A NEWDEEP CONVOLUTIONAL NEURAL NETWORK LEARNING MODEL FOR COVID-19 DIAGNOSIS

Henry Li[1] and Chen[2]

[1]Princeton International School of Mathematics and Science 19 Lambert Dr, Princeton NJ
[2]Chinese Academy of Science, Academy of Mathematics and System Science 55 ZhongGuanCun East Road, HaiDian District, Beijing

ABSTRACT

Ever since 2019, people from all over the world are talking about infection with SARS-CoV-2, also known as COVID-19. The symptoms range from asymptomatic conditions to fatal disease, with lung injury most frequently being the result of it. As time flies during the pandemic, the role of medical imaging has been more cortical than ever, with computed tomography being an alternative testing method combined with polymerase chain reaction testing to have a broader role. However, only performing medical imaging testing with suspected patients without classifying whether or not the patient has COVID-19 is not practical. Nevertheless, in many different areas, excellent pulmonology doctors are in extreme shortage for most of the developing counties. Even in developed countries, doctors are too busy with diagnosing and curing patients, so the need for classification for the medical image of patients to see whether they have COVID-19 or not is of high necessity. Moreover, studies of chest radiographs and CT images with applications of artificial intelligence have shown of big importance and necessity. In this paper, we mainly focus on the usage of deep learning and machine learning for categories with findings typical COVID-19 infected images and an application of medical images and testing of the system.

KEYWORDS

SARS-CoV-2, COVID-19, Lung injury, Medical imaging, pulmonology,Computed tomography, Polymerase chain reaction, Imaging classification, Deep learning, Machine learning

1. INTRODUCTION

It is generally believed that an infectious disease called COVID-19 was discovered in Wuhan, China in December 2019. The cause of it is generally believed to be Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2). In 2020, especially from March to June, the number of people infected with COVID-19 each day has tripled in many European and American Countries. Coronavirus disease is highly contagious. With the increasing number of confirmed cases, how to control this disease is becoming a public health challenge. The World Health Organization (WHO) declared the epidemic as a public health emergency of international concern (PHEIC) on January 30, 2020 [1], and declared it a pandemic on March 11, 2020 [2].

Researchers say that cases of infection can be confirmed by performing a test called reverse transcription-polymerase chain reaction (RT-PCR) [3]. More than 23 million COVID-19 positive cases have been reported until August 2020 [4]. However, the difficulty of quality control during sample preparation has resulted in a high false-negative rate. Therefore, accurate and fast image processing tools, especially those based on X-ray and computed tomography (CT), can help doctors. Otherwise, even if there are no common COVID-19 symptoms, suspicious patients will be hospitalized or isolated until laboratory test results are clear. The SARS-CoV-2 infection could cause asymptomatic to severe and even fatal diseases. The most common mortality rate is acute lung injury. During the pandemic, the role of imaging has become highly important.

Compared with RT-PCR testing, CT was initially an alternative method and a superior testing method but later evolved into a more limited effect based on specific indications. In the early stages of the pandemic, several chests imaging classification and reporting schemes were developed for patients suspected of COVID-19 to help classify patients with limited RT-PCR testing and unknown performance levels. Some studies have been carrying out observing lung involvement on chest radiographs and CT images, and they indicate that it is related to critical illness. In addition to pulmonary manifestations, cardiovascular complications such as thromboembolism and myocarditis have also been attributed to COVID-19, which sometimes leads to neurological and abdominal manifestations. [5] Finally, artificial intelligence has shown promise that can be used to determine the radiology and CT diagnosis and prognosis of COVID-19 pneumonia. [6] Therefore, the collection of chest images such as X-rays and CT scans plays an important role in limiting the spread of the virus and fighting COVID-19 at the appropriate stage of treatment. Artificial intelligence-based technology is used to make quick decisions in saving lives. [7] Among all possible methods, artificial intelligence-based methods tend to help doctors as an effective tool for diagnosing COVID-19. The image acquisition, segmentation, classification, and subsequent diagnosis phases developed between 2019 and 2020 are widely used [8].

