# Classification of Tuberculosis using a Convolutional Neural Network

**Husein Abdul Hamid*[1], Yufis Azhar[2], Zamah Sari[3]**
[1,2,3]Teknik Informatika/Universitas Muhammadiyah Malang
yufis@umm.ac.id

***Abstract***

*Tuberculosis disease causes the death of 1.8 million people worldwide, according to data from the World Health Organization in 2018 that nearly 10 million people were affected by tuberculosis and about 98,000 of them died. Technological developments, especially computer engineering, have helped accelerate the diagnosis of tuberculosis in poverty source areas. One of them is Computer Vision and Machine Learning Technology which causes development and utilization in all aspects, including the health sector. In the development of computer vision, there is a deep learning technique, this technique can automatically detect and classify various diseases with better accuracy. One method of deep learning that can produce precise accuracy and better efficiency is using a Convolutional Neural Network (CNN). This study uses CNN in the classification of the Tuberculosis Chest X-ray Database. The study was carried out in 4 scenarios, with each scenario getting accuracy results of 70%, 97%, 97%, and 72%..*

***Keywords:*** *Internet of Things Platform, Internet of Things, Message Queuing Telemetry Transport, MQTT Broker Server, Image Classification, Deep Learning, Tuberculosis Disease, Convolutional Neural Network*

## 1. Pendahuluan

Tuberculosis (TB) is an infectious disease caused by Mycobacterium tuberculosis and this virus attacks especially the lungs (Supartini & Hindarto, 2016). This disease causes the death of 1.8 million people worldwide, according to data from the World Health Organization in 2018 that almost 10 million people were affected by TB disease and around 98,000 of them died (Rochmawanti et al., 2021). This disease is curable and preventable, but in resource-poor and marginalized communities with weak health infrastructure, it is difficult to detect this disease early (C. Liu et al., 2018). Due to insufficient resources for better diagnosis of this disease and effective follow-up treatment. Detecting TB disease early is very important so that patients can be treated immediately and minimized (Septarini, 2017).

Technological developments, especially technical computers, have helped speed up the diagnosis of TB in poverty source areas. One of them is Computer Vision and Machine Learning Technology causing development and utilization in all aspects, including one in the health sector (Rochmawanti et al., 2021). Computers are used as a tool to understand information and recognize diseases like a doctor who can diagnose someone when they are sick (Putra et al., 2021). One form of development is that computers can recognize diseases contained in images like humans (Ahyuna & Aryasa, 2017). One of the diseases that can be recognized is Tuberculosis which is found in the human lungs

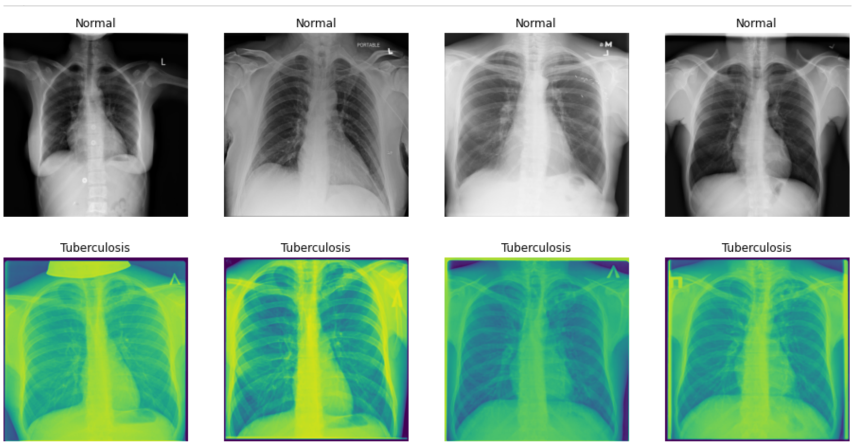
X-ray images are often used in radiological examinations to screen and diagnose many lung diseases (Rahmadewi & Kurnia, 2016). Many techniques can be obtained from x-ray images such as Magnetic Resonance Imaging (MRI), Computed Tomography (CT), or Positron Emission Tomography (PET) (Wang et al., 2019). This technique is used to see and describe the structure of human organs to make it easier for many people to recognize the disease suffered by a person. In using x-ray images, analysis is also needed so that the results obtained can be implemented for further actions such as treatment if someone is indicated to have a disease in the image. Not everyone can read and see the results of x-ray images, this is where the task of a radiologist is to read and see from various perspectives whether the patient is indicated or not.

