Jurnal Ilmiah KURSOR
Menuju Solusi Teknologi Informasi
Vol. 11, No. 3, Juli 2022
ISSN 0216 – 0544
e-ISSN 2301– 6914

APPLICATION OF COST-SENSITIVE CONVOLUTIONAL NEURAL NETWORK FOR PNEUMONIA DETECTION

[a]Rizki Anantama, [b]Hadi Suyono, [c]Muhammad Aswin

[a, b, c]Departement of Electrical Engineering Brawijaya University, Malang, Indonesia
E-mail: rizkianantama11@gmail.com, hadis@ub.ac.id, maswin@ub.ac.id

Abstract

Pneumonia is a disease that can be caused by a viral, bacterial, or fungal infection. One of the processes for diagnosing pneumonia is by using X-ray images. Several studies have tried to perform automatic classification of X-ray images to detect pneumonia images from healthy images. One of these approaches utilizes a convolutional neural network to perform the classification. However, the problem is that most of these studies only focus on accuracy, without considering performance criteria such as sensitivity and specificity. This study proposes the use of a Cost Sensitive approach in training Convolutional Neural Networks. Training with a cost sensitive approach provides different loss values for different types of misclassifications, to optimize a certain performance, in this case specificity. This training was conducted on a convolutional neural network model and compared with the same model which was trained with a non-cost sensitive approach. It was found that the cost sensitive training had a lower accuracy performance at 71.47%, but the specificity level was higher, at 85.1%, compared to the non-cost sensitive approach with an accuracy of 76.16% and specificity at 75.2%. This higher level of specificity is better for medical applications where false negative errors have a greater impact than other types of errors.

Key words: Convolutional neural network, cost-sensitive, classification, pneumonia.

INTRODUCTION

Pneumonia is a lung infection that has a moderate to severe level, which requires the patient to get further treatment in a hospital. Pneumonia is caused by infection by bacteria, viruses, or fungi that attack the air sacs in the lungs (alveoli) causing a buildup of fluid or pus, and interfering with breathing in absorbing the oxygen that the patient's body needs to reach the blood vessels [1].

One of the diagnostic methods to detect pneumonia is to use an X-Ray image of the patient's chest. However, to determine whether the patient has pneumonia, the classification of the X-ray image must be carried out by a qualified medical professional[2]. This classification is carried out to determine the class of an X-ray image, whether it is an X-ray image with pneumonia, or a healthy X-ray image. The main problem is the limited availability of qualified medical professionals. To overcome this problem, several studies have proposed the use of a convolutional neural network (CNN) [3]to perform this classification process automatically.

These studies include [4] using the Recurrent convolutional neural network (R-CNN) which produces 56% accuracy, and (Abiyev & Ma'aitah, 2018) using the convolutional neural network (CNN). and achieved an accuracy of 92.4% using a dataset with a size of 1000 images, then by [5] who trained their own CNN-based model with an accuracy of 93.73%, to the transfer learning approach and weighted classifier by [6] which results in an accuracy of up to 98.4%. One of the main problems of these studies, is that the discussion carried out places more emphasis on maximizing performance in terms of accuracy, by taking into account the performance of precision and recall[7]. In the realm of medical diagnostics, one of the additional performance values required is specificity (ratio of true negatives to true negatives and false positives). This is because an event due to failure to detect a patient with pneumonia has a worse impact than an event when a healthy patient is detected as having pneumonia. In the first case, the disease is not treated and can become severe to the point of death, while in the second event, further diagnosis will find that the case is simply a misdiagnosed, thus only resulting in increased medical costs. One approach that can be used is to use a cost sensitive approach [8]. The cost sensitive approach provides a different loss value at the time of learning, so it can shift the performance value towards a certain performance value, such as specificity. Based on this background, this research will focus on making a model classifier that aims to classify images into two classes – pneumonia or healthy – based on CNN but with a training focus on optimizing model specificity.

MATERIAL AND METHODS

Neural Network

An artificial neural network or often referred to as a neural network, is a mathematical model that attempts to replicate biological neural interaction patterns in animals. This computational model tries to emulate the ability of a biological nervous system to learn and do a job, without having to program specifically about the rules needed to do the job[9].

