Comparative Analysis Of Convolutional Neural Network Models For Digital Image-Based Melanoma Classification

Ana Kurniawati *[1], Aniqoh Hana Haura[2]
[1,2]Information System, Faculty of Computer Science and Information Technology, Universitas Gunadarma, Jl. Margonda Raya No.100, Depok 16424, Jawa Barat
e-mail: ***[1]ana@staff.gunadarma.ac.id**, [2]hana.haura06@gmail.com

Abstract

Melanoma is one of the most malignant forms of skin cancer, with an incidence rate of 7.9% in Indonesia. Traditional biopsy-based diagnosis, though crucial, is invasive and time-consuming, creating barriers for early detection. To address this issue, this research compares two Convolutional Neural Network (CNN) models for digital image-based melanoma classification. The study utilized a publicly available dataset from Kaggle, consisting of 17,805 images (melanoma and non-melanoma), which were divided into training, validation, and testing subsets. The models were trained using the Adamax and SGD optimizers for 100 epochs. The performance of the models was evaluated based on accuracy, loss, precision, recall, and F1-score. The CNN model with the best architecture, which consisted of two fully connected layers, achieved an accuracy of 93.18% and a loss of 0.1636, outperforming the alternative model. These results confirm the effectiveness of CNN models in classifying melanoma images and support the development of a web-based platform that allows users to upload or capture images for rapid and non-invasive detection.

Keywords— Convolutional Neural Network, Melanoma Skin Cancer, Website

1. INTRODUCTION

According to Global Burden Cancer (Globocan) data in 2020, the number of skin cancer cases in Indonesia reached 18,000 with approximately 3,000 deaths [1]. In Indonesia, the three most common types of skin cancer are basal cell skin cancer (65.5%), squamous cell skin cancer (23%), and melanoma skin cancer (7.9%) [2]. Melanoma is a skin cancer that develops in melanocytes, the skin pigment cells that produce melanin. Cancer-affected melanocytes can still form melanin, so melanoma usually appears brown or black. However, some melanocytes can no longer form melanin, causing melanoma to appear pink or white [3]. Early symptoms of melanoma are marked by the appearance of new moles or changes in the shape, size, and color of existing moles, accompanied by itching and possible bleeding. Body parts that often become locations for melanoma are the face, hands, back, and feet [2]. Melanoma is the most malignant type of skin cancer due to its rapid spread capability, even to internal organs.

Additionally, early symptoms resembling ordinary moles often make melanoma difficult for patients to recognize or detect directly with the naked eye. A skin cancer diagnosis can be performed through direct examination of the patient's skin, laboratory tissue testing (histopathology), or skin sampling (biopsy) [4]. Doctors typically perform biopsy examinations, but this procedure is time-consuming and invasive [5].

Computer vision technology can assist in rapid and non-invasive melanoma detection. Convolutional Neural Network (CNN) is a deep learning algorithm utilized to address computer vision problems such as classification, segmentation, and detection [6]. CNN enables computers to learn independently from data without explicit programming and extract features automatically [7]. CNN has high accuracy because it consists of layered networks that allow it to identify hidden features in image data that are not directly visible to the human eye. CNN is a development of

multilayer perceptron specifically designed to process data with grid topology, such as two-dimensional images [8].

Several studies have utilized Convolutional Neural Networks (CNNs) for melanoma skin cancer detection, each employing different architectures. In 2020, Jasman Pardede and Dwi Adi Lenggana Putra utilized the DenseNet121 architecture, which is known for its dense connectivity between layers, allowing for efficient feature extraction and reduced parameter numbers. However, one limitation of DenseNet121 is its computational complexity, which can lead to longer training times and higher resource consumption [2]. In 2022, Reynaldi Rio Saputro, Apri Junaidi, and Wahyu Andi Saputra performed melanoma classification using a CNN-based approach. Their model showed promising results but lacked a detailed exploration of hyperparameter optimization, which could potentially improve model performance [9]. In 2023, Muhammad Faris Fahru Rozi, Sri Mulyono, and Ghufron employed the MobileNet v2 architecture for Android-based melanoma detection, focusing on mobile devices. MobileNet v2 is known for its efficiency in mobile environments, offering a lightweight model with reduced computational overhead. However, a limitation of this architecture is its lower accuracy compared to more complex models like DenseNet121, especially when dealing with small datasets or highly varied input images [10]. These studies highlight the trade-offs between model complexity, computational efficiency, and accuracy in melanoma detection, which inform the approach taken in the present research.

