

The Compact 3D Convolutional Neural Network for Medical Images

Byung Bok Ahn
Stanford University
ntrant@stanford.edu

Abstract

In recent years, medical image processing has developed in the neural network field. There are many trials to enhance the diagnosis performance in the medical area. The grand challenges in biomedical image analysis are famous challenges such as lung nodules, lung cancer, breast cancer, and so on. Convolutional neural network helps radiologists to find cancer quickly. It is important to save the patients' lives. However, it is more time and resource consuming process. In this paper, it is suggested by squeezeNet modification from 2D to 3D, called squeezeNet3D. This compact 3D network helps to improve the speed and accuracy for the medical diagnosis.

1. Introduction

Recently, many approaches have remarkably shown excellent performance for medical diagnosis using convolutional neural networks. However, in the lung area, it is difficult that radiologists find cancer lesions, because normal structures are hardly distinguishable from lesions due to so many lymph nodes, nerves, muscles, and blood vessels. In this research, it will be introduced by the squeezeNet3D for three dimensional structures. It helps to improve more accurate and faster building the models.

The images for medical diagnosis are too big and many 2D slices. CT scans are stored by DICOM(Digital Imaging and Communications in Medicine) files that consists of much description header and raw image pixels. Therefore, DICOM files are not compressed and they need bigger storage size than other image compression files, such as jpeg, png, and so on.

Moreover, CT scans are composed by from 100 to 400 slices for one patients, because slices are usually divided by 2.5mm interval. In comparison medical CT size, region of interest or cancer lesions are very small, it is very sparse matrix. Even though specialists, radiologists and oncologists, they have trouble with finding lesions in the patient's lung CT images. Moreover, lung cancer strikes 225,000 per 1 year in the united states and health care costs over \$12 billion.[8]

3D convolutional networks are more expensive in the computation efficiency. 3D matrix needs more memories

in the computer. 3D convolutional operations needs more calculations than 2D convolutional operations.

In this paper, the input to our algorithm is a 3D matrix composed by stacked CT slices. A CT scan is two dimensional images in one channel different from ordinary images with RGB channels.

We then use a 3D convolutional neural network of which structure is modification of squeezeNet to output a predicted that the patient has cancer or not.

2. Related Work

2.1. Radiologists work

Radiologists have previously examined medical images of chest radiography to detect lung cancer. If they find the mass in the images, they suggest that patient have taken the more tests, such as blood test, CT(Computed Tomography), PET(Positron Emission Tomography), and biopsy. Early detection is the most important in the lung cancer, because lung cancer is also one of the least survivable common cancer with an average 5 year survival rate of less than 20% in the united states. Early detection on average will be at least twice the survival rate.

CT is one of the most powerful method to find cancer lesions, but in the lung, CT scans is quite error-prone, with a particularly high false-negative rate for detecting lesions due to their size, density, locaiton, and conspicuousness.

2.2. Previous work

To find cancer, much research is conducted. There are many lung nodule classification methods by 2D convolutional neural networks. [6]

Petros-Pavlos Ypsilantis, et al. suggested that recurrent convolutional networks be useful for pulmonary nodule detection in CT imaging. They generated a stack of cross-sectional CT scans and used recurrent neural network combined by 2D convolutional neural network. Their networks consists of convolutional neural network, LSTM(Long Short Term Memory) network, and multi layer perceptron with fully connected layers. [2]

Another methods is described by Kui Liu, et al. They suggested multiview convolutional neural networks for lung nodule classification. To find a better way to screen early detection of lung cancer, they suggested not only cross sectional, horizontal CT scans, but also coronal and

sagittal cross sectional images. [3]

Wei Shen, et al., suggested the multi-scale convolutional neural networks for lung nodule classification. Unlike traditional studies, hierarchical scaled convolutional neural network was used. They said that multi-scale nodule patches to learn a set of class-specific features simultaneously by concatenating response neuron activations obtained at the last layer from each input scale. [4]

In the Kaggle competition, named Data Science Bowl 2017, the second winner, Daniel Hammack designed 3D convolutional neural network for lung cancer classification. He said that the size of typical CT scans is about 512x512x400 volumetric pixel. However, the region of interest is generally on the order of 1cm³. Therefore, he used the complex network of 3D convolutional neural network, because he used the small size input voxel (64, 64, 64) with one channel. [8, 13]