## 2. Research Method

This research focuses on solving and finding the right CNN architecture in classifying Tuberculosis disease found in human lungs with data labels Normal and Tuberculosis which will be tested based on the results of accuracy in each model and the architecture built. Before the model is built, several things must be developed and processed by researchers, in determining the model to be built there are several processes.

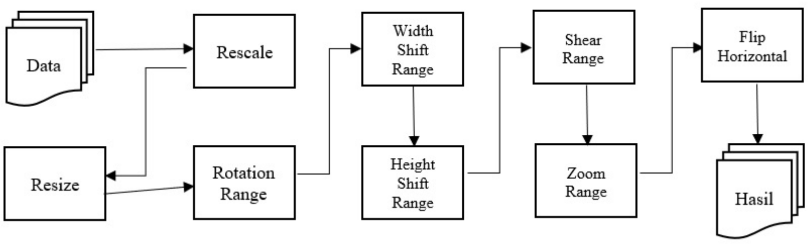
The first is to identify the problem, this is needed so that the research is as expected by the researcher after reading the reference journal. The problem that will be solved in this study is to make the right CNN architecture in classifying Tuberculosis diseases found in the human lung, there are 2 X-ray image labels namely Normal and Tuberculosis which will be tested for the accuracy of the architecture. To find out the problems that occurred in the writing of this final project, the authors conducted a literature study on previous research to explore information.

The two researchers collected data sets that can be used at this research stage, the data affect the test results. This dataset has a total data of 4200 X-ray images with a total image size of 512 x 5125 x 1 pixel. Can be seen in Figure 1 are 4 images of Normal X-Ray Images and in Figure 2 are 4 images of Xray Images of Tuberculosis.




The third is the distribution of the dataset, this needs to be done so that the data set can be following the architecture that will be built by the researcher. Researchers have provisions with the amount of data owned as much as 4200, the dataset will then be divided into 3 parts, namely 80% training data with a total of 3360 training images, 10% test data with a total of 420 test images, and 10% testing data with a total test image of 420.

Fourth is the use of augmented data, this technique is needed so that the architecture to be processed has the same image with several parameters determined by the researcher such as rotating the image randomly (rotation range), resizing the image (resizing), changing the raster value range (rescale), change the width shift range, change the height shift range, change the shear range, change the zoom range and flip horizontal.




The fourth is to implement several architectural models that will be used in this research. The researcher defines the input image by specifying the size of 100x100 with 4 main layers. The first layer uses the Conv2D function with (filters 64, kernel_size 3x3, padding 'same'). The second layer uses the Conv2D function with (filters 64, kernel_size 3x3, padding 'same'), MaxPooling2D, and Dropout. The third layer uses the Conv2D function with (filters 64, kernel_size 3x3, padding 'same'). The fourth layer uses the Conv2D function with (filters 64, kernel_size 3x3, padding 'same'), MaxPooling2D, and Dropout. And Fully Connected layer with conditions (Dense 512, BatchNormalization, Dropout 25%, Dense 2). In each layer, the activation function used is the use of 'Elu' and 'Relu' and the optimization function used is 'SGD' and 'Adam'.

The fifth is Evaluation Result, which is looking at the test scenario by comparing the accuracy value of the model proposed in this study with the method proposed by previous researchers. The evaluation also looks at Plotting Accuracy, Plotting Loss, Classification Report, and Confusion Matrix Test. Plotting Accuracy and Plotting Loss will use a plotting technique using the pyplot library so that it will form a graph that can be seen and compared by researchers. The Classification Report will use techniques from the sklearn library by comparing the predicted model to the Test class so that the output will see the predictions (precision, f1-score, recall, accuracy) for the 2 class results in this model (Tuberculosis, Normal). Then the Confusion Matrix uses the Sklearn matrix technique so that it will form a table that is predicted to be True Positive, False Positive, False Negative, and True Negative. In the results of this evaluation, it can be seen how far the model can study the image well through the scenarios that have been built.