2. DEEP LEARNING NETWORK AND MODELS

Deep Learning Networks are AI functions that imitate how the human brain works when processing data and creating patterns for decision making. It is a branch of machine learning of artificial intelligence. Its networks are capable of learning on their own without supervising. To train the network, unstructured or unlabeled data are being inputted. Until then, Artificial intelligence and machine learning technologies improved the accuracy of Covid-19 diagnosis. Also, most of the widely used deep learning models and methods were implemented and a small amount of data was used for COVID-19 diagnosis.[9] However, due to the rapid outbreak of Covid-19, there are not many real data sets available to the community. It is necessary to combine the observation and image information to make diagnoses of COVID-19. [10] AI can introduce more alternative ways to the medical system, and learn from multi-modal data to capture disease characteristics to obtain reliable results used for COVID-19 diagnosis for timely treatment. [9]

2.1. Convolutional NeuralNetwork

One kind of artificial neural network that is most commonly applied to analyze images is convolutional neural networks, which have properties of shift-invariant and space invariant based on the shared-weight structure of the convolution filters that slide along input features and provide translation equivariant responses known as feature maps. [11] Most of the convolutional neural networks are equivariant, which are being applied in for example image and video recognition, image classification, image segmentation, medical image analysis, natural language processing, brain-computer interfaces, and financial time series [12].

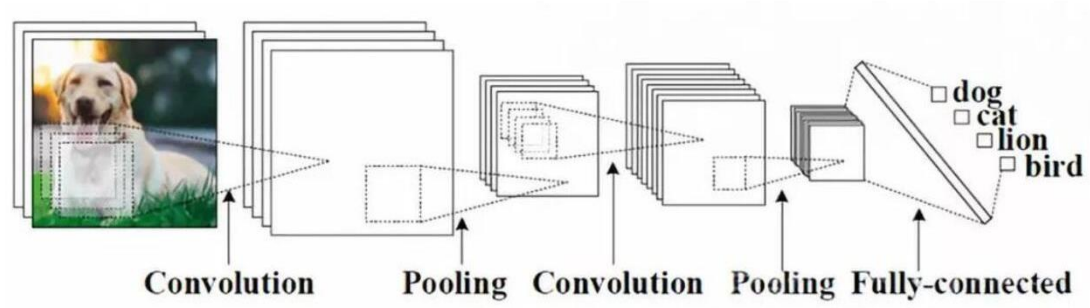


Figure 1: An example of the process of a convolutional network

Convolutional neural networks were designed base on the biological processes between neurons that resemble the human brain cortex. Individual cortical neurons respond to stimulations only in the receptive field, which is a restricted region of the visual field, and the receptive fields of different neurons partially overlap such that they cover the entire visual field. [13]

Convolutional neural networks have multilayer perceptron, which is thoroughly connected networks and every neuron in each layer is connected to all other neurons in the next layer. [14] The property of full connectivity of these networks makes them capable of processing overfitting data. CNNs take a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble patterns of increasing complexity using smaller and simpler patterns embossed in their filters [15].

Convolutional neural networks have a relatively shorter pre-processing procedure compared to other neural networks used for image classification algorithms. This means that the network learns to optimize the filters through automated learning, whereas in traditional algorithms these filters are hand-engineered, and the independence from prior knowledge and human intervention in feature extraction is a major advantage. [16]

2.2. U-net

What it takes to train deep convolutional networks are thousands of annotated training samples. An efficient sliding-window convolutional network and training strategy that relies on the strong use of data augmentation to use the available annotated samples more efficiently is named U-net [17]. This network structure consists of a contracting path for capturing contexts as well as the asymmetric expanding path that enables localization precisely, and it is also capable of training an end-to-end result from very few images and outputs the best and prior method and it is also extremely fast. Segmentation of a 512x512 image takes less than a second. [17] In the last few decades, deep convolutional networks have successfully finished a huge amount of visual recognition tasks. One of the most typical applications of convolutional networks is imaging classification, where the output for an image is a single class label. However, in many visual tasks, especially in biomedical image processing, the desired output should include localization. [17] Also, thousands of images used for training are hard to fulfill in the medical task. So, the idea of training a network in a sliding-window set up to predict the class label of each pixel by providing a local patch around that pixel as input has been performed by Ciresan et al, while this network can localize while the training data (counted by patches) much larger than the number of training images [18]. Due to the property of convolution, the output image is smaller than the input images with a constant ratio of border width.