The neural network computational model tries to do its job without first having prior knowledge about the image, that for example there is a special feature in the object to be recognized. This is for example that the image of a cat has the characteristics of eyes, whiskers, nose, and so on. In the training process, the neural network will try to identify these characteristics, to ensure a probability value of whether an object is in an image[10].

This computational model takes several inputs, then applies a weight to these inputs, adds them, and then puts them into an activation function. The activation model of this function is that if the input is within a certain range, the output will be active. This model is shown in Figure 1.

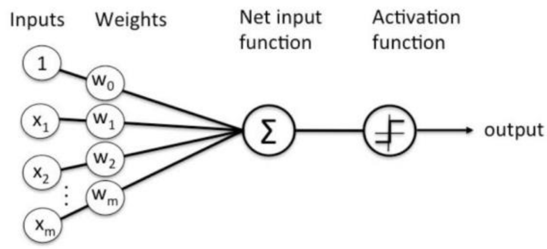

Fig 1. Perceptron model[11]

Convolutional Neural Network

Convolutional Neural Network (CNN) is a type of neural network, which excels in classifying 2-dimensional images. CNN is a regularized neural network of perceptrons[12]. CNN has a better efficiency level than the perceptron, especially in the image recognition process. In addition, the relationship pattern on CNN has a smaller pattern, where the relationship between input and output is a convolution operation using a kernel [13]. This kernel is used for each filter, so that in the CNN model, fewer parameters need to be changed in an image recognition problem[14]. CNN inherently solves the problem by extracting features that are hierarchically patterned in the data, and rearranging a more complex pattern, using smaller and simpler patterns, so that the pattern of connectivity and its complexity is greatly reduced[15]. One of the uses of convolutional neural network in image classification is shown in Figure 2.

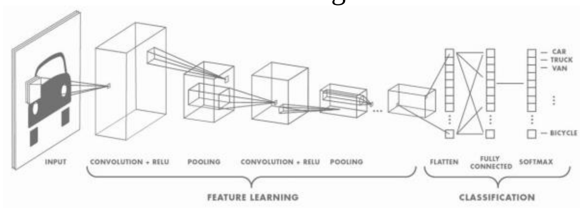

Fig 2. An example of using a convolutional neural network

In CNN, the output of the network is the result of the convolution product between the input and the kernel. The size of the kernel used by CNN can vary, but in general the shape of the kernel has a square shape, where the number of rows and columns of the kernel is the same[16]. The kernel is also referred to as a filter, which is a filter that separates a feature in the image into a separate value, and if there is more than one convolution filter in one layer, the output results will consist of several sets of feature maps with a total number of features. feature maps that are less or equal to the number of filters[13].

After a convolutional layer, generally a pooling layer is created, where this pooling layer will apply a pooling function that detects the presence of a feature in the area designated by the pooling layer[18]. The purpose of this layer is to reduce the sensitivity of the CNN to image shift. This function sums the images in the range, performs the operation (in this case the average operation), and outputs the result as a composite data of several entries[19]. It is shown in Figure 3.

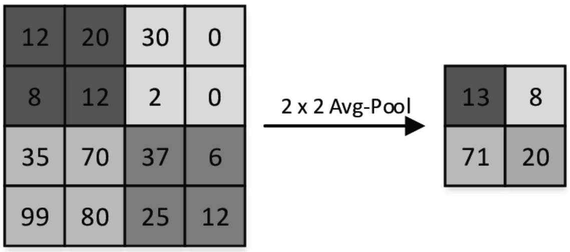

Fig 3. Calculation on average pooling layer [11]

Deep Learning

Deep learning is an implementation of a neural network. The deep learning approach aggregates neurons into layers. The layers containing these neurons are then arranged so [19] that the output of one layer becomes the input for the next layer. The arrangement of these layers departs from the nature that a perceptron alone will not be able to become a universal classifier. However, sequentially arranged perceptrons can act as a universal classifier [20]. The application of deep learning is not limited to only using the perceptron, but also using other models such as convolutional neural network, and recurrent neural network.[12].

