ICOMTA: International Conference on Maritime Technology and Its Application
Vol. 3, April 2025, pp. 22 - 30
ISSN 3089-9125 online | doi.org/10.35991/icomta.v1i1.3

Detection of Screen Printing Result Using U-Net Convolutional Neural Network Method to Improve Quality Control

Ivan Nur Rahman[1], Edy Setiawan[2], Adianto[3]

Department of Automation Engineering, Shipbuilding Institute Of Polytechnic Surabaya, Indonesia

edy_setiawan@ppns.ac.id

Abstract. The increasingly competitive convection industry has compelled all enterprises to enhance production quality. One sector affected is the screen printing industry. According to information from one of the companies, customer complaints often arise from suboptimal screen printing results (defects). To achieve satisfactory outcomes, it is crucial to address this issue. Currently, the classification of result eligibility or quality control in screen printing relies on human observation. Pattern recognition technology is significantly transforming the convection industry, particularly within screen printing. This technology enhances quality control, shifting from manual processes to automated quality detection of results. Real-time pattern recognition employs image processing techniques. In this implementation, we utilize image processing with Convolutional Neural Networks for object classification., successfully identifying screen printing defects with an accuracy rate of 97%.

Keywords: Screen Printing, Pattern Recognition, Automated Quality Detection.

1. Introduction

The fabric screen printing process typically requires 2-3 workers; some are responsible for printing, others for moving the screen, and some for drying the screen printing ink. This process, from the screen printing stage to the drying stage, still relies on human or manual labor (Sahibuddin, 2020). In this process, screen printing often results in poor quality due to minimal quality control. It is deemed to have poor quality because the screen printing ink yields uneven results, stemming from inconsistencies during the screen printing stage and a lack of quality control. Classification of screen printing feasibility results is crucial for achieving high-quality products, maintaining marketing standards, and preserving product value (Lili Mulyaningsih1, 2023). A common issue in the t-shirt screen printing process is ensuring that the screen printing feasibility results meet established standards, particularly concerning screen printing defects. Currently, quality control is performed manually, relying on human observation, which can lead to errors and oversights by operators in the quality control of defective screen printing products.

Indications of product defects are identified through screen printing issues that lead to customer dissatisfaction. Based on findings from one of the conventions in Jombang city, it was observed that after the t-shirt printing process, the subsequent step involves quality control in the production table area, which is performed manually by the human eye. This reliance on manual inspection contributes to the presence of defective products that go undetected, as the quality control process still depends on human observation. Defects such as holes in the fabric, black spots following the screen printing process, and color mixing in screen printing can be classified as screen printing defects and can be identified by camera sensors to evaluate and standardize the quality of screen printing results. This research focuses on processing data from camera sensors with the goal of standardizing the quality of screen printing outcomes.

2. Methods

2.1. U-Net Convolutional Neural Network

U-Net is a type of CNN commonly used for image segmentation. U-Net includes a downsampler and an upsampler. The downsampling layer consists of two unfinished 3x3 convolutions, which pass through a ReLU activation function and a 2x2 maxpool layer with 2 steps. Feature duplication occurs at each downsampling step. The upsampling process consists of an upsampling process, a 2x2 convolutional layer that halves the number of features, a pooling process, and two 3x3 convolutional layers (each with a ReLU activation function). The last layer is a 1x1 transpose convolutional layer used to transform the 64 components of the feature vector into 3 classes.

