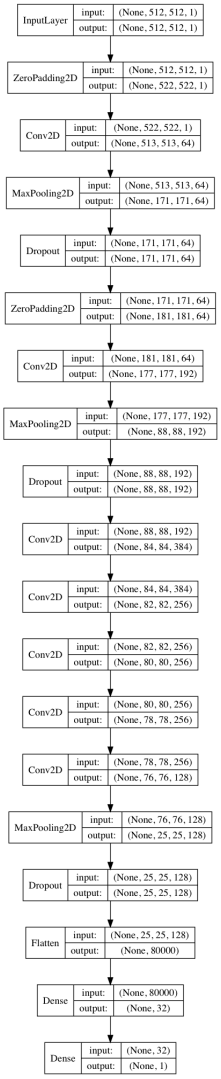


Fig. 4. Our Keras CNN architecture

layers, along with their associated padding, pooling and dropout layers. The authors did not specify the number of dense layers at the end of their convolutional layers. Thus, we opted for flattening the convolutional layer, and connecting it to two fully-connected layers. The authors also did not specify which activation function they used for the dense layers, nor did they indicate their gradient descent algorithm. Due to the output being binary, two options can be considered: 1) A Softmax activation function with categorical cross-entropy loss function and one-hot encoded inputs or 2) A Sigmoid activation function with binary cross-entropy loss function. We opted for the latter, thus outputting an integer value of 0 or 1 depending on the predicted output. Furthermore, the Adam optimizer was chosen as it appears to perform well.

Due to the data size being substantial, it is impossible to load the whole set into memory. Thus we opted for batch processing of the .png files using Keras's ImageGenerator function while generating from directory. The advantages are two-fold: image size is verified and resized before feeding into the network and no manual loading is required to feed pixels into the neural network, because the function accepts .png files directly. The main drawback is the complexity of cross-validation due to it not supporting automatic partitioning of the data. Thus, partitioning for validation was done manually. The resulting network was trained on an NVidia GTX970 GPU for around 10 hours.

VI. Testing and Validation

A. Results

Training Sets	Test Sets
$0, 1, 4, 5, 7, 8, 9$	$2, 3, 6$

Table 1. Randomized subset selection for training and testing sets. 70% allocated for training purposes, 30% for testing purposes.

Validation Set	Training Sets	Baseline*	CNN
0	$1, 4, 5, 7, 8, 9$	99.79%	99.79%
1	$0, 4, 5, 7, 8, 9$	99.71%	99.77%
4	$0, 1, 5, 7, 8, 9$	99.73%	99.83%
5	$0, 1, 4, 7, 8, 9$	99.80%	99.75%
7	$0, 1, 4, 5, 8, 9$	99.80%	99.81%
8	$0, 1, 4, 5, 7, 9$	99.72%	99.73%
9	$0, 1, 4, 5, 7, 8$	99.78%	99.83%

Table 2. 7-fold cross validation on the 7 training sets set aside. Baseline refers to a classifier that classifies all examples as benign. Accuracy = (tp + tn)/(tp + tn + fp + fn)

Classifier	Accuracy	FPR	MCC
$Baseline*$	99.73%	0.00%	0.00
AlexNet	99.72%	54.26%	0.37
Proposed CNN	99.79%	0.00%	0.56
$OurCNN$	99.78%	0.11%	0.58

Table 3. Results obtained on the test subsets. Italicized classifiers are novel results. Baseline refers to a classifier that classifies all examples as benign. Accuracy = (tp + tn)/(tp + tn + fp + fn), FPR = fp/(tn + fp), Matthews Correlation Coefficient (MCC) = (tp * tn - fp * fn)/sqrt((tp + fp)(tp + fn)(tn + fp)(tn + fn))

B. Validation Methods

Due to some ambiguities in how validation was conducted in the original paper, we adjusted our validation process to our best ability based on tidbits of information provided by the authors. This meant first, to divide the dataset set into training set and test set, based on a partition of 70% training and 30% testing. This was done by simply randomly diving the already partitioned subsets into 7 and 3 respectively, with the actual the amount of CT scans divided of no concern to us. The training set was thus used in the validation process. While it is assumed a 10-fold validation process was used by the authors, we utilized a 7-fold validation due to its ease of application (training data was already originally partitioned into 7 subsets). While not exactly the same procedures as the original authors,

the paper failed to provide enough information as to how cross validation was conducted an it remains unclear whether the dataset was partitioned into training and testing data at all when analyzing the results table.

VII. Discussion

A. Data Preprocessing

While slightly differing from the original paper, it can be assumed with high confidence that the retrieved PNG images from the original CT scans did not differ. It should be noted though however that the authors omitted certain details with regards preprocessing steps. This may have had lead to significant differences in preprocessing were it not for LUNA2016 providing tutorials by which to convert medical images into either PNG and JPEG format. Assuming that the authors utilized the same methods to convert .mhd CT scan files into the PNG files of candidate nodule cross sections, the final end-product, that is, PNG images for each candidate, is expected to be the same. Regardless, if simply taken for its face-value, preprocessing the obtained dataset would not be possible solely based on the information provided by the authors.

