Tumor Identification in Colorectal Histology Images Using a Convolutional Neural Network

Hongjun Yoon^{1,2} · Joohyung Lee¹ · Ji Eun Oh¹ · Hong Rae Kim¹ · Seonhye Lee¹ · Hee Jin Chang³ · Dae Kyung Sohn^{1,3}

Published online: 31 July 2018
© Society for Imaging Informatics in Medicine 2018

Abstract

Colorectal cancer (CRC) is a major global health concern. Its early diagnosis is extremely important, as it determines treatment options and strongly influences the length of survival. Histologic diagnosis can be made by pathologists based on images of tissues obtained from a colonoscopic biopsy. Convolutional neural networks (CNNs)—i.e., deep neural networks (DNNs) specifically adapted to image data—have been employed to effectively classify or locate tumors in many types of cancer. Colorectal histology images of 28 normal and 29 tumor samples were obtained from the National Cancer Center, South Korea, and cropped into 6806 normal and 3474 tumor images. We developed five modifications of the system from the Visual Geometry Group (VGG), the winning entry in the classification task in the 2014 ImageNet Large Scale Visual Recognition Competition (ILSVRC) and examined them in two experiments. In the first experiment, we determined the best modified VGG configuration for our partial dataset, resulting in accuracies of 82.50%, 87.50%, 87.50%, 91.40%, and 94.30%, respectively. In the second experiment, the best modified VGG configuration was applied to evaluate the performance of the CNN model. Subsequently, using the entire dataset on the modified VGG-E configuration, the highest results for accuracy, loss, sensitivity, and specificity, respectively, were 93.48%, 0.4385, 95.10%, and 92.76%, which equates to correctly classifying 294 normal images out of 309 and 667 tumor images out of 719.

Keywords Colonoscopic biopsy · Convolutional neural network · Histology image · Visual geometry group

Introduction

According to GLOBOCAN data from the International Agency of Research on Cancer-World Health Organization (IARC-WHO), colorectal cancer (CRC) is a major global health concern [1]. Although recent mortality rates of CRC have stabilized or are even declining, it is still the third most common type of cancer in Korea [2] and the fourth most common in the USA [3]. In Korea, the age-adjusted incidences of CRC for men and women were 40.2 and 22.2 per 100,000 in 2015, respectively. The United

States National Cancer Institute estimates that 140,250 people will be diagnosed with CRC in 2018, representing 8.1% of all new cases of cancer in the USA. Further, it estimates that 50,630 people will die because of CRC in 2018. Early diagnosis of CRC is extremely important, as it determines treatment options and strongly influences the length of survival [4]. One way to diagnose CRC is an endoscopic biopsy, in which a small amount of tissue is removed during a colonoscopy. Other tests may be able to determine whether a tumor is present, but examining a histology image obtained from a biopsy is the only way to make a definitive diagnosis of CRC.

Many recent studies have suggested using machine learning techniques to classify, localize, and segment tumor areas in histology images [5]. Deep neural networks (DNNs) are used extensively to extract and learn features of subjects [6]; DNNs specifically adapted to image data, called convolutional neural networks (CNNs), can effectively classify or locate tumors [7]; however, comparatively few studies have examined CRC using these techniques.

Sirinukunwattana et al. proposed a deep learning approach to classifying routine stained histology images of colorectal

✉ Dae Kyung Sohn
gsgsbal@ncc.re.kr

¹ Innovative Medical Engineering and Technology Branch, Research Institute and Hospital, National Cancer Center, Goyang, Gyeonggi, South Korea

² College of Engineering and Computer Science, Syracuse University, Syracuse, NY, USA

³ Center for Colorectal Cancer, Research Institute and Hospital, National Cancer Center, 323 Ilsan-ro, Ilsandong-gu, Goyang-si, Gyeonggi-do 10408, Republic of Korea

adenocarcinomas into four classes: epithelial, inflammatory, fibroblast, and miscellaneous [8]. They used sophisticated methods to combine color and texture information from each image into superpixels. Despite being a challenging multiclass classification task, this approach achieved a multiclass AUC (area under the receiver operating curve) of 91.7%.