In this paper, we reviewed the most important materials in the previous work. Lung cancer in the CT scans is hardly found due to small size and its sparsity. Our technical approach is a 3D convolutional neural network. It helps to find the volumetric mass in the CT scans like recurrent network and multiview convolutional neural network. Bigger input size with small filters may be effective like multi scaled convolutional neural network. [5]

3. Method

In this paper, it is objective to identify the convolutional neural network architecture that have few parameters while remaining efficient accuracy. SqueezeNet architecture suggests that 1x1 filters instead of 3x3 filters be used, because 1x1 filter 9x fewer than 3x3 filter. In the 3D structures, 1x1x1 filters are 27x fewer than 3x3x3 filters. Secondly, the total quantity of paramters, (number of input channel) * (number of filters) * (filter size) is one convolutional layer. For example, if filter is 3x3x3, filter size is 27. In the squeezeNet paper, it decreases the number of input channels to 3x3 filters using squeeze layers. In similarly, 3D convolutional neural network has multichannel, even though medical images is initially one channel. Its channels may be reduced by squeeze layers. [1]

3.1. The Fire module

SqueezeNet has special entity, named fire module. It consists of a squeeze convolution layer, feeding into an expand layer that has a mix of 1x1 and 3x3 convolution filters. Similarly, squeezeNet3D has a mix of 1x1x1 filters and 3x3x3 filters.

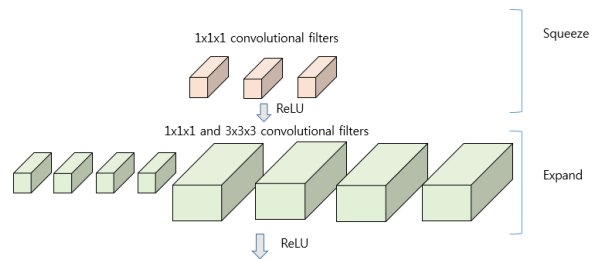


Figure 1. Fire module block

3.2. The squeezeNet3D architecture

The compact 3D convolutional neural network architecture is modification of squeezeNet from 2D to 3D. It is illustrated in figure 2. It consists of 10 convolutional layers from conv3D1 to conv3D10, in the middle 8 fire modules. It allows to use lower computational costs, high performance due to deeper architecture, small memory size.

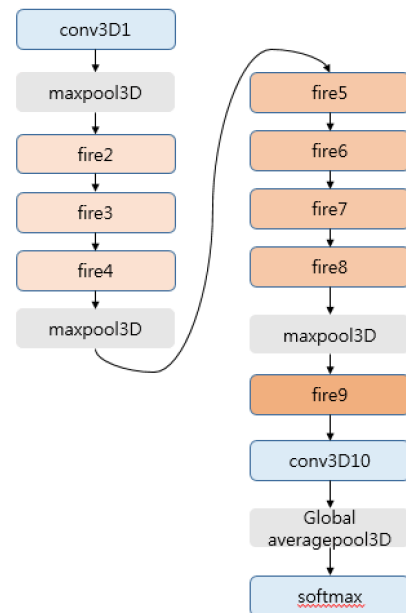


Figure 2. Macroarchitecture view of SqueezeNet3D

Layer (type)	Output Shape	Param #	Connected to
input_3 (InputLayer)	(None, 1, 3, 64, 64)	0	
batchnormalization_3 (BatchNorm)	(None, 1, 3, 64, 64)	256	input_3[0][0]
convolution3d_7 (Convolution3D)	(None, 64, 3, 64, 64)	4864	batchnormalization_3[0][0]
maxpooling3d_7 (MaxPooling3D)	(None, 64, 3, 32, 32)	0	convolution3d_7[0][0]
convolution3d_8 (Convolution3D)	(None, 128, 3, 32, 32)	221312	maxpooling3d_7[0][0]
maxpooling3d_8 (MaxPooling3D)	(None, 128, 3, 16, 16)	0	convolution3d_8[0][0]
convolution3d_9 (Convolution3D)	(None, 256, 3, 16, 16)	884992	maxpooling3d_8[0][0]
maxpooling3d_9 (MaxPooling3D)	(None, 256, 3, 8, 8)	0	convolution3d_9[0][0]
flatten_3 (Flatten)	(None, 49152)	0	maxpooling3d_9[0][0]
dense_7 (Dense)	(None, 1024)	5032672	flatten_3[0][0]
dropout_5 (Dropout)	(None, 1024)	0	dense_7[0][0]
dense_8 (Dense)	(None, 256)	262400	dropout_5[0][0]
dropout_6 (Dropout)	(None, 256)	0	dense_8[0][0]
dense_9 (Dense)	(None, 2)	514	dropout_6[0][0]
Total params: 51,787,810			
Trainable params: 51,706,882			
Non-trainable params: 128			