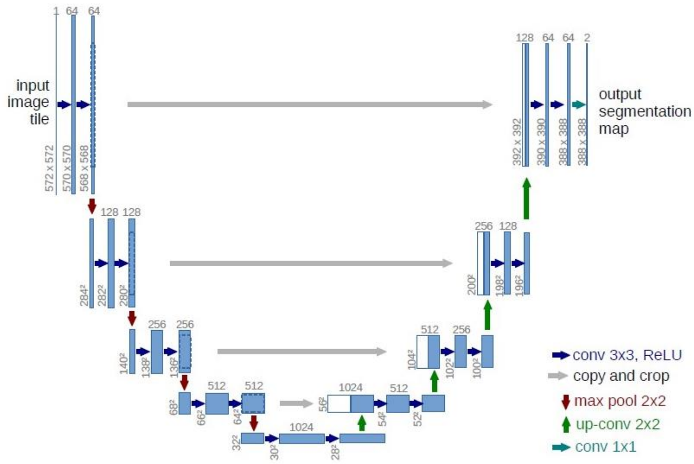


Figure 2: An example of a U-net structure for 32x32 pixels. Each blue box corresponds to a multi-channel map. The number of channels is shown on top of the box while the x-y-size is shown at the lower left edge of the box. White boxes represent copied feature maps, and the arrows denote the different operations. [17]

The network structure in Figure 2 consists of a contracting path, which is on the left side, and an expansive path which is on the right side. The former follows the typical structure of a convolutional network, which consists of the repeated application of two 3x3 unpadded convolutions, each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 for down-sampling. [19] By doubling the number of feature channels for each down-sampling step in the expansive path, which consists of an up-sampling of the feature map followed by a 2x2 convolution called "up-convolution" that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path with two 3x3 convolutions, each followed by a ReLU. [20] At the final layer, a 1x1 convolution is used to map each 64-component feature vector to the desired number of classes which in total, the network has 23 convolutional layers. [17]

A high momentum of 0.99 isused, as the energy function is definedwith a pixel-wise and soft-max with the cross-entropy loss function. The soft-max is defined as

$$p_k(x) = \frac{\exp(a_k(x))}{\sum_{k'=1}^{K} \exp(a_{k'}(x))} \qquad (1)$$

where $a_k(x)$stands for the activation for channel k at the pixel position x $\in$ Ω with $\Omega \subset \mathbb{Z}^2$, and K is the numberof classes and $p_k(x)$ is the approximated maximumfunction.

Thecross entropy then penalizes at each position the deviation of $p_{l(x)}(x)$ from (1) using

$$E = \sum_{x \in \Omega} w(x) \log(p_{l(x)}(x)) \qquad (2)$$

where $l : \Omega \rightarrow \{1, \ldots, K\}$ is the true label of each pixel and w: $\Omega \rightarrow \mathbb{R}$ isa weight map that gives some pixels more importance in thetraining. [17]

By pre-computing the weight map for basic segmentation to recompense the different pixels frequencies from a certain class in the training data set, and also forcing the network to learn small separation borders being introduced between each element, the separation border is computed using morphological operations. [21] The weight map is then computed as

$$w(x) = w_c(x) + w_0 \cdot \exp\left(-\frac{\left(d_1(x) + d_2(x)\right)^2}{2\sigma^2}\right) \qquad (3)$$

where $w_c : \Omega \rightarrow \mathbb{R}$ is the weight map for balancing the class frequencies, $d_1 : \Omega \rightarrow \mathbb{R}$ means the distance to the border of the nearest cell and $d_2 : \Omega \rightarrow \mathbb{R}$ the distanceto the border of the second nearest cell. In our experiments we set w_0 = 10 andσ= 5 pixels [17].In a deep network, many convolutional layers could be foundwith different paths throughoutthe whole network, which makes a good initialization weight extremely important sinceparts of the network might give excessive activations, and other parts never contribute otherwise. Ideally, the initial weights should be adapted such that each feature map in the network has approximately unit variance. [22] For a network with alternating convolutional and ReLU layers, this can be achieved by drawing the initial weights from a Gaussian distribution. [23]

3. Application of Deep Learning Models to Recognize COVID-19 Infected Lung Images

The main purpose of this research is to train the network tolet it recognize the COVID-19 Infected Lung Images. All the images are open-source chest CT images from Società Italiana di Radiologia Medica e Interventistica (Italian Society of Medical and Interventional Radiology) [24], National Institutes of Health (NIH), the Radiological Society of North America (RSNA),[25] the American College of Radiology (ACR), Centers for Disease Control and Prevention (CDC)[26].