The ImageNet Large Scale Visual Recognition Competition (ILSVRC) invites research teams to compete on several visual recognition tasks [9]. The winner of the 2014 ILSVRC classification and localization task, the Visual Geometry Group (VGG), showed that the visual classification and localization accuracy of a CNN increases with the number of layers it contains [10]. Anthimopoulos et al. later used a modified VGG architecture to classify lung diseases [11], improving its performance by shrinking the kernel size, replacing the activation function, and adding an optimizer from the VGG. “Implementation Details” section explains details of the VGG and the modification we employ in this study to predict whether colorectal histology images contain tumors.

Dataset

Colorectal histology images were obtained from the Center for Colorectal Cancer, National Cancer Center, Korea. All protocols were conducted according to the principles of the Declaration of Helsinki and were approved by the institutional review board of our institution (NCC2016-0048). All patients provided written informed consent to participate. Thirty patients who were diagnosed with CRC, with ages ranging from 20 to 80 years, participated in the study; from these participants, we obtained 30 normal and 30 tumor images. However, two normal and one tumor images were damaged; hence, 28

normal and 29 tumor images were used in this study. The average size of these images was 12,762 pixels by 13,939 pixels. Feeding these original images to the experimental computer’s graphics processing unit (GPU) caused memory issues; the largest images the GPU could handle were 256 pixels by 256 pixels, so the raw images were cropped to this size, as shown in Fig. 1. Previous studies have used larger images to use CNN methods similar to ours, but since our CNN model is deeper—i.e., has more layers—than those methods, the GPU required more memory capacity [11]. Each coordinate for the regions of interest was manually localized by pathologists in the Center for Colorectal Cancer in National Cancer Center. Each original image was fed into a program written in Python that read the coordinate points and cropped the images based on the recorded data. All images were extracted so that the localized area was located exactly in the center of the cropped images. The normal patch sets did not overlap any part of the cancer region. Thus, we obtained 10,280 smaller images, consisting of 6806 normal and 3474 tumor images. The two groups of images were shuffled separately to randomize them and 680 normal and 348 tumor images were taken as validation images. Consequently, the whole 10,280-image dataset was divided into training and validation sets with 9252 and 1028 images. Compared to other recent studies that have proposed methods for classifying tumor areas, we have a relatively small dataset—the original consists of histology images from only 30 patients. However, determining whether an image contains a tumor relies on examining the cells, meaning that the small cropped images serve the purpose of creating a large dataset, as each of the 10,280 images has at least one cell in it [7].

In the first experiment, VGG configurations were evaluated to determine the most suitable for identifying tumors from

Fig. 1 Original tumor histology image (left) with tumor area marked with green boundaries and the resulting fixed-size images (right)

histology images. In this experiment, we randomly chose 400 images—200 tumor and 200 normal images—from the 10,280 images. Of the chosen images, 360 were used as training images and 40 as validation images. In the second experiment, all 10,280 images were used to evaluate the performance of the chosen configuration and to find the best VGG model by adjusting its weight values during training.

VGG

With advancements in automated image processing research, many research groups have started to participate in the ILSVRC—an annual software contest that evaluates the ability of automated machine learning programs to correctly classify and detect objects and scenes from a dataset.

In 2014, VGG won the classification and localization competition, based on their hypothesis that the depth of a CNN largely affects classification and detection accuracy. In the classification and detection program report, the authors proposed six configurations [10, 12].

As presented in Table 1, VGG fixed the size of the input to 224×224 RGB images. To compare the performance of these configuration, which only varied by the depth of the ConvNet, all configurations were designed using the same principles. Except for configurations A and A-LRN, the configurations are updated by adding another convolutional layer—the main building block of CNNs—to capture all the information in the input using a filter that slides across the input image, or by increasing the size of the filter or the number of channels [13].