Figure 3. A general 3D convolutional model for patch 3x64x64 in Keras[10]

Layer (type)	Output Shape	Param #	Connected to
input_4 (InputLayer)	(None, 1, 256, 512, 5, 0)		
conv1 (Conv3D)	(None, 96, 128, 256, 33024)		input_4[0][0]
maxpool1 (MaxPooling3D)	(None, 96, 63, 127, 1, 0)		conv1[0][0]
fire2_squeeze (Conv3D)	(None, 16, 63, 127, 1, 1552)		maxpool1[0][0]
fire2_expand1 (Conv3D)	(None, 64, 63, 127, 1, 1088)		fire2_squeeze[0][0]
fire2_expand2 (Conv3D)	(None, 64, 63, 127, 1, 27712)		fire2_squeeze[0][0]
concatenate_25 (Concatenate)	(None, 128, 63, 127, 0)		fire2_expand1[0][0] fire2_expand2[0][0]
fire3_squeeze (Conv3D)	(None, 16, 63, 127, 1, 2064)		concatenate_25[0][0]
fire3_expand1 (Conv3D)	(None, 64, 63, 127, 1, 1088)		fire3_squeeze[0][0]
fire3_expand2 (Conv3D)	(None, 64, 63, 127, 1, 27712)		fire3_squeeze[0][0]
concatenate_26 (Concatenate)	(None, 128, 63, 127, 0)		fire3_expand1[0][0] fire3_expand2[0][0]
fire4_squeeze (Conv3D)	(None, 32, 63, 127, 1, 4128)		concatenate_26[0][0]
fire4_expand1 (Conv3D)	(None, 128, 63, 127, 4224)		fire4_squeeze[0][0]
fire4_expand2 (Conv3D)	(None, 128, 63, 127, 110720)		fire4_squeeze[0][0]
concatenate_27 (Concatenate)	(None, 256, 63, 127, 0)		fire4_expand1[0][0] fire4_expand2[0][0]
maxpool4 (MaxPooling3D)	(None, 256, 31, 63, 6, 0)		concatenate_27[0][0]
fire5_squeeze (Conv3D)	(None, 32, 31, 63, 63, 8224)		maxpool4[0][0]
fire5_expand1 (Conv3D)	(None, 128, 31, 63, 6, 4224)		fire5_squeeze[0][0]
fire5_expand2 (Conv3D)	(None, 128, 31, 63, 6, 110720)		fire5_squeeze[0][0]
concatenate_28 (Concatenate)	(None, 256, 31, 63, 6, 0)		fire5_expand1[0][0] fire5_expand2[0][0]
fire6_squeeze (Conv3D)	(None, 48, 31, 63, 63, 12336)		concatenate_28[0][0]
fire6_expand1 (Conv3D)	(None, 192, 31, 63, 6, 9408)		fire6_squeeze[0][0]
concatenate_29 (Concatenate)	(None, 384, 31, 63, 6, 0)		fire6_expand1[0][0] fire6_expand2[0][0]
fire7_squeeze (Conv3D)	(None, 48, 31, 63, 63, 18480)		concatenate_29[0][0]
fire7_expand1 (Conv3D)	(None, 192, 31, 63, 6, 9408)		fire7_squeeze[0][0]
fire7_expand2 (Conv3D)	(None, 192, 31, 63, 6, 249024)		fire7_squeeze[0][0]
concatenate_30 (Concatenate)	(None, 384, 31, 63, 6, 0)		fire7_expand1[0][0] fire7_expand2[0][0]
fire8_squeeze (Conv3D)	(None, 64, 31, 63, 63, 24640)		concatenate_30[0][0]
fire8_expand1 (Conv3D)	(None, 256, 31, 63, 6, 16640)		fire8_squeeze[0][0]
fire8_expand2 (Conv3D)	(None, 256, 31, 63, 6, 442624)		fire8_squeeze[0][0]
concatenate_31 (Concatenate)	(None, 512, 31, 63, 6, 0)		fire8_expand1[0][0] fire8_expand2[0][0]
maxpool8 (MaxPooling3D)	(None, 512, 15, 31, 3, 0)		concatenate_31[0][0]
fire9_squeeze (Conv3D)	(None, 64, 15, 31, 31, 32832)		maxpool8[0][0]
fire9_expand1 (Conv3D)	(None, 256, 15, 31, 3, 16640)		fire9_squeeze[0][0]
fire9_expand2 (Conv3D)	(None, 256, 15, 31, 3, 442624)		fire9_squeeze[0][0]
concatenate_32 (Concatenate)	(None, 512, 15, 31, 3, 0)		fire9_expand1[0][0] fire9_expand2[0][0]
fire9_dropout (Dropout)	(None, 512, 15, 31, 3, 0)		concatenate_32[0][0]
conv10 (Conv3D)	(None, 3, 1, 1, 1, 1) 1539		fire9_dropout[0][0]
avgpool10 (AveragePooling3D)	(None, 3, 1, 1, 1, 1) 0		conv10[0][0]
flatten (Flatten)	(None, 3)		avgpool10[0][0]
softmax (Activation)	(None, 3)		flatten[0][0]