3.1. TensorFlow Coding

TensorFlow is Google Brain's second-generation system, it is an open-source software library for machine learning, often focus on the training and inference of deep neural networks. [27] U-net models are available on TensorFlow, which is used to create a virtual environment for the networking building of this learning program.

3.2. Applications and Final Results

The used images are from the website mentioned before to train the network. The images in these websites are typical with labeled information, which is important for the training. We train the traininggroup images by classifying them. In total, there are 1000 images for training the network, and 100 for testing the network, and they are all being concluded into a dataset.

Figure 3: A part of the training image set

So first, inputting the training image data set allows the program to analyze and get a conclusion of the data distribution. We want to put the images into 4 groups, which are train images, train labels, test images, and test labels. All of the pictures are whether healthy lung chest CT images or COVID-19 infected lung chest CT images. So, these two labels are the ones that the program will be classicizing on.

```
(train_images, train_labels), (test_images, test_labels) = train_image.load_data()

class_names = ['Infected_image', 'Uninfected_image']
```

Before we train the network, we want to see the results of input.

```
train_images.shape
len(train_labels)
train_labels
test_images.shape
len(test_labels)
```

However, the input images are not the same size, so we want to pre-process the images for different images, there is the need to modify them into the same size.

```
plt.figure()
plt.imshow(train_images[0])
plt.colorbar()
plt.grid(False)
plt.show()
```

In order to test the size of the images, we show the first 25 images in the dataset.

```
plt.figure(figsize=(10,10))
for i in range(25):
        plt.subplot(5,5,i+1)
        plt.xticks([])
        plt.yticks([])
        plt.grid(False)
        plt.imshow(train_images[i], cmap=plt.cm.binary)
        plt.xlabel(class_names[train_labels[i]])
plt.show()
```

After all the preparation is done, we will begin to build our deep learning network. The first step is to set up the layers.The basic building blocks of a neural network is layers. Layers can extract representations from the data given.

```
model = keras.Sequential([
        keras.layers.Flatten(input_shape=(28, 28)),
        keras.layers.Dense(128, activation='A'),
        keras.layers.Dense(2)
])

model.compile(optimizer='B',
                          loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True
                          metrics=['accuracy'])
```

To begin with, the layer: tf.keras.layers.Dense hasa parameter that is learned in the process of training.The first layer: tf.keras.layers.Flatten of this networktransforms the format of the images from two-dimensional arraysto one-dimensional arrays. Then, after the pixels are flattened, the network consists of a sequence of two tf.keras.layers.Dense layers will be fully connected neural layers. The first Dense layer has 128 nodes. The second and last layer returns a logits array with a length of 10. Each node contains a score that indicates the current image belongs to one of the 10 classes.[28] In this case, another network is also available.

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(128, activation='A'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(2, activation='B')
])

model.compile(optimizer='C',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

Training the neural network model requires the following steps: 1) Input the training data into the model, which are train_images and train_labels arrays in this example. 2) The model learns to correlate images and labels. We ask the model to make predictions about a test set—in this example, the test_images array. [29] 3) labels from the test_labels array match the predictions [30]. To start training, we need to call the model.fit method. As the model trains, the loss and accuracy metrics are displayed. This model reaches an accuracy of about 0.89 on the training data. [17]

```
model.fit(train_images, train_labels, epochs=10)
```

The accuracy of the testing dataset is not as good as the accuracy of the training dataset, this gap between the training accuracy and the test accuracy is called overfitting. [31] It happens when a machine learning model does not perform as well as on new inputs other than the training data. An overfitted model "memorizes" the noise and details in the training dataset to a point where it negatively impacts the performance of the model on the new data [32]. There are a few ways to prevent overfitting, including 1) Simplifying the model, 2) Early stopping, 3) Use data augmentation, 4) Use regularization and 5) Use dropouts. [33]

```
test_loss, test_acc = model.evaluate(test_images, test_labels, verbose=2)

print('\nTest accuracy:', test_acc)
```

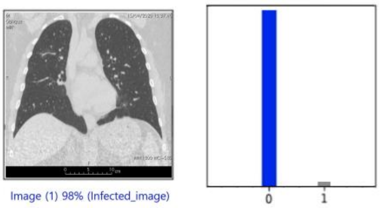
```
313/313 - 0s - loss: 0.3333 - accuracy: 0.8809

Test accuracy: 0.8809000253677368
```