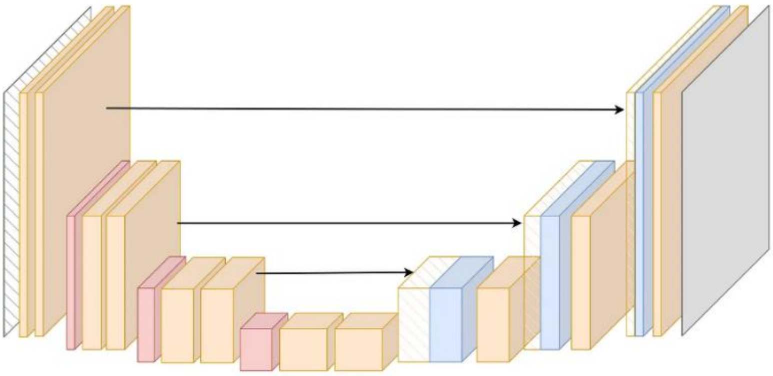


Fig. 1. U-Net Convolutional Neural Network Architecture

2.2. U-Net Convolutional Neural Network Architecture

a. Encoder

U-Net has a typical architecture with an encoder (characteristic processing part) and a decoder (reconstruction part). The encoder helps extract image features. The first convolution layer in the encoder is used to extract features from the input image. Each convolution layer can use filters to capture patterns and features at different levels of detail. After the convolution layer, an activation function such as ReLU (Rectified Linear Unit) is usually applied to introduce non-linearity into the model. Additionally, a reduction layer, often a pooling layer, is used to reduce the dimensionality of the image by taking the maximum or average value of a small region. Max pooling is commonly used to simplify feature representation. This layer helps retain the most significant features while reducing computational complexity, making the model more efficient in processing images. Decoder.

b. Bottleneck

Convolution and Activation Layer The convolution layer at the centre of the U-Net architecture acts as the bottleneck. Activation functions like ReLU are used after the convolution layer to introduce non-linearity.

c. Decoder

The decoder helps to reconstruct the image at a higher resolution and can be modified for classification tasks using U-Net. The Transposed Convolution Layer, also known as the deconvolution or upconvolution layer, increases the dimensionality of the image for better reconstruction. This is followed by a Concatenation Layer, where the result of the transposed convolution layer is combined with the corresponding layer's result in the encoder to integrate spatial and contextual information. After concatenation, another convolution is performed with an activation function like ReLU to extract features further, enhancing the model's ability to reconstruct or classify the input image accurately.

d. Output Layer

The final convolution and activation layer produces the segmentation output. The activation function at this layer is customized to the specific task, such as sigmoid for binary classification or softmax for multi-class classification.

3. Result and Analyze

3.1. Dataset

Before making the U-Net CNN model, the first and most important step is retrieving and collecting datasets. Here the researches makes 7 classifications such as burnt defects, missing letter defects, hole fabric defects, skewed defects, chipped defects, mixed color defects, normal. The following are the results of collecting data sets that have been made by the researches.

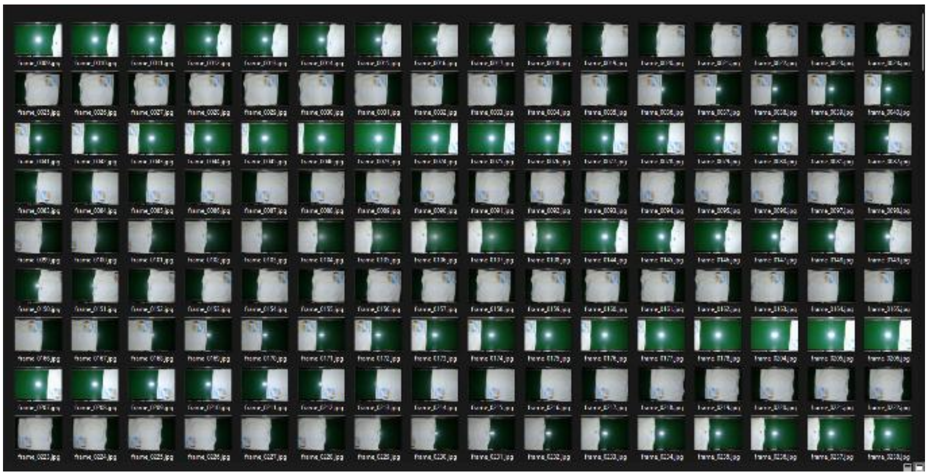


Fig. 2. Dataset

In this study, researchers created seven classes with the categories of burned defects, missing letter defects, perforated fabric defects, skewed defects, peeling defects, and mixed color defects, normal with a total image dataset of 4500 images, 3600 for training data, 450 images for test data, and 450 images for valid data. The image size is set with an image pixel value of 240 x 240 pixels. After that, the research divides the dataset into three parts, namely 80% training data, 10% test data, and 10% valid data. You can see the difference between normal and defective data at the following link https://bit.ly/4cRnks2.