Total params: 1,861,699
Trainable params: 1,861,699
Non-trainable params: 0

Figure 4. The squeezeNet3D model in Keras

3.3. Other architecture details

In the squeezeNet paper, squeezeNet is detailed by ReLU[15], NiN[14], and so on. It is focused on NiN that is the lack of fully-connected layers. This design is more powerful than other models, because fully-connected layers has much consumption of memory resources. Also, global average pooling method is useful. It enables to show the improved performance, because max pooling process on the layer has features of activation maps and fully-connected layers are prone to overfitting. However, global average pooling is much more deterministic. For conventional convolutional neural network classification, the feature maps of the last convolutional layers are vectorized and fed into fully-connected layers. This structure is merged by the convolutional structure and

traditional deep neural network classifiers.

SqueezeNet and our squeezeNet3D overcome the problem from the fully connected layers. [16]

3.4. Loss function

The end layer of squeezeNet is softmax layer. Similarly, squeezeNet3D has softmax layer. However, the output is binary classification, cancer or not. The loss function is binary crossentropy as follows:

$$Loss(p, q) = -\frac{1}{N} \sum_{i=1}^N p_i \log q_i = -\frac{1}{N} \sum_{i=1}^N y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_i)$$

p : true probability, q : predictive distribution

y : true label, \hat{y} : predictive label

3.5. Pseudo-labeling[11]

DH. Lee suggested that pseudo-labeling of test or validation datasets be useful. Diminishing false positive is one of the most interest regions in medical approaches. In this paper, pseudo-labeling is efficient for reduction of false positive, because this model learns complicated validation or test domain. It was something useful.

4. Dataset and Features

Lung CT scans are provided by the National Cancer Institutes. Stage1 (~240GB) from Kaggle competition (Data Science Bowl 2017). It consists of 362 cancer patients and 1035 no cancer candidates.

Data	Patients
Cancer	362
No Cancer	1035

Table 1. Data Science Bowl 2017 dataset[9]

Luna2016 datasets are used for evaluation datasets for nodule in the lung CT. They are annotated by radiologists, size and malignancy. Small designed 3D convolutional neural network outperforms 2D convolutional neural network. It is convinced by 3D convolutional neural network. [7, 12]

Data	Patients
Nodule	281
No Nodule	1340

Table 2. Luna 2016 challenge dataset

It is observed that traditional 3D convolutional neural network shows the 90% more validation accuracy over

5~8% of 2D convolutional neural network.

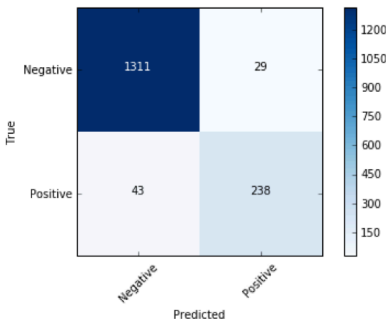


Figure 4. The confusion matrix of Luna2016 with 3D convolutional neural network

5. Experiments

In this paper, the input is the candidate 3D matrix of CT slices, and the output was the predicted class label. Environment is the local system of Keras 2.0, python 2.7, numpy 1.11, Nvidia gtx 1080, Ubuntu 16.04 LTS.