After the model is trained, we can then make predictions about our testinggroup of images. Adding a soft-max layer to convert logits to probabilities will be easier to interpret.After this step, we will make a plot of how the classification class predicts the result.

```
probability_model = tf.keras.Sequential([model, tf.keras.layers.Softmax()])
predictions = probability_model.predict(test_images)
predictions[0]
np.argmax(predictions[0])
test_labels[0]
def plot_image(i, predictions_array, true_label, img):
    true_label, img = true_label[i], img[i]
    plt.grid(False)
    plt.xticks([])
    plt.yticks([])

    plt.imshow(img, cmap=plt.cm.binary)

    predicted_label = np.argmax(predictions_array)
    if predicted_label == true_label:
        color = 'blue'
    else:
        color = 'red'

    plt.xlabel("{} {:2.0f}% ({})".format(class_names[predicted_label],
                                100*np.max(predictions_array),
                                class_names[true_label]),
                                color=color)
```

We can use this model that we trained to first make predictions about some images for the first image in the testing groups, also for prediction array.

```
i = 0
plt.figure(figsize=(6,3))
plt.subplot(1,2,1)
plot_image(i, predictions[i], test_labels, test_images)
plt.subplot(1,2,2)
plot_value_array(i, predictions[i], test_labels)
plt.show()
```



Last but not least, our last goal is to make predictions about a single image based on the trained model.The tf.keras models are optimized to make predictions on a series of examples at once.[17] The method: tf.keras.Model.predict returns a list of lists for each image in the batch of data. In the end, it predicts the correct label as expected.

```
img = test_images[1]
print(img.shape)
img = (np.expand_dims(img,0))
print(img.shape)
predictions_single = probability_model.predict(img)
print(predictions_single)
plot_value_array(1, predictions_single[0], test_labels)
_ = plt.xticks(range(2), class_names, rotation=45)
np.argmax(predictions_single[0])
```

4. Conclusion

After comparing the characters of the network and the already-known characters of chest CT images with lung affected with COVID-19 [34], the COVID-19 infected lung images characters includes the presence of ground-glass opacities, presence of consolidation, presence of nodules, presence of a pleural effusion, presence of thoracic lymphadenopathy, airways abnormalities that include airway wall thickening, bronchiectasis, and endoluminal secretions, and last but not least, the presence of underlying lung disease such as emphysema or fibrosis. [35]

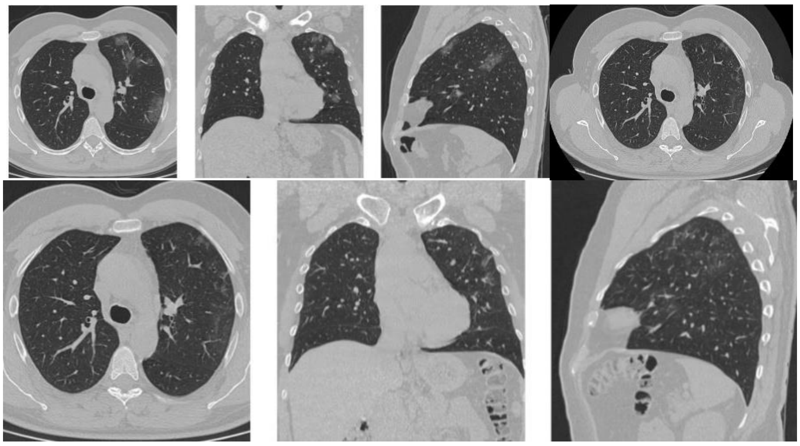


Figure4: COVID-19 infected chest CT lung Images provided by S.S. Oncological Radiology San Giuseppe Moscati Hospital, Taranto. A patient of a 51-year-old man presented himself to the hospital emergency department with an 8-day fever, chest tightness, and mild dyspnea. The image suggests that this patient with COVID-19 pneumonia manifests at thoracic CT investigation in the form of bilateral subpleural GGO opacities with aerial bronchogram, ill-defined margins, and a slight predominance in the right lower lobe [36].

5. Discussion

The ground-glass opacification is defined as an area of increased lung attenuation with preservation of bronchial and vascular margins [37], in other words, an area of increased hazy lung opacity with vessels and bronchial structures may still be seen. While pulmonary consolidation was defined as opacification with obscuration of margins of vessels and airway walls [38], in other words, a region of normally compressible lung tissue that has filled with liquid instead of air.