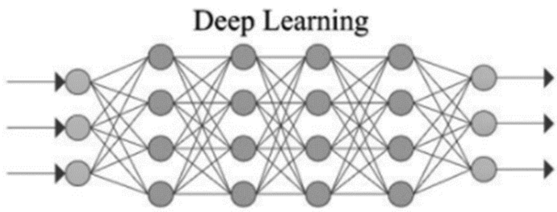

Fig 4. Deep learning architecture

Cost-Sensitive Learning

Cost sensitive learning is one of the methods for machine learning, to solve problems related to data imbalances, or imbalances resulting from prediction results. The concept of this learning is to allocate a different cost to each misclassification event, so that the performance of the classifier can increase, or shift towards the desired performance value[8]. This method assumes that a misclassification event that is carried out on a minority class, or a class that affects the performance value, will have a greater cost, so that the learning process will focus more on the process of classifying towards the desired performance value [8]. The output of this cost sensitive learning is a cost function value, which is the weight given to a misclassification event. In the neural network learning process, the cost function is used in learning as an optimization parameter, where the neural network will be directed to optimize the cost, by iteratively pursuing a smaller cost[21]. For example, in a supervised learning problem in a two-class classification, there are four classification events: true positive, true negative, false positive, and false negative[8]. When it is desired to increase the accuracy parameter, the cost is assigned as much as 1 when a false positive or false negative occurs. Vice versa, the cost of true positive and true negative events is 0. This can be used to tune the model by applying a value for each different type of error.

RESULT AND DISCUSSION

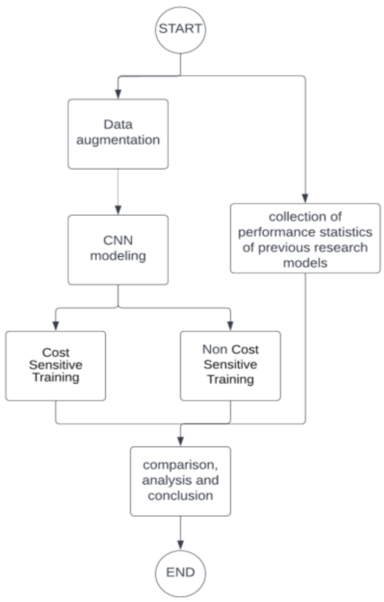

Fig 5. Flowchart of research stages

Added Noise

To the data, it is planned to add inner noise with a normal distribution, or often referred to as Gaussian noise. This type of noise is noise that has a mean of 0, but with a specified standard deviation value. The addition of one of the techniques used to force the model to try to learn a clearer pattern in an image, and indirectly also prevents the occurrence of an overfitting of data. This concept is similar to the use of a dropout layer which turns off connections between layers in the model, except that this process is carried out on image data that will be trained on the convolutional layer. The process of adding this Gaussian noise uses Eq (1).

$$x' = x + \text{noise}(n, \text{stdev}) \qquad (10)$$

Where:
x' = Data after adding noise
x = Data before adding noise
noise = Function that returns the noise value
n = average value of noise
stdev = standard deviation of noise

Convolutional Neural Network Modeling

A convolutional neural network based model is made. This model will be divided into two main parts. The first part consists of a series of convolutional neural network layers that will extract the features contained in the image. Then after extraction of the features in the image, it will be entered into a dense network layer which is in charge of determining whether the image belongs to a certain category. Because this problem is only discussed regarding the characteristics of images infected with pneumonia, in this discussion it is determined as a binary classification problem, namely X-ray image data that is healthy or has pneumonia. The software library runs on top of the python language. The parameters of the model to be used are compiled and shown in Table 1.

Table 1. Convolutional layer arrangement

No	Layer	Pooling Layer	Units/Filter	Activation Function
1	Convolutional	MaxPool	32	Rec. Linear Unit
2	Convolutional	MaxPool	64	Rec. Linear Unit
3	Convolutional	MaxPool	128	Rec. Linear Unit
4	Convolutional	MaxPool	256	Rec. Linear Unit

The parameters of the model used, will take input data with a two-dimensional image data size of 300x300x3, where 3 is the number of color channels from three RGB color compositions (red, green, blue). The input is passed into four convolutional layers, with a unit count of 32, 64, 128, and 256, respectively. The output of this convolutional layer will then be included in the next row of layers. The next layer consists only of the dense network layer, which is described in more detail in Table 2

Table 2. Layers of dense network

No	Layer	Units/Filters	Activation Function
1	Flatten	-	
2	Dense	32	Rec. Linear Unit
3	Dense	1	Sigmoid

Non-Cost-Sensitive Model Training

In the first stage, training will be conducted without using a cost sensitive approach. In this normal approach, a misclassification value will have the same weight, both against false positive and false negative misclassifications. This will result in the training process trying to improve overall accuracy. The confusion matrix to be used refers to the Table 3.