Except for three convolutional layers in configuration C, all the convolutional layers have a receptive field size, i.e., a 3×3 filter, which is the smallest size that can capture the

Table 1 ConvNet configurations. The depth of the configurations increases from the left (A) to right (E), as more layers are added (the added layers are shown in italics). The convolutional layer parameters

are denoted as “conv <receptive field size> – <number of channels>.” For brevity, the ReLU activation function is not shown

ConvNet configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
Input (224×224 RGB image)					
conv 3–64	conv 3–64	conv 3–64	conv 3–64	conv 3–64	conv 3–64
	<i>LRN</i>	<i>conv 3–64</i>	conv 3–64	conv 3–64	conv 3–64
maxpool					
conv 3–128	conv 3–128	conv 3–128	conv 3–128	conv 3–128	conv 3–128
		conv 3–128	conv 3–128	conv 3–128	conv 3–128
maxpool					
conv 3–256	conv 3–256	conv 3–256	conv 3–256	conv 3–256	conv 3–256
conv 3–256	conv 3–256	conv 3–256	conv 3–256	conv 3–256	conv 3–256
			<i>conv 1–256</i>	<i>conv 3–256</i>	conv 3–256
					<i>conv 3–256</i>
maxpool					
conv 3–512	conv 3–512	conv 3–512	conv 3–512	conv 3–512	conv 3–512
conv 3–512	conv 3–512	conv 3–512	conv 3–512	conv 3–512	conv 3–512
			<i>conv 1–512</i>	<i>conv 3–512</i>	conv 3–512
					<i>conv 3–512</i>
maxpool					
conv 3–512	conv 3–512	conv 3–512	conv 3–512	conv 3–512	conv 3–512
conv 3–512	conv 3–512	conv 3–512	conv 3–512	conv 3–512	conv 3–512
			<i>conv 1–512</i>	<i>conv 3–512</i>	conv 3–512
					<i>conv 3–512</i>
maxpool					
FC-4096					
FC-4096					
FC-4096					
soft-max					

Each new component from the previous (left) model configuration is *italicized*

Fig. 2 The classification workflow. First, the coordinates of the region of interest are extracted from the original images. Using those images and the extracted coordinates, images are cropped to 256×256 pixels. Images are converted to numpy format for efficiency, shuffled, and separated

to training and validation sets at a ratio of 9:1. Finally, the training set is fed to the modified VGG model for training; after each epoch, the training and validation sets' prediction evaluated

notion of left/right and up/down directions as well as a center. The smallest filter gives the same effect as relatively larger filters by using multiple convolutional layers consecutively. The real advantage of this approach is in breaking up the larger filter into smaller filters and adding activation functions to make clearer decisions and decrease the size of parameters, ultimately reducing the size and memory of the process. Activation functions in the activation layer decide the result of each pixel point in the filter based on whether the input value is higher or lower than a threshold. In VGG, a rectified linear unit (ReLU) activation function is used for all activation layers [13].

The pooling layers, which are for dimensionality reduction, allowing the CNN to focus on the most relevant patterns, were fixed to a maximum pooling layer with a size of 2×2 pixels, a stride of 2, and no padding on any side. The last section has three fully connected layers, to determine the class meaning of the output of the previous layers; finally, a soft-max function is utilized to make the final decision [9, 13]. A summary and a visualization of the workflow from “VGG” and “Implementation Details” sections are shown in Fig. 2.

Implementation Details

Our implementation of the main training and testing was with accomplished with Python packages TensorFlow and Numpy [14, 15]. During preprocessing, we obtained the 256×256 images using Matplotlib. We utilized four NVIDIA GTX 1080 Ti GPUs to speed up the learning process.

The main structure of the VGG configuration adapted for this project is shown in Table 2 and integrates several differences into the modified VGG configuration. First, the input image size was fixed to 256×256 RGB instead of 224×224 RGB. The original VGG had six structures: A, A-LRN (local response normalization), B, C, D, and E. However, in the modified VGG ConvNet, A-LRN was removed from the list because LRN, which is used to highlight more significant points and suppress insignificant points in the extracted data from the convolutional layer, was considered outdated [16]. Instead, batch normalization was added before each ReLU activation layer, with a decay rate of 0.99 [9].