An important direction for future research is to alter the model used in the research to see if the images could also be classified into the correct group. Or using more images to test the stability of the network, maybe actually using it in medical applications. Also, noticing that we have concluded some important characteristics of the COVID-19 infected lung images, we could create more labels that include different kinds of symptoms. For example, labels with ground-glass opacification, or labels of thoracic lymphadenopathy. However, it requires the algorithm and the network to be more complicated, with more layers of convolutional network or more complicated models. Once finished, it would become a more mature neural network.

6. Acknowledgments

I would like to express my gratefulness to my mentor Dr. Chong Chen of Chinese Academy of Science. He has provided guidance and advices. I would also like thank Ms. Betty Wang for providing me this research opportunity. Last but not least, I would like to thank the organizers of the Yau Awards for giving me the platform to show my work.

References

[1] https://www.who.int/publications/m/item/covid-19-public-health-emergency-of- international-concern-(pheic)-global-research-and-innovation-forum

[2] https://www.who.int/director-general/speeches/detail/who-director-general-s -opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020

[3] Ai T, Yang Z, Hou H et al. Correlation of Chest CT and RT-PCR Testing for Coronavirus Disease 2019 (COVID-19) in China: A Report of 1014 Cases. Radiology. 2020;296(2): E32–E40. doi: https://doi.org/10.1148/radiol.2020200642. Link, Google Scholar

[4] Soomro, T.A., Zheng, L., Afifi, A.J. et al. Artificial intelligence (AI) for medical imaging to combat coronavirus disease (COVID-19): a detailed review with direction for future research. Artif Intell Rev (2021). 15 April 2021, https://doi.org/10.1007/s10462-021-09985-z

[5] https://dergipark.org.tr/en/pub/sdutfd/article/902875

[6] Kanne, J. P., Bai, H., Bernheim, A., Chung, M., Haramati, L. B., Kallmes, D. F., Little, B. P., Rubin, G., & Sverzellati, N. (2021). COVID-19 imaging: What we know now and what remains unknown. Radiology, 299(3), E262-E279. https://doi.org/10.1148/radiol.2021204522 [7] https://mayoclinic.pure.elsevier.com/en/publications/covid-19-imaging-what-we-know-now-and-what-remains-unknown

[8] Alexander Selvikvåg Lundervold, Arvid Lundervold, An overview of deep learning in medical imaging focusing on MRI, Zeitschrift für Medizinische Physik, Volume 29, Issue 2, 2019, Pages 102-127, ISSN 0939-3889, https://doi.org/10.1016/j.zemedi.2018.11.002. (https://www.sciencedirect.com/science/article/pii/S0939388918301181)

[9] http://web.archive.org/web/20210420225353/https://pubmed.ncbi.nlm.nih.gov/33875900/

[10] Soomro, T.A., Zheng, L., Afifi, A.J. et al. Artificial intelligence (AI) for medical imaging to combat coronavirus disease (COVID-19): a detailed review with direction for future research. Artif Intell Rev (2021). https://doi.org/10.1007/s10462-021-09985-z

[11] Avilov, Oleksii; Rimbert, Sebastien; Popov, Anton; Bougrain, Laurent (July 2020). "Deep Learning Techniques to Improve Intraoperative Awareness Detection from Electroencephalographic Signals". 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). Montreal, QC, Canada: IEEE. 2020: 142–145.

[12] https://blog.avenuecode.com/using-deep-convolutional-neural-networks-dcnns-for-time-series-forecasting-using-tensorflow-part-1

[13] https://github.com/ankit201/cnn

[14] Zhang, Wei (1988). "Shift-invariant pattern recognition neural network and its optical architecture". Proceedings of Annual Conference of the Japan Society of Applied Physics.

[15] Tsantekidis, Avraam; Passalis, Nikolaos; Tefas, Anastasios; Kanniainen, Juho; Gabbouj, Moncef; Iosifidis, Alexandros (July 2017). "Forecasting Stock Prices from the Limit Order Book Using Convolutional Neural Networks". 2017 IEEE 19th Conference on Business Informatics (CBI). Thessaloniki, Greece: IEEE: 7–12.