Table 3. Confusion Matrix Non Cost Sensitive Training

	Predicted Negative	Predicted Positive
Actual Negative	0	1
Actual Positive	1	0

This training process will use the same parameters, except the loss function parameter. The parameters that will be used in this training are as listed in Table 4

Tabel 4 Non cost sensitive model training parameters

No	Parameter	Value
1	Optimizer	RMSprop
2	Learning Rate	0.001
3	Metrics	Accuracy
4	Batch size	128
5	Epochs	20

This training is then carried out using augmented data. The addition of this augmentation is carried out at the time of calling the data for the training batch, because the data used is provided by streaming. In each training batch, there are 128 data provided by the object based on the software library used. The training progress process is shown in Figure 6.

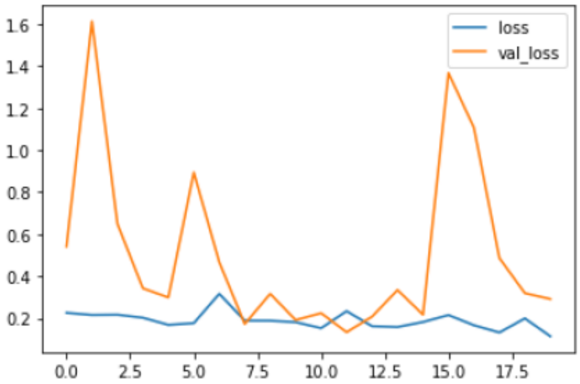

Fig 7. Progress of loss value based on training epoch

From the training process that has been carried out, then testing the model. This test is carried out using unaugmented test data. The augmentation process in this training is only carried out on the data used for training. The value of the test results is shown in Table 5.

Table 5. Non cost sensitive training model test results

Parameter	Value
Loss	1.1318
Accuracy	75.16%

Cost-Sensitive Model Training

In the second stage, training will be conducted using a cost sensitive approach. In this particular learning approach, a misclassification value will have a different weight, when compared to misclassification in different classes. In this case, the FP misclassification (false positive) has a lower weight than the FN misclassification (false negative). This will result in the training process being lopsided in the direction of suppressing the incidence of false negatives, which can result in an increase in false positive values. The confusion matrix to be used refers to the Table 8.

Table 9. Confusion matrix non cost sensitive training

	Predicted Negative	Predicted Positive
Actual Negative	0	0.05
Actual Positive	1	0

This training process will use the same parameters, except the loss function parameter, which in training with this approach, uses the weight parameters described above. The parameters that will be used in this training are as listed in the Table 7.

Table 7. Cost sensitive model training parameters

No	Parameter	Value
1	Optimizer	RMSprop
2	Learning Rate	0.001
3	Metrics	Accuracy
4	Batch size	128
5	Epochs	20

This training is then carried out using augmented data. The addition of this augmentation is carried out at the time of calling the data for the training batch, because the data used is provided by streaming. In each training batch, there are 128 data provided by the object based on the software library used. The training progress process is generated in Figure 7.

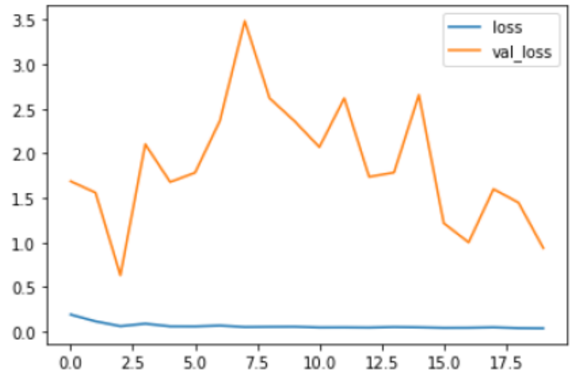

Fig 10. Training results and loss value to training epochs

After the training process with the cost sensitive approach is complete, the training process is carried out, then the next step is to test the model. This test was carried out using a test dataset with a total of 624 data. The test results are shown in Table 8.