In this research, two CNN models with different architectures were developed to obtain the best model. In recent years, Convolutional Neural Networks (CNNs) have been widely used for melanoma detection due to their ability to automatically extract complex features from images. This study compares two CNN models with different architectures to identify the most effective approach for melanoma classification. Both models were trained for 100 epochs, with the first model using an SGD optimizer and the second model employing an Adamax optimizer. The models were evaluated and tested on a dataset to select the best performer, based on performance metrics such as accuracy and loss. The model with the highest accuracy and lowest loss was implemented on the website. The website allows users to upload or capture images directly for melanoma detection, providing immediate prediction results and their associated probabilities. The development of the CNN model and the web interface was completed using Python, TensorFlow, and Streamlit. To ensure widespread accessibility, Streamlit Cloud was employed for the deployment process.

While this approach demonstrates the effectiveness of CNN models for melanoma classification, it is important to acknowledge the limitations. One key limitation is the dependency on high-quality input images, as the model's performance can degrade with poor image quality or lighting conditions. Additionally, while the CNN model performs well on the dataset used, the generalization to more diverse and larger datasets remains a challenge. The current state of the art in melanoma detection has seen the use of more complex architectures such as ResNet, VGG, and DenseNet, which offer improved accuracy but at the cost of increased computational resources. The novelty of this work lies in the combination of a CNN-based detection model with an accessible, user-friendly web interface, making early melanoma detection faster and non-invasive, while also prioritizing efficiency for mobile and web deployment.

2. RESEARCH METHODS

2.1. Research Design

This research follows a structured 5-stage process, illustrated in Figure 1. To provide better clarity, the stages have been reorganized into distinct sections.

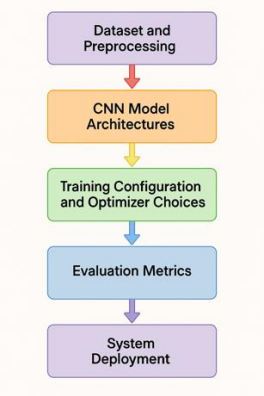
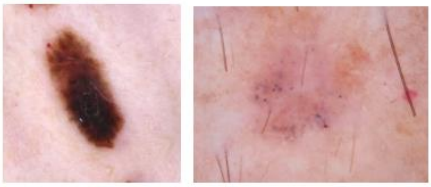


Figure 1. Research Design

1. Dataset and Preprocessing

The first stage involves determining the object and classification method. This study focuses on comparing two Convolutional Neural Network (CNN) models for digital image-based melanoma detection. The data used in this research is secondary data obtained from the Kaggle platform, which consists of 17,805 melanoma and non-melanoma images. The preprocessing stage standardizes pixel values and image sizes to ensure consistency across the dataset. This is achieved by rescaling and resizing images, which helps improve the model's learning efficiency. Each sub-dataset consists of two classes: melanoma class and not melanoma class, as shown in Fig. 2.



(A) Melanoma (B) Not Melanoma

Figure 2. Sample Data

2. CNN Model Architectures

The second stage is the development of two distinct CNN architectures. The two CNN models are designed to compare different architectural approaches and determine the best model for melanoma classification. The first model, referred to as **Model CNN-1FC**, consists of 4 convolutional layers, 4 max-pooling layers, 1 flattened layer, 1 dropout layer, and 1 fully connected layer. The second model, referred to as **Model CNN-2FC**, adds one more fully connected layer, resulting in a total of 2 fully connected layers. These models are built to vary in depth and complexity to assess their impact on classification accuracy.

The CNN architecture shown in Fig. 3 consists of two main parts: feature learning (or feature extraction) and classification. Feature learning is responsible for extracting features from input data, while classification functions classify input data into specific classes based on features learned by the model [6]. The feature learning section consists of convolutional layers, pooling layers, and ReLU activation functions. Meanwhile, the classification section includes flattening layers, fully connected layers, and softmax activation functions [11].