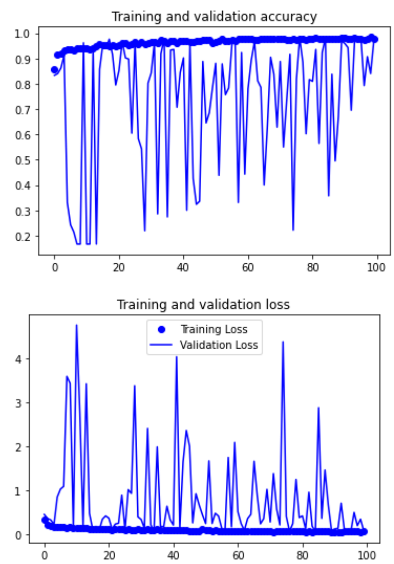
| | | Nilai Sebenarnya | |
|---|---|---|---|
| | | **True** | **False** |
| **NILAI PREDIKSI** | **TRUE** | TP | FP |
| | **FALSE** | FN | TN |

## 3. Results and Discussion

Researchers built 4 models with different architectures, the models built had been determined by the researchers, starting with the stages of the data collection process until the evaluation results. The differentiators in each model built are the activation functions in each layer 'elu' and 'relu', the use of data augmentation, and the optimizer functions 'SGD' and 'Adam'. From several architectural models that were built, the researchers analyzed the results based on the needs that have been achieved in each model.

### 3.1 Scenario Results Model 1

In the scenario stage, the first test model uses an augmentation process and does not use hyper parameter tuning and at the modeling stage, it uses the Adam optimizer function. When the model can be studied, the researcher then analyzes the model such as accuracy, precision, recall confusion matrix. The first model test has 329,282 parameters which consist of 4 convolutional layers. The first layer uses the 64 kernel with the Relu activation function. The second layer uses kernel 64 with Relu activation function and Maxpooling2D and 25% dropout. The third layer uses a 128 kernel with Relu activation. The fourth layer uses 128 kernels with Relu activation functions and Maxpooling2D and 25% dropout. Has 1 Fully Connected Layer of 512 with Relu activation function, Batch Normalization, 25% dropout, output layer using softmax activation function.

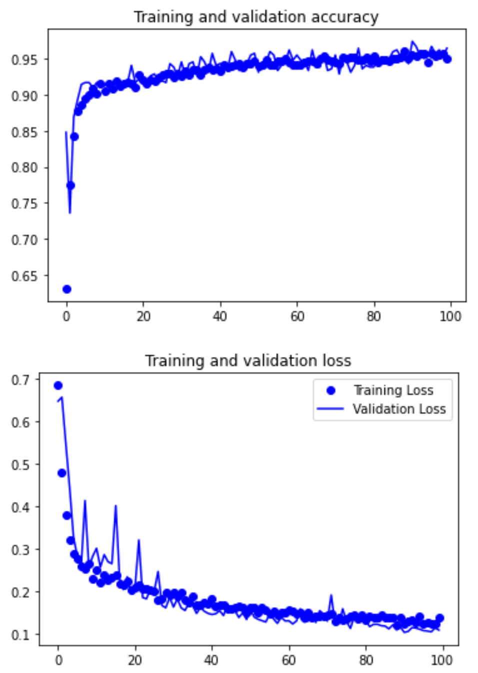


Classification Report

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Normal | 0.83 | 0.81 | 0.82 | 350 |
| Tuberculosis | 0.16 | 0.17 | 0.16 | 70 |
| accuracy | | | 0.71 | 420 |
| macro avg | 0.49 | 0.49 | 0.49 | 420 |
| weighted avg | 0.72 | 0.71 | 0.71 | 420 |

### 3.2 Scenario Results Model 2

At this stage is a test scenario with a CNN model that uses an augmentation process and uses hyper parameter tuning and the optimizer function in the modeling is SGD. In this model 2 scenario, analyze the model such as accuracy, precision, recall, and confusion matrix. In the first model scenario, it has 329,282 parameters consisting of 4 convolutional layers with the number of kernels 64(Conv-1 and Conv-2), 128(Conv-3 and Conv-4), and the kernel size used is 3 x 3, 2 max-pooling layer with a size of 2 x 2 then followed by 1 fully connected layers, namely FC-1 of 512, at the output layer using the softmax activation function. The activation function (Elu) as a hyper parameter tunning is used in all hidden layers except for the output layer. Dropout layers were added to layers 2 and 4 after max-pooling layers by 25% and fully connected layers for dropouts were used by 25%.