Table 8. Cost sensitive training model test results

Parameter	**Value**
Loss	0.8887
Accuracy	71.47%

Performance Comparison

At this stage, a performance comparison between the two models will be carried out, and a comparison with the approaches and results of previous research. The results of this complete comparison will include comparisons of accuracy, precision, recall, sensitivity, and specificity. This comparison is described more fully in the Table 9.

Table 9. Comparison With Previous Research

No	**Researcher**	**accuracy**	**Precision**	**Recall**	**Sensitivity**	**Specificity**
1	Stephen et. al.	93.73	-	-	-	-
2	Cohen et. al.	92.8	90.1	93.2	-	-
3	Rajaraman et. al.	96.2	97.0	99.5	-	-
4	Hashmi et. al.	98.43	98.26	99.0	-	-
5	Rizky-CS*	71.47	-	-	66.4	85.1
6	Rizky-NCS**	76.16	-	-	77.3	75.2

Performance Analysis

Research has been carried out, namely by making a convolutional neural network model, which is trained using a cost sensitive approach, and which is trained using a non-cost sensitive approach. In training with a cost sensitive approach, the accuracy value tends to be lower than the non-cost sensitive approach, at 71.47%, compared to 76.16%. However, the model trained with the cost sensitive approach had an advantage in performance specificity at 85.1%, compared to 75.2% in the non-cost sensitive training approach. From these results, it is concluded that the cost sensitive training approach is able to increase specificity performance, but with the effect of experiencing a slight decrease in accuracy performance.

A further pragmatic explanation of this performance value is carried out as follows using the help of a confusion matrix. In this confusion matrix, four possible events are shown. The actual true-predicted true, actual true predicted false, actual false predicted true, and actual false predicted false events.

The accuracy rate is the number of actual true predicted true and actual false predicted false. It shows the ratio of the predicted number of pneumonia when a patient has pneumonia and the prediction of being healthy when a patient is completely healthy. When the accuracy rate is high, the prediction error rate is low, and the problems caused by incorrect predictions are less.

Then, the specificity level shows the actual number of false predicted false, against the number of predicted false. It shows how many people detected as healthy people, are actually healthy people, against the predicted number of healthy people. This matter

It is important to improve, because when this value is low, many patients actually have pneumonia but are predicted to be healthy. This of course can be dangerous, because the disease becomes untreated. The example in Table 10 shows a specificity value of 96.33%, which is from 315 / (315 + 12).

Table 10. Example of confusion matrix for specificity calculation

	Predicted True	Predicted False
Actual True	312	12
Actual False	6	315

In addition, when a comparison is made between the results of this study and the performance of models from previous studies. It was found that in general the model has an accuracy performance that is not as good as the model from previous studies. However, a high specificity value was found compared to the normal approach, which is one of the important indicators of system performance.

CONCLUSION

The conclusions that can be drawn from this research are as follows:

1. Convolutional neural network-based model has been successfully created and trained using cost-sensitive and non-cost-sensitive training. In cost sensitive training, accuracy and specificity were found at 71.47% and 85.1%, respectively. Meanwhile, training using a non-cost sensitive approach was found to have accuracy and specificity of 76.16% and 75.2%, respectively.
2. Performance comparison shows that the model trained with the cost sensitive approach has a higher specificity value, but a slightly lower accuracy value than the model trained with the non-cost sensitive approach.
3. However, in general the accuracy of the two models is still lower than the model in the previous study. For future development, it is recommended that research use a model that has a higher accuracy performance, but uses a cost sensitive approach to optimize the model towards specificity.
4. In the future, research can be carried out using models that have proven performance in terms of accuracy, however, with a cost sensitive approach to encourage models to be more focused on this.

REFERENCES

[1] D. S. Kermany et al., “Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning,” Cell, vol. 172, no. 5, pp. 1122-1131.e9, Feb. 2018, doi: 10.1016/j.cell.2018.02.010.

[2] W. Khan, N. Zaki, and L. Ali, “Intelligent Pneumonia Identification from Chest X-Rays: A Systematic Literature Review”, doi: 10.1101/2020.07.09.20150342.

[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2323, 1998, doi: 10.1109/5.726791.