Table 1. Dataset split table

No	Dataset	Split Percentage	Number of Split (Normal and Defective Data)
1	train	80%	3600
2	test	10%	450
3	valid	10%	450
	Total	100%	4500

3.2. U-Net Convolutional Neural Network Architecture

The determination of the architecture of the U-Net CNN model in this study aims to select the optimal image convolution model by considering the specifications of image pixels. The pixels utilized in this research are 240 x 240 pixels, along with several other parameters incorporated into the U-Net Convolutional Neural Network model. Figure 3 below illustrates the architecture of the CNN model that has been developed. The U-Net CNN model in Figure 3 represents the architecture employed in training and testing. In the architecture used, researchers extract U-Net features and subsequently combine them with CNN architecture for the classification process. The purpose of feature extraction is to enhance the recognition of objects in greater detail, followed by the CNN architecture for object classification.

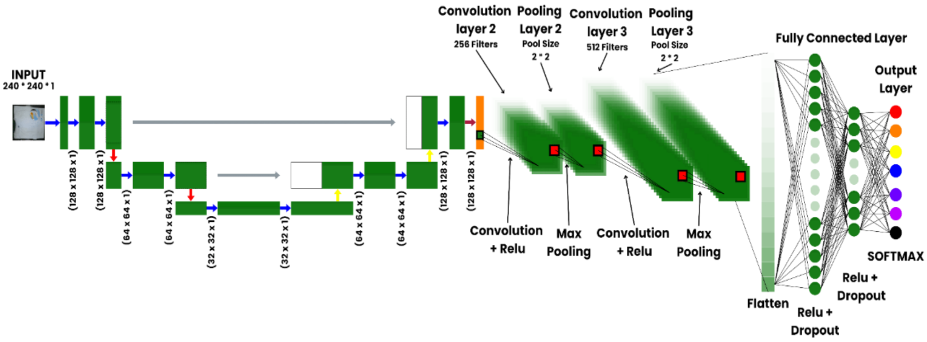


Fig. 3. U-Net Convolutional Neural Network Architecture

3.3. Training Model U-Net CNN

In the next stage, the training stage is tested. The training process is carried out when the image convolution process has been completed. This stage has a considerable influence on the accuracy graph of the architecture that has been made. The amount of training during the training process has been determined by the researches with the number of epochs of 80. To get optimal accuracy and accuracy of detection, the researches conducted research using Adam's optimizer. In testing, Adam's optimizer was used with the graph output model presented in Figure 4 below.

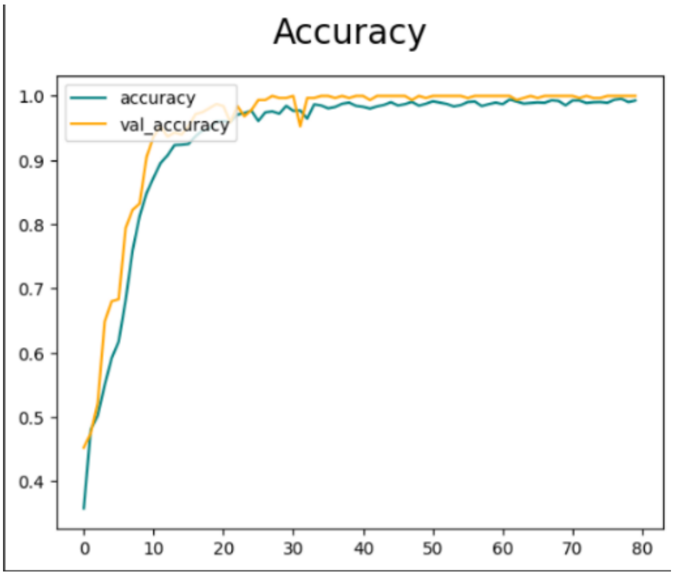


Fig. 4. Accuracy

Based on the U-Net CNN model accuracy results using the Adam optimizer in Figure 4, the training accuracy value is 1.0000 or equivalent to 100%, while the validation accuracy value is 0.9277 or equivalent to 92.77%. A graph pattern that tends to increase in the accuracy graph is obtained with a consistently increasing training accuracy graph pattern. Even so, several points still experience underfitting in validation accuracy, shown at point 0.9518 to point 0.8675, but after that, the graph managed to go back up to 0.9518. In addition to the accuracy graph, a Loss model graph with Optimizer Adam is presented in Figure 5 below.