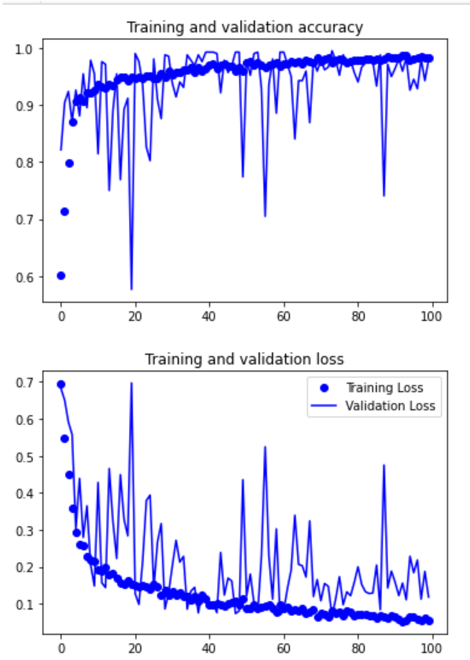



Classification Report

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Normal | 0.97 | 1.00 | 0.98 | 350 |
| Tuberculosis | 1.00 | 0.84 | 0.91 | 70 |
| accuracy | | | 0.97 | 420 |
| macro avg | 0.98 | 0.92 | 0.95 | 420 |
| weighted avg | 0.97 | 0.97 | 0.97 | 420 |

### 3.3 Scenario Results Model 3

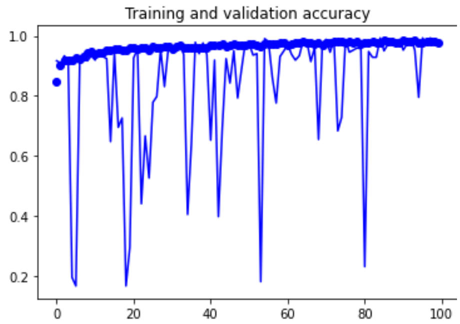
At the stage of the 3rd model testing scenario, it does not use the augmentation process but uses the method of using binary processing data consisting of gathering data with the provisions that the train data and test data are resized (100,100) and the training data is resized (3360, 100,100,3) , Train label ( 3360), Test data(420,100,100,3), Test label(420) data. Then enter the Normalization and Labeling Encoder stages. In this scenario, the model still uses the Hyperparameter Tunning method with the same provisions, namely the use of the 'elu' activation function with a total of 328,769 parameters consisting of 4 layers which are the same as the 2nd model scenario ending with 1 Fully Connected Layer with the activation function of the output layer being 'Sigmoid'. '.

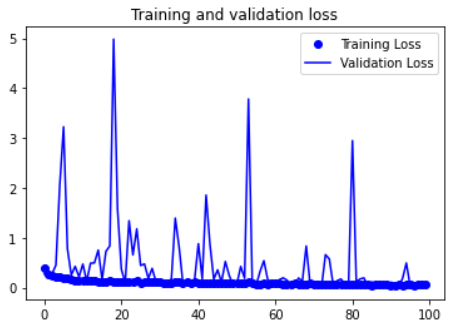



```
Model Kedua
              precision    recall  f1-score   support

           0       1.00      0.98      0.99       350
           1       0.90      0.99      0.94        70

    accuracy                           0.98       420
   macro avg       0.95      0.98      0.96       420
weighted avg       0.98      0.98      0.98       420
```

### 3.4 Scenario Results Model 4

At this stage of the fourth scenario, the test model uses an augmentation data process using an image data generator in accordance with the wishes of the researcher and the optimizer function used is Adam. The process of using the layer that will be applied to this model is the same as the implementation and implementation of the previous model, namely the first layer using the 64 kernel with the elu activation function. The second layer uses kernel 64 with elu activation function and Maxpooling2D and 25% dropout. The third layer uses a kernel numbering 128 with elu activation. The fourth layer uses a kernel totaling 128 with elu activation functions and Maxpooling2D and 25% dropout. Has 1 Fully Connected Layer of 512 with elu activation function, BatchNormalization, 25% dropout, output layer using softmax activation function.




```
Classification Report
              precision    recall  f1-score   support

      Normal       0.85      0.87      0.86       350
Tuberculosis       0.25      0.21      0.23        70

    accuracy                           0.76       420
   macro avg       0.55      0.54      0.54       420
weighted avg       0.75      0.76      0.75       420
```

### 3.5 CNN Model Performance Comparison

Researchers examined 4 models built with different architectures, researchers wanted the model built to produce better accuracy and efficient use of data in detecting Tuberculosis disease. From the four models that have been studied, it can be seen that the second model is the best model because hyperparameter tuning and the use of augmented data and the right optimizer function can reduce the amount of overfitting.