[4] A. Ismail, T. Rahmat, and S. Aliman, “CHEST X-RAY IMAGE CLASSIFICATION USING FASTER R-CNN,” MALAYSIAN JOURNAL OF COMPUTING, vol. 4, no. 1, p. 225, Jul. 2019, doi: 10.24191/mjoc.v4i1.6095.

[5] O. Stephen, M. Sain, U. J. Maduh, and D. U. Jeong, “An Efficient Deep Learning Approach to Pneumonia Classification in Healthcare,” Journal of Healthcare Engineering, vol. 2019, 2019, doi: 10.1155/2019/4180949.

[6] M. F. Hashmi, S. Katiyar, A. G. Keskar, N. D. Bokde, and Z. W. Geem, “Efficient pneumonia detection in chest xray images using deep transfer learning,” Diagnostics, vol. 10, no. 6, Jun. 2020, doi: 10.3390/diagnostics10060417.

[7] T. Malygina, T. M. Ai, E. Ericheva, and I. Drokin, “GANs ’N Lungs: improving pneumonia prediction.”

[8] S. H. Khan, M. Hayat, M. Bennamoun, F. Sohel, and R. Togneri, “Cost Sensitive Learning of Deep Feature Representations from Imbalanced Data,” Aug. 2015, [Online]. Available: http://arxiv.org/abs/1508.03422

[9] N. Samani, M. Gohari-Moghadam, and A. A. Safavi, “A simple neural network model for the determination of aquifer parameters,” Journal of Hydrology, vol. 340, no. 1–2, pp. 1–11, Jun. 2007, doi: 10.1016/j.jhydrol.2007.03.017.

[10] I. M. Nasser and S. S. Abu-Naser, “Lung Cancer Detection Using Artificial Neural Network,” 2019. [Online]. Available: www.ijeais.org

[11] Simplilearn, “What is Perceptron: A beginners guide for Perceptron [updated],” *Simplilearn.com*, 20-Jul-2022. [Online]. Available: https://www.simplilearn.com/tutorials/deep-learning-tutorial/perceptron. [Accessed: 25-Jul-2022].

[12] B. Md Anwar Hossain, M. Shahriar Alam Sajib, M. Anwar Hossain α, and M. Shahriar Alam Sajib σ, “Classification of Image using Convolutional Neural Network (CNN),” 2019.

[13] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, “A Convolutional Neural Network for Modelling Sentences,” Apr.

2014, [Online]. Available: http://arxiv.org/abs/1404.2188

[14] T. Liu, S. Fang, Y. Zhao, P. Wang, and J. Zhang, "Implementation of Training Convolutional Neural Networks."

[15] M. Mostavi, Y. C. Chiu, Y. Huang, and Y. Chen, "Convolutional neural network models for cancer type prediction based on gene expression," *BMC Medical Genomics*, vol. 13, Apr. 2020, doi: 10.1186/s12920-020-0677-2.

[16] B. Chen, H. Li, and W. Luo, "Image Processing Operations Identification via Convolutional Neural Network," Sep. 2017, doi: 10.1007/s11432-018-9492-6.

[17] R. H. Abiyev and M. K. S. Ma'aitah, "Deep Convolutional Neural Networks for Chest Diseases Detection," *Journal of Healthcare Engineering*, vol. 2018, 2018, doi: 10.1155/2018/4168538.

[18] F. Han, J. Yao, H. Zhu, and C. Wang, "Underwater Image Processing and Object Detection Based on Deep CNN Method,". *Journal of Sensors*, vol. 2020, 2020, doi: 10.1155/2020/6707328.

[19] H. M. Ahmad and A. Rahimi, "Deep learning methods for object detection in smart manufacturing: A survey," *Journal of Manufacturing Systems*, vol. 64, pp. 181–196, Jul. 2022, doi: 10.1016/j.jmsy.2022.06.011.

[20] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, Apr. 2016, doi: 10.1016/j.neucom.2015.09.116.

[21] M. Mostavi, Y. C. Chiu, Y. Huang, and Y. Chen, "Convolutional neural network models for cancer type prediction based on gene expression," *BMC Medical Genomics*, vol. 13, Apr. 2020, doi: 10.1186/s12920-020-0677-2.

.