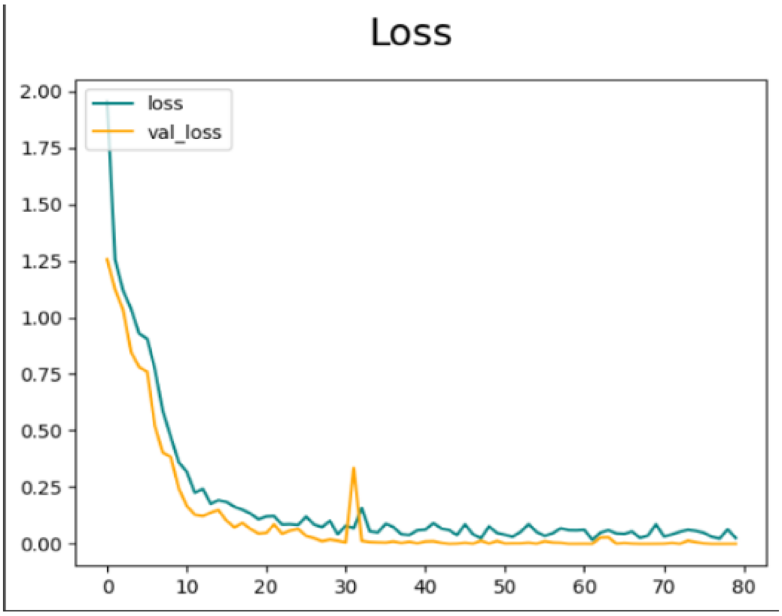


Fig. 5. Loss

Loss results of U-Net CNN models using Adam's optimizer in Figure 5 obtained a training loss value of 0.0055 or equivalent to 0.55%, while the validation loss value is 0.2113 or 21.13%. The loss graph using Adam based on training loss (red line) and validation loss (blue graph) from the structural line cumulatively tends to fall slowly. Based on this, the training loss graph is good because a good loss graph has a graph that tends to decrease steadily. However, the validation loss graph has reduced, but several points still experience underfitting in the validation loss, as shown at point 0.2585 to point 1.1806. Furthermore, we also examines the classification report indication from Adam's optimizer by displaying several parameters, as shown in Figure 6 below.

```
10/10 [==============================] - 27s 2s/step
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        28
           1       1.00      1.00      1.00        25
           2       1.00      1.00      1.00        29
           3       1.00      1.00      1.00        42
           4       1.00      1.00      1.00        41
           5       1.00      1.00      1.00        33
           6       1.00      1.00      1.00        34
           7       1.00      1.00      1.00        84

    accuracy                           1.00       316
   macro avg       1.00      1.00      1.00       316
weighted avg       1.00      1.00      1.00       316
```

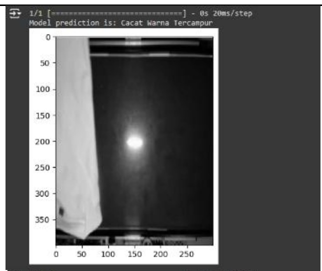
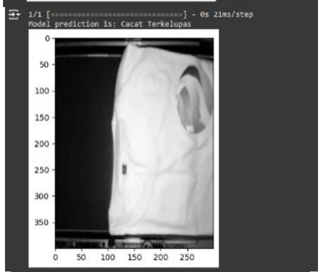
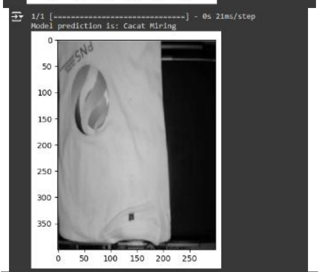
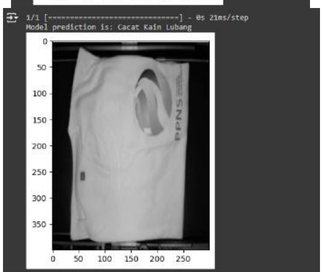
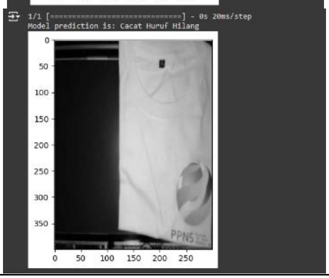
Fig. 6. Classification Report