5.1. Preprocessing

As a pre-processing step for all the experiments, each CT scan is resized so that each voxel represents a 1mm³ volume. This is necessary so that the same model can be applied to scans with different 'slice thickness'. It refers to the distance between consecutive slices and can vary by up to four times between scans. The scans are also normalized between -1000 and 400 Hounsfield units. -1000 HU is air and 400 HU is dense bone. HU values in this range are informative for medical diagnosis. Pixel array of the scans in DICOM is converted to integer type and Hounsfield units and resized by 3D voxels which are (128, 128, 128) with padding. It enables to use the huge input data, eight times bigger than Daniel Hammack's. There is no scanning the matrix with the voxels.

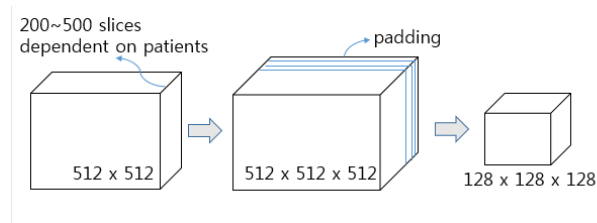


Figure 4. Convert the CT scans

5.2. Hyperparameter

In terms of batch size, we used a relatively small size

because GPU has only 8GB memory. It has no trouble when batch size is 4 or less. Dropout rate is just 0.9.

Learning rate is one of the most important for 3D convolutional neural networks. It is used by Adam(Adaptive momentum) and roughly 1e⁻⁵. The learning rate is 1e⁻⁶ during fine tuning process. Learning rate and momentum methods are found by many random trials.

5.3. Evaluation Metrics

In this paper, primary metric is accuracy for model evaluation. It is used by general modeling procedure. However, in medical fields, sensitivity and specificity is more important. Sensitivity, recall, also called true positive rate(TPR) shows the reliability of the tests. It measures the proportion of positives that are correctly identified. Another is specificity. It is called the true negative rate(TNR) that measures the proportion of negatives that are correctly identified. In oncology, type 2 error, false negative must be removed. Sensitivity is more important in the medical field.

- Sensitivity = TP / (TP + FN)
- Specificity = TN / (TN + FP)
- Accuracy = (TP + TN) / (TP + FP + FN + TN)

In this paper, we investigate the results with confusion matrix.

5.4. Results

The accuracy of the model is over 0.99 after 250 epochs. K-fold cross validation are used. SqueezeNet3D model converges faster in the training set and the accuracy of validation set nearly approaches over 0.99 in periodically.

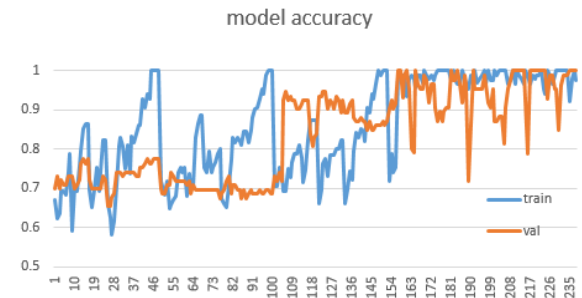


Figure 5. SqueezeNet3D model accuracy for training and validation set of lung cancer data

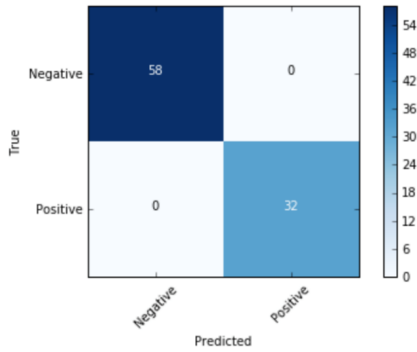


Figure 6. the confusion matrix of validation set

In the validation sets, sensitivity is 1.0 and specificity is 1.0. Accuracy is just 1.0. It is perfect score. SqueezeNet3D has features to explain the cancer classifier. However, It shows the risk of overfitting training data and validation data.