| ***Factor(s)*** | ***Values*** |
|---|---|
| Jumlah Convolutional + elu layers | 1,2,3,4 |
| Kernel Convolutional layer | 64, 128 |
| Jumlah dropout layers | 2,4 |
| Epoch | 100 |
| Pooling layers | Max Pooling |
| Batch_size | 32 |
| Dropout rate | 0.25 |
| Optimizers | SGD |

In the table above is the model used in previous journals which became the main reference for modeling and the researcher also tried modeling with the 'elu' data augmentation architecture. The 'elu' function in each layer(Conv1-4) with kernel_size in each layer(3,3), filters 64 for (Conv 1-2), filters 128 for (Conv 3-4), removing batchnorm in each layer and replaced batchnorm on Fully Connected Layer, Maxpooling on layer(Conv2 and Conv4). The dropout technique is found on the hidden layer (Conv2 and Conv4) and the dropout technique on the Fully Connected layer is 25%.

In the table below is a comparison of the performance of models 1 to 4 using the confusion matrix method that has been defined in each model and of course it has been calculated systematically against the model that has been built.

| **Performa (%)** | **Model Klasifikasi** | | | |
|---|---|---|---|---|
| | **1** | **2** | **3** | **4** |
| Accurcy | 70% | 97% | 97% | 72% |
| Precission | 81% | 100% | 97% | 84% |
| Recall | 83% | 96% | 99% | 82% |
| F1-Score | 81% | 97% | 97% | 82% |

From the table above by calculation using accuracy ($\frac{TP+TN}{TP+TN+FP+FN}$) , precission ($\frac{TP}{TP+FP}$) , recall ($\frac{TP}{TP+FN}$) , f1-score ($2 * \frac{(recall*preccsion)}{(recall+preccsion)}$) it can be concluded that the design of the 2nd and 3rd models is the best model. The second model uses hyperparameter tuning and an image data generator following the wishes of the researcher with the use of the SGD optimizer function and the graph plotting results show that it can suppress overfitting. While the 3rd model does not use an image data generator and only uses the same tuning hyperparameter as the previous model, resulting in an accuracy performance that is also almost close to the 2nd model, but when the model is running, there is overfitting between the train data and the test data so that even though the accuracy shown can match the 2nd model but cannot be said to be a perfect model. The 1st model is a modeling that is almost the same as the previous writing reference, namely the definition of layers by removing batch normalization in the main layer, using the 'relu' layer in each layer and using Adam in the optimization function. However, the first model has not been able to achieve the performance expected by the researchers. While the 4th model is a comparison model to the 2nd model with the use of augmented data and the 'elu' layer and in the optimization function there is a difference where the 2nd model uses 'SGD' and the 4th model uses 'Adam. From the results of the scenario that has been built the 4th model has not been able to achieve the desired performance of the researcher.

## 4. Conclusion

Based on the testing of several scenarios carried out by researchers by looking at the modeling structure built by previous researchers, several analyzes will be presented by this time researcher. First, the use of data augmentation/image data generator and the use of hyper parameter tunning ('elu') and the optimizer function 'SGD' can produce good performance and the graphs are shown in a stable condition. Second, the use of augmented data/image data generator with the same modeling structure by using the 'relu' function in each layer and the 'SGD' optimizer function does not produce maximum performance and the graph shows that the model built is in an overfitting condition. The third use of data augmentation/image data generator with the use of hyperparameter tunning('elu') and optimizer 'Adam' has not been able to produce optimal modeling.

Therefore, in determining the optimal modeling, it is necessary to pay attention to several modeling indicators that will be built and developed by researchers. From the various models that have been built by researchers that the use of augmented data greatly affects the results of modeling performance, in this condition without the use of augmented data shows that the modeling graph tends to be unstable compared to modeling using augmented data. The use of tuning hyperparameters such as the use of 'elu' and 'relu' layers greatly affects the modeling results.

## References

[1] W. Supartini and H. Hindarto, “Sistem Pakar Berbasis Web Dengan Metode Forward Chaining Dalam Mendiagnosa Dini Penyakit Tuberkulosis Di Jawa Timur,” *Kinetik*, vol. 1, no. 3, p. 147, 2016, doi: 10.22219/kinetik.v1i3.123.

[2] O. Rochmawanti, F. Utaminingrum, and F. A. Bachtiar, “Analisis Performa Pre-Trained Model Convolutional Neural Network dalam Mendeteksi Penyakit Tuberkulosis,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 4, p. 805, 2021, doi: 10.25126/jtiik.2021844441.