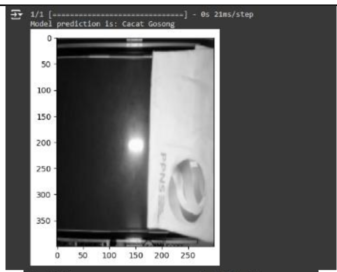
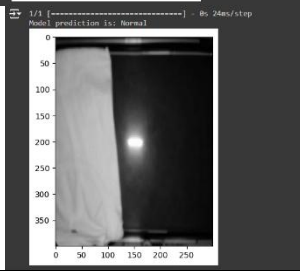
Based on the results in Figure 6, the defect and normal precision values are 0.95 and 0.91, respectively; the defect and normal data recall values are 0.90 and 0.95, respectively, and the defect and normal data f1-score values are 0.92 and 0.93, with a total defect support of 41 and normal support of 42.

3.4. Testing Model U-Net CNN

Testing the U-Net CNN model is carried out to try the accuracy of the results of the recognition and classification of defective or normal products, which are expected to get the best U-Net CNN model so that the detection classification decisions are optimized. In the test, experiments were carried out by inputting random images outside the dataset images that had been collected. In the testing stage, testing is carried out with the Adam optimizer. The data tested at the testing stage consists of 1 variation of standard data and six variations of defective data taken outside the dataset.

Table 2. Testing results with U-Net CNN method

No	Condition	Detect Results	Test Documentation	The results obtained
1	Mixed Color Defects	Mixed Color Defects		True
2	Chipped Defects	Chipped Defects		True
3	Tilt Defects	Tilt Defects		True
4	Hole Fabric Defect	Hole Fabric Defect		True
5	Missing Letter Defects	Missing Letter Defects		True

6	Burning Defects	Burning Defects		True
7	Normal	Normal		True

Based on the results of the testing phase of the model with Adam's optimizer, it is found that the classification of defective and normal data has no detection errors; in other words, Adam's optimizer has been able to classify the type of sample based on the defective and normal categories correctly. That way, testing the model with Adam's optimizer obtained a total classification error of 0 samples from 7 samples that have been tested, so Adam's testing accuracy is 100%. Based on the valid data, a total classification error of 14 samples from 450 samples of data that have been tested is obtained so that the accuracy of the train data is 97%. Based on the test data, 39 classification errors from 450 data samples have been tested, so the test data accuracy is 92%. In real-time testing, the system demonstrated a relatively short processing time for the U-Net CNN architecture in reading objects, specifically 3-5 seconds, while the system operated continuously.

3.5. Performance testing of U-Net CNN model system

After testing the testing data samples, here are the overall test results for model validation. The validation data consists of 450 data or 10% of the entire dataset, while the testing data also has 450 data or 10% of the whole dataset.

Table 3. Valid dataset test results with the U-Net CNN model

No	Class	Number of correct detections	Number of false detections	Error value
1	Burning Defects	59	1	1%
2	Missing Letter Defects	59	1	1%
3	Hole Fabric Defect	58	2	3%
4	Tilt Defects	58	2	3%
5	Chipped Defects	59	1	1%
6	Mixed Color Defects	56	4	6%
7	Normal	87	3	3%

Based on the results of model testing on valid datasets, it is known that for 450 random valid test data there are 14 prediction errors so that it has an error value of 3%. Based on the results of the error value obtained, it can be concluded that the model is able to identify and classify with a high accuracy value.

Table 4. Test dataset test results with the U-Net CNN model

No	Class	Number of correct detections	Number of false detections	Error value
1	Burning Defects	55	5	8%
2	Missing Letter Defects	56	4	6%
3	Hole Fabric Defect	53	7	11%
4	Tilt Defects	54	6	10%
5	Chipped Defects	56	4	6%
6	Mixed Color Defects	53	7	11%
7	Normal	84	6	6%

Based on the results of model testing on valid datasets, it is known that for 450 random valid test data there are 39 prediction errors so that it has an error value of 8%. Based on the results of the error value obtained, it can be concluded that the model is able to identify and classify with a high accuracy value.