Several details were added to the last section of the ConvNet. In the original VGG ConvNet configurations, there

Table 2 Modified VGG ConvNet configurations. Only five of the structures are included. The details of each configurations follow the VGG model. Batch normalization and ReLU activation function are not shown for brevity

Modified VGG model structure				
A	B	C	D	E
11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
Input (256 × 256 RGB image)				
conv 3–64	conv 3–64	conv 3–64	conv 3–64	conv 3–64
	conv 3–64	conv 3–64	conv 3–64	conv 3–64
maxpool				
conv 3–128	conv 3–128	conv 3–128	conv 3–128	conv 3–128
	conv 3–128	conv 3–128	conv 3–128	conv 3–128
maxpool				
conv 3–256	conv 3–256	conv 3–256	conv 3–256	conv 3–256
conv 3–256	conv 3–256	conv 3–256	conv 3–256	conv 3–256
		conv 1–256	conv 3–256	conv 3–256
				conv 3–256
maxpool				
conv 3–512	conv 3–512	conv 3–512	conv 3–512	conv 3–512
conv 3–512	conv 3–512	conv 3–512	conv 3–512	conv 3–512
		conv 1–512	conv 3–512	conv 3–512
				conv 3–512
maxpool				
conv 3–512	conv 3–512	conv 3–512	conv 3–512	conv 3–512
conv 3–512	conv 3–512	conv 3–512	conv 3–512	conv 3–512
		conv 1–512	conv 3–512	conv 3–512
				conv 3–512
maxpool				
FC-4096				
soft-max				

were three fully connected layers in the last section, but there is only one fully connected layer in the modified VGG configuration. VGG was originally designed to complete the ILSVRC task, which required classifying more classes than simply distinguishing tumor and normal images. Reducing the number of fully connected layers resulted in the task being completed more quickly and efficiently [17]. In addition, 10% of the neurons were randomly removed using dropout after the fully connected layer. These changes enabled the network to make more robust decisions.

The learning rate started at 0.001 and slowly decayed exponentially, with 4% of the original learning rate decaying after 1000 steps. Stochastic gradient descent (SGD) was used as the core of the CNN. It samples a batch of data, forward propagates it through the graph, determines the loss, back propagates it to calculate gradients, and updates the parameters using these gradients. We used the Adam optimizer for calculating the SGD in our configuration. The batch size for learning was set to 10, which means that 10 images were fed to the CNN in each iteration. In addition, the entire dataset, 10,280 images, was processed 50 times—i.e., 50 epochs.

We compared each ground truth image label—“0” for normal and “1” for tumor—to the estimation from the CNN. Based on this result, each image result was categorized as being a true positive (TP), false positive (FP), false negative (FN), or true negative (TN). The performance metrics used in our experiments were accuracy, loss, sensitivity, and specificity, which were computed using the following equations:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}, \quad \text{Sensitivity} = \frac{TP}{TP+FN},$$

$$\text{Specificity} = \frac{TN}{TN+FP}.$$

Experiments and Results

We conducted two experiments: In the first, we determined the most accurate and best suited of the five CNN structures for tumor identification using 400 images—200 normal and 200 tumor images. Both normal and tumor images were chosen randomly from the dataset, and the same subset was used for all five structures. In the second experiment, the accuracy, sensitivity, specificity, and execution time of the best

performing CNN structure were determined for the entire dataset.

Configuration Comparison

The modified VGG E model structure provided the best accuracy and loss results. The accuracy and loss curves for the structure were the most stable of the five models, and it had the highest accuracy rate (94.29%) and lowest loss value (0.365) (Fig. 3).

As described by Simonyan and Zisserman, the deeper a CNN gets, the better it performs. We demonstrated that the VGG method is well suited to tumor classification [9, 10].