[3] C. Liu *et al.*, “TX-CNN: Detecting tuberculosis in chest X-ray images using convolutional neural network,” *Proc. - Int. Conf. Image Process. ICIP*, vol. 2017-Septe, pp. 2314–2318,

2018, doi: 10.1109/ICIP.2017.8296695.

[4] N. W. Septarini, “Metode Pengendalian Penyakit Menular,” *Fak. Kedokteran, Univ. Udayana*, pp. 121–125, 2017.

[5] B. P. Putra, Y. Yunus, and Sumijan, “Sistem Pakar dalam Mendiagnosis Penyakit Mata dengan Menggunakan Metode Forward Chaining,” *J. Inf. dan Teknol.*, vol. 3, pp. 128–133, 2021, doi: 10.37034/jidt.v3i3.122.

[6] A. Ahyuna and K. Aryasa, “Sistem Pakar Diagnosa Dan Tatalaksana Penyakit Demam Berdarah Dengue Menggunakan Jaringan Saraf Tiruan Backpropagation,” *E-JURNAL JUSITI J. Sist. Inf. dan Teknol. Inf.*, vol. 6, no. 1, pp. 1–11, 2017, [Online]. Available: http://ejurnal.dipanegara.ac.id/index.php/jusiti/article/view/40.

[7] R. Rahmadewi and R. Kurnia, “Klasifikasi Penyakit Paru Berdasarkan Citra Rontgen dengan Metoda Segmentasi Sobel,” *J. Nas. Tek. Elektro*, vol. 5, no. 1, p. 7, 2016, doi: 10.25077/jnte.v5n1.174.2016.

[8] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, “ChestX-ray: Hospital-Scale Chest X-ray Database and Benchmarks on Weakly Supervised Classification and Localization of Common Thorax Diseases,” *Adv. Comput. Vis. Pattern Recognit.*, pp. 369–392, 2019, doi: 10.1007/978-3-030-13969-8_18.

[9] D. Litchfield, L. J. Ball, T. Donovan, D. J. Manning, and T. Crawford, “Viewing another person’s eye movements improves identification of pulmonary nodules in chest x-ray inspection,” *J. Exp. Psychol. Appl.*, vol. 16, no. 3, pp. 251–262, 2010, doi: 10.1037/a0020082.

[10] N. Suhermi, S. Suhartono, I. M. G. M. Dana, and D. D. Prastyo, “Pemilihan Arsitektur Terbaik pada Model Deep Learning Melalui Pendekatan Desain Eksperimen untuk Peramalan Deret Waktu Nonlinier,” *Stat. J. Theor. Stat. Its Appl.*, vol. 18, no. 2, pp. 153–159, 2019, doi: 10.29313/jstat.v18i2.4545.

[11] I. M. Baltruschat, H. Nickisch, M. Grass, T. Knopp, and A. Saalbach, “Comparison of Deep Learning Approaches for Multi-Label Chest X-Ray Classification,” *Sci. Rep.*, vol. 9, no. 1, pp. 1–10, 2019, doi: 10.1038/s41598-019-42294-8.

[12] H. Liu, L. Wang, Y. Nan, F. Jin, Q. Wang, and J. Pu, “SDFN: Segmentation-based deep fusion network for thoracic disease classification in chest X-ray images,” *Comput. Med. Imaging Graph.*, vol. 75, pp. 66–73, 2019, doi: 10.1016/j.compmedimag.2019.05.005.

[13] G. Kim and C. Lee, *Convolutional Neural Network 를 이용한 한국어 영화평 감성 분석 Using Convolutional Neural Network*, vol. 2644. Springer International Publishing, 2016.

[14] D. A. V. Dyk and X. L. Meng, “The art of data augmentation,” *J. Comput. Graph. Stat.*, vol. 10, no. 1, pp. 1–50, 2001, doi: 10.1198/10618600152418584.

[15] J. Sanjaya and M. Ayub, “Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup,” *J. Tek. Inform. dan Sist. Inf.*, vol. 6, no. 2, pp. 311–323, 2020, doi: 10.28932/jutisi.v6i2.2688.

[16] B. Liu, X. Wang, R. Kwitt, and N. Vasconcelos, “Feature Space Transfer for Data Augmentation Mandar Dixit Microsoft,” *Cvpr*, pp. 9090–9098, 2018, [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/papers/Liu_Feature_Space_Transfer_CVPR_2018_paper.pdf.