4. Conclusion

1. The U-Net Convolutional Neural Network method effectively classifies screen printing defects. This U-Net CNN artificial neural network successfully identifies seven screen printing categories: Burned Defects, Chipped Defects, Missing Letter Defects, Normal, Mixed Color Defects, Tilted Defects, and Perforated Fabric Defects, achieving a success rate of 97% for valid data and 92% for test data.
2. The normal dataset achieves the highest percentage because its quantity exceeds that of the datasets for each defect classification. To enhance the classification success rate, researchers can increase the number of datasets for each defect classification.
3. In real-time testing, the system demonstrated a relatively short processing time for the U-Net CNN architecture in reading objects, specifically 3-5 seconds, while the system operated continuously.

References

1. Akbar Ganang Pradana1, D. R. (2024). Fine tuning model Convolutional Neural Network EfficientNet-B4 dengan augmentasi data untuk klasifikasi penyakit kakao. Journal of Information System and Application Development Volume 2 Number 1 , 01-11.
2. Dina Dwi Affifah*, Y. P. (2022). Teknik Konvolusi pada Deep Learning untuk Image Processing. Bandung Conference Series: Mathematics,, 103-112.
3. Farah Mufida Qotrunnada1, P. H. (2022). Metode Convolutional Neural Network untuk Klasifikasi Wajah Bermasker. PRISMA 5, 799-807.
4. Fino Charli1, H. S. (2020). Implementasi Metode Faster Region Convolutional Neural Network (Faster R-CNN) Untuk Pengenalan Jenis Burung Lovebird. Journal of Information Technology Ampera Vol. 1, No. 3,.
5. Guntara, R. G. (2023). Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7. Jurnal Teknologi Dan Sistem Informasi Bisnis Vol. 5 No. 1, 55-60.
6. Khairul Azmi 1, S. D. (2023). Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat. Jurnal Unitek Vol.16 No.1 Januari – Juni, 1273 - 1278.
7. Muhammad Zahir1, R. A. (2024). DETEKSI PENYAKIT RETINOPATI DIABETES MENGGUNAKAN CITRA MATA DENGAN IMPLEMENTASI DEEP LEARNING CNN. JURNAL TEKNOINFO Volume 18, Nomor 1, 121-132.
8. [8] Novelinda Permata Wulandari 1, D. F. (2021). Analisa Perbandingan Algoritma CNN Dan MLP Dalam Mendeteksi Penyakit COVID-19 Pada Citra X-Ray Paru. Sains, Aplikasi, Komputasi dan Teknologi Informasi Vol 3, No 2, 44-52.

9. Okta Saputra1, D. I. (2022). Implementasi Algoritma Convolutional Neural Network (CNN) Untuk Klasifikasi Senjata Tradisional Di Jawa Tengah Dengan Metode Transfer Learning. Jurnal Sistem Komputer dan Kecerdasan Buatan Volume 5 Nomor 2.
10. Putri, A. A. (2023). Convolution Neural Networks Untuk Deteksi Covid-19 Melalui CT Scan. Sains dan Teknologi Vol.2 No.1.
11. Richo1, R. Y. (2023). Eksplorasi Keandalan Sistem Sortir dan Klasifikasi Kecacatan Perekat Kemasan Menggunakan Arsitektur UNet-Inception Convolutional Neural Network. Jurnal Elkolind Volume 10, Nomor 3, 321 - 333.
12. Richo1, R. Y. (2023). nalisis Pengaruh Optimizer pada Model CNN untuk Identifikasi Cacat pada Perekat Kemasan. JURNAL SISFOTENIKA Vol. 13 No. 2, 217 - 229.
13. Rizal Fachmi, A. H. (2020). SISTEM IDENTIFIKASI UKURAN TUBUH MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN). TRANSIENT, VOL. 9, NO. 1.
14. Sahibuddin, A. A. (2020). QUALITY OF SCREEN PRINTING WITH PLASTISOL INK IN ANUKU SCREEN PRINTING STUDIO PARANG TAMBUNG MAKASSAR. KUALITAS CETAK SABLON DENGAN BAHAN PASTA PLASTISOL, 1-18.
15. Waliidaturrahmaniaha, H. ,. (2023). Identifikasi Stroke Menggunakan Metode Transfer learning Arsitektur Convolutional Neural Network Pada Citra CT-scan Kepala. PRISMA FISIKA, Vol. 11, No. 3 , 78 – 83.