Best Performing CNN Configuration

In this experiment, we used all 10,280 dataset images, comprising a mix of 6806 normal images and 3474 tumor images. We randomly combined approximately 10% of each image type (680 normal and 348 tumor images) into a validation set. The rest were used as a training set. We trained the

Fig. 3 **a** Accuracy curve for modified VGG ConvNet configurations. **b** Loss curve for the modified VGG ConvNet configurations

Fig. 4 Modified VGG E model graph with feature maps of the model's interpretation of the input image after each max pooling layer

VGG E structure on the training set and tested it with both the training and validation sets, which were evaluated separately (Fig. 4).

When the CNN model performed better for an epoch based on the chosen performance metrics, its model weights were saved, to extract the best model for the specific task.

Table 3 Performance measurement evaluation for the highest-performance epochs (13 and 14)

Epoch	TP	FN	TN	FP	Accuracy (%)	Loss	Sensitivity (%)	Specificity (%)
13	305	13	645	65	92.41	0.4382	95.91	90.84
14	294	15	667	52	93.48	0.4385	95.10	92.76
50	294	21	641	72	90.96	0.4464	93.33	89.90

Table 4 Statistics for the sample-based performance in Fig. 5

	Normal accuracy (sensitivity, %)	Tumor accuracy (specificity, %)
Minimum	86.71	75.00
Q1	93.44	88.77
Median	95.10	92.44
Mean	94.33	91.87
Q3	95.92	96.35
Maximum	97.45	100.00
Range	10.74	25.00
Standard deviation	2.45	5.99

Based on the performance measurement graph in Fig. 5, Epochs 13 and 14 showed the best performance of the 50 epochs. In the 13th epoch, accuracy, loss, sensitivity, and specificity were 92.41%, 0.4382, 95.91%, and 90.84%, respectively. In the 14th epoch, accuracy, loss, sensitivity, and specificity were 93.48%, 0.4385, 95.10%, and 92.76%, respectively. In the 50th epoch, accuracy, loss, sensitivity, and specificity were 90.96%, 0.4464, 93.33%, and 89.90%, respectively (Table 3). Table 4 shows the sample-based performance statistics for the original 28 normal and 29 tumor images, which is also visually represented in Fig. 6 as a box plot.

The running time for the experiment using the VGG E configuration with the entire dataset over 50 epochs was 10 days, 7 h, 14 min, and 20 s. To examine the performance more thoroughly, we evaluated the performance at each

epoch, which reduced the runtime; the runtime included training time, the time taken to test on the training and validation sets, and performance measuring time (evaluating the accuracy, loss, sensitivity, and specificity). Training took 3 days, 11 h, 37 min, and 29 s for 50 epochs, which means that each image was learned in less than 0.65 s. Checking the performance of each image took 1.11 s. The time taken for training and a performance evaluation of the total runtime, each epoch, and each image are shown in Table 5.

Discussion

Many artificial intelligence-based methods have been proposed for differentiating diagnoses, response prediction, and prognosis prediction in medicine. However, several challenges still limit the application of artificial intelligence to pathological imaging diagnosis. Among these challenges is that pathological images have a relatively large data volume, because of the high magnification of medical images used for diagnosis; these images may vary widely in color and tone depending on the staining method used for microscopic diagnosis. Consequently, standardizing the data for analysis is difficult. To overcome these challenges, this study divided the image slides into images containing tumors and images containing no tumors. The method proposed in this study may be useful in other pathological diagnostic studies using small datasets. However, to work with large datasets, it will be necessary to automate many manual and time-consuming tasks.

Fig. 5 **a** Accuracy, sensitivity, and specificity for the modified VGG E ConvNet configuration. **b** Loss curve for the modified VGG E ConvNet configuration

Fig. 6 Box plot representation of sample-based performance for normal (sensitivity) and tumor (specificity) images. Boxes represent the 25th–75th percentile region, with the median in the middle of the gray and black sections. Upper whisker represents from the 75th percentile to the maximum and lower whisker represents from the minimum to the 25th percentile of the data

As stated in “Introduction” section, various automated programs have been proposed for detecting tumors using DNNs in general and CNNs in particular. Further, studies have classified and located tumors in various areas, including the brain [18]. Moreover, recently, many research groups have begun using CNNs to identify CRC.