B. CNN architecture

The CNN proposed by the authors contains 7 Convolutional layers and a dense layer. Convolutional layers have filter of size 3x3, which works as segment of the images. The authors explain that using several of these convolutional layers helps to decrease False Positive Rate. Compared with AlexNet, the proposed CNN reduces over-fitting and have a higher accuracy. The original paper's description of the architecture is rather clear and concise, and provides a significant amount of detail. However, some information has been left out, such as the amount of dense layers, the loss function or the optimizer. These had to be inferred and chosen based on our judgement. Moreover, the figure of the network provided by the authors does not reflect the actual architecture. More specifically, it omits two convolutional layers before the Pooling layer.

C. Results

As seen in table 3, the obtained results for our CNN are similar to that of the proposed CNN. This suggests that preprocessing and our developed CNN architecture were reproduced in similar manners to that of the original authors. Interestingly enough, a baseline classifier, one that simply classifies all examples as benign scored relatively high as well in terms of accuracy, suggesting the number of cancerous nodules in the dataset being of very low quantities from the start. Logically enough, this baseline classifier scored poorly in terms of false negative rates (100%) and MCC (0) seeing as all cancerous nodules would be classified as benign. However if one were to focus solely on false positive rates as so many medical classifiers do, a baseline classifiers scores 0%. Evidently, such a classifier defeats the purpose such projects. A baseline as described above simply serves to pass caution on the conclusions applied to obtained results about what they really mean and whether they are in fact marketed improvements. As for results concerning the proposed CNN, our CNN performed at relatively even rates. However this result simply does not have the same impact when understanding that the baseline can have same desired effects. It thus becomes imperative to dive deeper into the nature of the dataset. Interestingly enough, out of all candidate nodes (551,073), only 1355 candidates are annotated as cancerous. It'd be interesting to see how such classifiers fare with additional data where cancerous to benign distributions are more varied.

D. Validation Methods

Several ambiguities exists as to how exactly the validation process was conducted. The first source of concern comes from the fact that the authors mention dividing the data for testing purposes. Apart from providing the ratios of training to testing partitioning, no further details are given as to how exactly the data was divided. As such, it was unclear whether the data was partitioned simply based on the LUNA2016 partitions or rather if each individual subset was divided into training and testing data. Furthermore, this confusion translates itself to the validation process, where it is in fact unclear whether the entirety of the dataset was utilized for the validation process. Based table entries of their results, it is assumed a 10-fold cross-validation was conducted. However, the validation set was noted as an entire subset as originally defined by LUNA 2016. This is problematic due to that fact that there is no mention as to what or rather what parts of these subsets are in fact testing data. It can be even reasonable to assume, due to this confusion, that a 10-fold cross validation was conducted using all of the data, and thereafter, the model was tested on the stipulated testing data, which is in fact was also part of the training data as well! This premise would is deemed irrational, since test data is not to be touched during the training process. In sum, the multiple levels of confusion does not bode well for groups vying to reproduce the paper in its entirety.

E. Reproducibility criterias

The data sets used by the authors have been referenced, although their download links have not explicitly been provided. Unfortunately the authors explained their data partitioning in an ambiguous way, which lead to a large amount of time spent on our part to determine the best approach to faithfully replicate the authors' results. The authors indicated that they performed preprocessing using MATLAB, but unfortunately did not provide any indications of the libraries used for their neural networks. In turn, the names and versions of dependencies were not found either. Randomization details about partitioning have not been provided, and thus, we decided to shuffle the input batches before feeding the data into the network. The authors provided very clear infrastructure and computation requirements. They provided a detailed architecture of their CNN, and stated that training took 1.6 hours on four NVidia GTX1080 GPUs. Most of the reimplementation effort consisted of attempting to understand the authors' reasoning behind their partitioning. Also the data set used

was large, which lead to storage and memory constraints. Overall, the expertise necessary to reproduce the results of the authors is not unreasonable. Finally, our team has attempted to contact the authors for possible clarifications about methods and potential code have been in vain, as they did not respond to our inquiries on Openreview.

VIII. Conclusion

In sum, while the results of the paper may have been replicated to a satisfiable degree, the process by which this was done was not so straightforward. The original authors did not list a detailed procedure for many aspects of the project, leaving other groups to assume many points, as we did. Among the more notable points of confusion include, a lack of preprocessing procedure, minor ambiguities and mistakes in CNN architecture design, a confusing testing and validation process and a lack of insight into the actual results obtained. It can be easily concluded, that if groups were to be entirely left to reproduce the paper based on the information provided, severe hindrances would present themselves to said group. As such, this task has demonstrated the importance of being clear and concise when writing scientific literature.

IX. Statement of Contributions

Afuad Hossain contributed to the conceptual and theoretical research of the proposed CNN architectures, preprocessing, and general understanding of medical image formats, including aiding in the development of the preprocessing script for the CT scans. Qianyuan Sun developed substantial amounts of the preprocessing script. Xin Ton Wang implemented the proposed CNN architecture, including training and testing on the post-processed PNG images of the cancer nodules. All members contributed to the writing of this paper in equal parts. We hereby state that all the work presented in this report is that of the authors.

X. Open Review Executive Summary

The executive summary can be found at https://openreview.net/forum?id=rJr4kfWCb

References

[1] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[2] Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105)

[3] Sergio Pereira, Adriano Pinto, Victor Alves, and Carlos A Silva. Brain tumor segmentation using convolutional neural networks in mri images. IEEE transactions on medical imaging, 35(5): 12401251, 2016.

[4] Kingma, Diederik P. and Ba, Jimmy. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG], December 2014.

[5] Cancer factsheet, Feb 2017. URL: http://www.who.int/mediacentre/factsheets/fs297/en/.