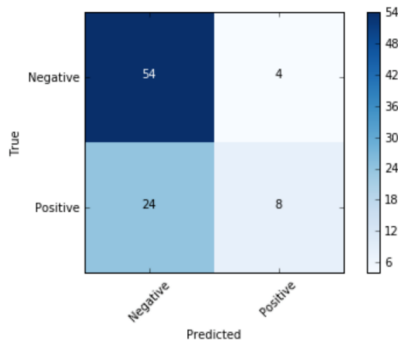


Figure 7. the confusion matrix of test set

However, in the test set, randomly about 8% partitioned dataset, sensitivity is 0.25 and specificity is 0.93. It remains the unsatisfied result. Maybe various cancer domain, such as adenocarcinoma, squamous cell carcinoma needs to be classified in the test set. In the general modeling process, overfitting may be avoided by regularization. However, regularization still has been studied in 3D convolutional neural network, because 3D matrix is more complex than 2D matrix or images.

6. Conclusion

Squeezenet3D is more powerful than other contemporary 3D CNN, because it reduces memory size and improves the speed. It consists of layers of seven depth that sufficiently converges the global minimum. It

minimizes the training loss faster. Also, it has small sizes that has 1,861,186 parameters and needs only 8.9 MB. In this paper, it shows four times faster than other ordinary 3D CNN with GPU, Nvidia gtx 1080 in Ubuntu 16.04 LTS.

Even though it is the state of the art result for validation with squeezeNet3D, test scores are not satisfied. Its test scores are about 70% accuracy. It needs more regularization methods, weight regularizer or data augmentation methods to improve the test scores.

It is planning to test the data preprocessing of variable size and implementation of squeezeNet3D architecture not dependent on variable stacked sizes of CT. Finally, it will be tested with 3D data augmentation for unbalanced patient data.

References

1. Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, Kurt Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. arXiv:1602.07360, 2016
2. Petros-Pavlos Ypsilantis, Giovanni Montana. Recurrent Convolutional Networks for Pulmonary Nodule Detection in CT Imaging. arXiv:1609.09143, 2016
3. Kui Liu, Guixia Kang. Multiview Convolutional Neural Networks for Lung Nodule Classification. International Journal of Imaging Systems and Technology, Volume 27, Issue 1, 2017: pages 12-22
4. Wei ShenMu ZhouFeng Yang. Multi-scale Convolutional Neural Networks for Lung Nodule Classification. IPMI 2015: Information Processing in Medical Imaging pp 588-599
5. Sarfaraz Hussein, Robert Gillies, Kunlin Cao, Qi Song, Ulas Bagci. TUMORNET: Lung Nodule Characterization using Multi-view Convolutional Neural Network with Gaussian Processor arXiv:1703.00645, 2017
6. Sheila Ramaswamy, Karen Truong, Pulmonary Nodule Classification with Convolutional Neural Networks, http://cs231n.stanford.edu/reports/2016/pdfs/324_Report.pdf
7. Luna 16 grand challenge. <https://luna16.grand-challenge.org/>.
8. Daniel Hammack. 2nd place solution to 2017 ndsb. <https://dhammack.github.io/kaggle-ndsb2017/>.
9. Kaggle. Data science bowl 2017. <https://www.kaggle.com/c/data-science-bowl-2017>.
10. Francois Chollet. Keras. <https://github.com/fchollet/keras>, 2015.
11. Dong-Hyun Lee. Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks. <http://kaggle.com>
12. Samuel G Armato, Geo rey McLennan, Luc Bidaut, Michael F McNi -Gray, Charles R Meyer, Anthony P Reeves, Binsheng Zhao, Denise R Aberle, Claudia I Henschke, Eric A Ho man, and others. 2011. elungimagedatabaseconsortium (LIDC) and image database resource initiative (IDRI): a completed reference database of lung nodules on CT scans. Medical physics 38, 2 (2011), 915–931.

13. Kingsley Kuan, Mathieu Ravaut, Gaurav Manek, Huiling Chen, Jie Lin, Babar Nazir, Cen Chen, Tse Chiang Howe, Zeng Zeng, Vijay Chandrasekhar. Deep Learning for Lung Cancer Detection: Tackling the Kaggle Data Science Bowl 2017 Challenge, arXiv:1705.09435
14. Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. arXiv:1312.4400, 2013.
15. Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In ICML, 2010.
16. Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark A Horowitz, and William J Dally. Eie: Efficient inference engine on compressed deep neural network. International Symposium on Computer Architecture (ISCA), 2016a.