Despite the excellent results obtained in this study, there is room for improvement because we restricted some variables for ease of computation. First, we divided the histology image into smaller 256 pixels by 256 pixels images during preprocessing because of the inability of the GPU’s maximum capacity to handle larger images. However, the sliding-window technique, which enabled large images to be processed by breaking them into smaller images and efficiently concatenating all the features, could be used instead. Further, as ILSVRC is held every year,

there will be more efficient and better CNN configurations that could be implemented for CRC identification. Finally, this program only indicated whether an image contains tumors. Because each individual image has a predicted value from 0 to 1, we could average them and give a threshold to determine whether the original image can be classified as a normal or tumor image. Additionally, it would be significantly more helpful to implement it with localization, such as box-regression or segmentation, using a CNN [19]. This study was a first step toward demonstrating not only that these methods will be useful in localization tasks, but also for practical usage as in CRC classification.

Conclusion

In this paper, we proposed a method for tumor classification from colorectal histology images using a CNN: a ConvNet based on VGG, the winner of the 2014 ILSVRC classification and localization task. Because some of the components of the VGG model proposed in 2014 were outdated, we replaced them with newer tools [12].

In our first experiment, we used 400 images (200 normal and 200 tumor images) to determine the best VGG configuration for classification of colorectal histology images and verified that as the depth of the model increased, its performance improved steadily. As the accuracies for VGG configurations A–E were 82.5%, 87.5%, 87.5%, 91.4%, and 94.3%, respectively, we conducted the second experiment using VGG E, which had an accuracy of 94.3%, on the 10,280-image dataset. The best accuracy and specificity (93.48% and 92.76%, respectively) were exhibited in the 14th epoch; the best loss and sensitivity (0.4382 and 95.91%, respectively) were exhibited in the 13th epoch. This experiment, comprising 50 epochs, took 10 days, 7 h, 14 min, and 20 s, which is equivalent to taking 0.65 s to learn an image and 1.1 s to classify an image.

Table 5 The time taken for training and performance evaluation for the total runtime, each epoch, and each image

Total Execution	
Training Time	3 days 11 hours 37 minutes 29 seconds
Training Performance Time	5 days 22 minutes 53 minutes 23 seconds
Validation Performance Time	15 hours 52 minutes 35 seconds
<i>Total Execution Time</i>	<i>10 days 7 hours 14 minutes 20 seconds</i>
Single Epoch	
Training Time	1 hour 40 minutes 20 seconds
Training Performance Time	2 hours 51 minutes 28 seconds
Validation Performance Time	19 minutes 3 seconds
<i>Single Epoch Time</i>	<i>4 hours 56 minutes 41 seconds</i>
Single Image	
Training Time	0.65 second
Training Performance Time	1.11 second
Validation Performance Time	1.11 second
<i>Single Image Time</i>	<i>1.73 seconds</i>

Each italicized row is the sum of the above time spent for the total execution and single epoch time. Single image time is the sum of training time and either the training or validation performance time

Funding This work was supported by the National Cancer Center (grant numbers NCC-1710070 and NCC-1511670) and the Chungcheongbuk-do Value Creation Program (through the Osong Medical Innovation Foundation of Korea, funded by the Chungcheongbuk-do). The funding sources had no role in the study design; in the collection, analysis, and interpretation of data; in the writing of the manuscript; or in the decision to submit the manuscript for publication.

Compliance with Ethical Standards

The study was conducted according to the principles of the Declaration of Helsinki and was approved by the institutional review board of our institution (NCC2016-0048). All patients provided written informed consent to participate.