[16] Fukushima, K. (2007). "Neocognitron". Scholarpedia. 2 (1): 1717. Bibcode:2007SchpJ...2.1717F. doi:10.4249/scholarpedia.1717

[17] Ronneberger 2015 unet, U-Net: Convolutional Networks for Biomedical Image Segmentation, Olaf Ronneberger and Philipp Fischer and Thomas Brox, 2015, 1505.04597

[18] Ciresan, D.C., Gambardella, L.M., Giusti, A., Schmidhuber, J.: Deep neural networks segment neuronal membranes in electron microscopy images. In: NIPS, pp. 2852–2860 (2012)

[19] https://heartbeat.fritz.ai/deep-learning-for-image-segmentation-u-net-architecture-ff17f6e4c1cf

[20] N. Siddique, S. Paheding, C. P. Elkin and V. Devabhaktuni, "U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications," in IEEE Access, vol. 9, pp. 82031-82057, 2021, doi: 10.1109/ACCESS.2021.3086020.

[21] Perseus AI, U-net[Electronic resource]. URL: https://perseusai.blogspot.com/

[22] Talley, Nicholas Joseph (2001). Clinical Examination, a Clinical Guide to Physical Diagnosis, Wiley, 4th ed., p. 121, ISBN 0632059710.

[23] Cornell University, U-net[Electronic resource]. URL:https://arxiv.org/abs/1806.05034

[24] Italian Society of Radiology Via della Signora 2[Electronic resource]. URL: https://sirm.org/

[25] RSNA, Radiology [Electronic resource].URL:https://pubs.rsna.org/doi/full/10.1148/radiol.2021204522

[26] Imaging Technology News [Electronic resource]. URL:https://www.itnonline.com/content/photo-gallery-how-covid-19-appears-medical-imaging

[27] Google (2015). TensorFlow: Open source machine learning. "It is machine learning software being used for various kinds of perceptual and language understanding tasks" – Jeffrey Dean, minute 0:47 / 2:17 from YouTube clip

[28] TensorFlow[Electronic resource]. URL:https://www.tensorflow.org/guide/keras/train_and_evaluate

[29] RICE University [Electronic resource]. URL:https://www.caam.rice.edu/~caam519/notes/module-8/MNIST.ipynb

[30] TensorFlow[Electronic resource]. URL:https://colab.research.google.com/github/tensorflow/docs-l10n/blob/master/site/zh-cn/guide/basic_training_loops.ipynb

[31] IBM Cloud Learn Hub [Electronic resource]. URL:https://www.ibm.com/cloud/learn/overfitting

[32] The Data Engineering Cloud [Electronic resource]. URL:https://www.kdnuggets.com/2019/12/5-techniques-prevent-overfitting-neural-networks.html

[33] Machine Learning [Electronic resource]. URL:https://machinelearningmastery.com/introduction-to-regularization-to-reduce-ovrfitting-and-improve-generalization-error/

[34] Tensorflow[Electronic resource]. URL:https://colab.research.google.com/github/tensorflow/docs-l10n/blob/master/site/zh-cn/tutorials/keras/classification.ipynb?hl=zh_cn#scrollTo=xvwvpA64CaW_

[35] Jerome JTJ et al. Perspectives and Consensus among International Orthopaedic Surgeons during Initial and Mid-lockdown Phases of Coronavirus Disease. J Hand Microsurg. 2020 Dec;12(3):135-162. doi: 10.1055/s-0040-1713964. Epub 2020 Jul 6. PMID: 33408440; PMCID: PMC7773504.

[36] Chest CT Findings in Coronavirus Disease-19 (COVID-19): Relationship to Duration of Infection, Adam Bernheim, Xueyan Mei, Mingqian Huang, Yang Yang, Zahi A. Fayad, Ning Zhang, Kaiyue Diao, Bin Lin, Xiqi Zhu, Kunwei Li, Shaolin Li, Hong Shan, Adam Jacobi, and Michael Chung, Radiology 2020 295:3

[37] https://sirm.org/2020/05/31/covid-19-caso-115/

[38] Differential diagnosis and management of focal ground-glass opacities, M. Infante, R. F. Lutman, S. Imparato, M. Di Rocco, G. L. Ceresoli, V. Torri, E. Morenghi, F. Minuti, S. Cavuto, E. Bottoni, F. Inzirillo, U. Cariboni, V. Errico, M. A. Incarbone, G. Ferraroli, G. Brambilla, M. Alloisio, G. Ravasi, European Respiratory Journal Apr 2009, 33 (4) 821-827; DOI: 10.1183/09031936.00047908