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. Ferlay J, Soerjomataram I, Dikshit R, Eser S, Mathers C, Rebelo M, Parkin DM, Forman D, Bray F: Cancer incidence and mortality worldwide: Sources, methods and major patterns in GLOBOCAN 2012. *Int J Cancer* 136(5):E359–E386, 2015
2. Jung KW, Wong YJ, Kong HJ, Lee ES: Community of Population-Based Regional Cancer Registries: Cancer statistics in Korea: Incidence, mortality, survival, and prevalence in 2015. *Cancer Res Treat* 50:303–316, 2018
3. National Cancer Institute: Surveillance, Epidemiology, and End Results Program, Cancer Stat Facts: Colon and Rectum Cancer. Maryland, USA <https://seer.cancer.gov/statfacts/html/colorect.html>
4. Torre LA, Bray F, Siegel RL, Ferlay J, Lortet-Tieulent J, Jemal A: Global cancer statistics, 2012. *CA Cancer J Clin* 65(2):87–108, 2015. <https://doi.org/10.3322/caac.21262>
5. Teramoto A, Tsukamoto T, Kiriya Y, Fujita H: Automated classification of lung cancer types from cytological images using deep convolutional neural networks. *BioMed Res Int* 2017. Accessed June 3, 2018. doi:<https://doi.org/10.1155/2017/4067832>
6. Bengio Y: Learning deep architectures for AI. Boston: Now Publishers, Inc., 2009
7. Nicholson CV, Gibson A: A Beginner's guide to deep convolutional neural networks (CNNs). What's the difference between artificial intelligence (AI), machine learning and deep learning?. DeepLearning4j: Open-source, Distributed Deep Learning for the JVM. Accessed June 04, 2018. <https://deeplearning4j.org/convolutionalnetwork>
8. Sirinukunwattana K, Ahmed Raza SE, Tsang Y-W, Snead DRJ, Cree IA, Rajpoot NM: Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. *IEEE Trans Med Imaging* 35(5):1196–1206, 2016
9. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L: ImageNet: A large-scale hierarchical image database. *IEEE Conf Computer Vision Pattern Recogn (CVPR2009)* 248–255, 2009
10. Simonyan K, Zisserman A: Very deep convolutional networks for large-scale image recognition, ICLR 2015. Available at <https://arxiv.org/abs/1409.1556>. Accessed June 10, 2017.
11. Anthimopoulos M, Christodoulidis S, Ebner L, Christe A, Mougiakakou S: Lung pattern classification for interstitial lung diseases using a deep convolutional neural network. *IEEE Trans Med Imaging* 35(5):1207–1216, 2016. <https://doi.org/10.1109/tmi.2016.2535865>
12. CS231n, Convolutional neural networks (CNNs / ConvNets). CS231n Convolutional Neural Networks for Visual Recognition. Available at <http://cs231n.github.io/convolutional-networks/#case>. Accessed October 10, 2017
13. Joshi P: What is local response normalization in convolutional neural networks. Perpetual Enigma. Available at <https://prateekvjoshi.com/2016/04/05/what-is-local-response-normalization-in-convolutional-neural-networks/>. Accessed October 02, 2017
14. Géron ACA: Hands-on machine learning with Scikit-Learn and TensorFlow: Concepts, tools, and techniques to build intelligent systems. California: O'Reilly Media, 2017
15. Numpy and Scipy Documentation. Available at <https://www.numpy.org/>. Accessed June 03, 2017
16. MathWorks. Machine Learning with MATLAB - Quantcast. Available at https://www.mathworks.com/campaigns/products/display/machine-learning-with-matlab.html?s_eid=PSB_16169. Accessed October 03, 2017
17. Large Scale Visual Recognition Challenge 2014. ImageNet Large Scale Visual Recognition Competition 2014 (ILSVRC2014). Available at <http://www.image-net.org/challenges/LSVRC/2014/results>. Accessed October 01, 2017
18. Havaei M, Davy A, Warde-Farley D, Biard A, Courville A, Bengio Y, Pal C, Jodoin P-M, Larochelle H: Brain tumor segmentation with deep neural networks. *Med Image Anal* 35:18–31, 2017. <https://doi.org/10.1016/j.media.2016.05.004>
19. Tang S, Yuan Y: Object detection based on convolutional neural network, CS231n Stanford Final Writeup. Available at http://cs231n.stanford.edu/reports/2015/pdfs/CS231n_final_writeup_sjtang.pdf. Accessed June 02, 2017