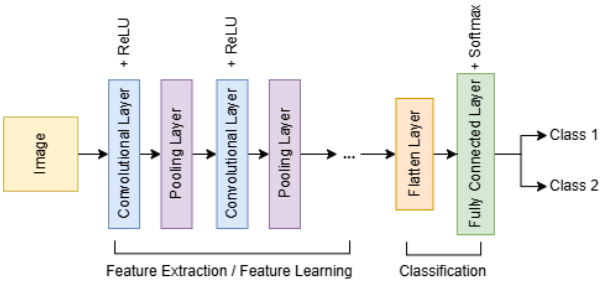


Figure 3. CNN Architecture

The convolutional layer functions extract features from images through convolution operations, which are dot product multiplications between image matrices and kernels. The result of the convolution operation is a two-dimensional matrix called an activation map or feature map, which represents features extracted from the image [12]. The pooling layer is a layer that performs downsampling operations or dimension reduction on the feature map. Its purpose is to speed up computation and reduce the risk of overfitting in the model. There are several pooling methods, including max pooling, which selects the largest value in a window, and average pooling, which calculates the average value in a window. The flattening layer functions transform feature maps from multidimensional matrices into one-dimensional vectors. This process is done so that image data can be processed in the fully connected layer [13]. The fully connected layer or dense layer is typically located at the end of the CNN architecture, functioning to classify input data into specific classes based on features extracted by previous layers. The final result of the fully connected layer is a vector containing probabilities for each class in the classified image [14]. Like the convolutional layer, the fully connected layer also performs dot product operations. The difference is that in the fully connected layer, each neuron is connected to all neurons in the previous layer [15]. Softmax is an activation function commonly used in output layers. This function plays a role in calculating probabilities for multi-class classification by selecting the class with the highest probability value. The output produced by the softmax activation function is probability values ranging from 0 to 1 [16].

3. Training Configuration and Optimizer Choices

The third stage involves training the two CNN models using the preprocessed dataset. The **Model CNN-1FC** is trained using the Stochastic Gradient Descent (SGD) optimizer, while the **Model CNN-2FC** uses the Adamax optimizer. Both models are trained for 100 epochs. The optimizer choices are made based on their suitability for image classification tasks. The training process ensures that the models can learn from the data and generalize well to new, unseen data.

4. Evaluation Metrics

In the fourth stage, the models' performance is evaluated using various metrics, including accuracy, loss, precision, recall, and F1-score [17]. These metrics are crucial for assessing the models' ability to classify melanoma images correctly. The evaluation process helps identify the best-performing model, which will be implemented for melanoma detection.

5. System Deployment

The final stages of the research involve deploying the selected CNN model onto a web-based platform. A website is developed to allow users to upload or capture images for melanoma detection. The deployment process is completed using Streamlit Cloud, ensuring that the platform is widely accessible to users. After developing the website, it undergoes local testing to ensure its functionality before being made publicly available.

3. RESULT AND DISCUSSION

3.1. Dataset and Data Collection

In this research, a secondary melanoma dataset obtained from the Kaggle platform (https://www.kaggle.com/datasets/drscarlat/melanoma) is utilized for training, validation, and testing purposes. The dataset consists of 17,805 images, divided into three distinct subsets: training, validation, and testing. Each subset includes two classes: melanoma and non-melanoma. The training dataset contains 10,682 images, equally divided between melanoma (5,341 images) and non-melanoma (5,341 images). The validation dataset consists of 3,562 images, with 1,781 images for each class. The test dataset has 3,561 images, with 1,781 melanoma images and 1,780 non-melanoma images, as summarized in Table 2. This balanced division ensures that each model can be trained, validated, and tested on a representative sample, minimizing the risk of bias in the classification process.

Table 2. Dataset

	Melanoma Class	**Not Melanoma Class**	**Total**
Training Dataset	5.341	5.341	10.682
Validation Dataset	1.781	1.781	3.562
Testing Dataset	1.781	1.780	3.561
Total			17.805

3.2. Preprocessing Data

Before training the models, the image data undergoes essential preprocessing steps to ensure consistency, uniformity, and improved model performance. Preprocessing helps standardize the input data, which is crucial for CNNs to perform effectively. The preprocessing steps include rescaling and resizing. Rescaling: This step standardizes pixel values across images to a fixed range, typically between 0 and 1. This is important because neural networks, especially CNNs, are sensitive to the scale of input data. Without rescaling, the network might struggle to converge during training, as it would require more time to adjust weights for inputs with varied scales. Resizing: All images are resized to a uniform spatial dimension of 224x224 pixels. This ensures that the model receives input images of the same size, which is required for CNNs. By resizing, we standardize the spatial dimensions of the images, preventing the model from being biased by images with different sizes or aspect ratios.

These preprocessing steps are implemented using the ImageDataGenerator class from the TensorFlow library. ImageDataGenerator allows for real-time data augmentation and modification during the training process without altering the original data, enabling more efficient and scalable training. The .flow_from_directory() method is used for automatic data labeling, as the images are stored in subdirectories where each folder represents a different class (melanoma or non-melanoma). The target_size parameter is set to (224, 224) to resize images to the required dimensions, and class_mode is set to 'categorical' for one-hot encoding of class labels. The batch_size is set to 64, meaning that the model processes 64 images at a time. For the test dataset, the shuffle parameter is set to False to maintain the original order of images, ensuring consistency in model evaluation and prediction.

Impact of Preprocessing:

a. Improved Convergence: By rescaling the pixel values and resizing images to a fixed size, the model trains more efficiently. This reduces the risk of issues like vanishing or exploding gradients and speeds up convergence during the training process.
b. Consistency Across the Dataset: Preprocessing ensures that all images are standardized in terms of pixel range and size, which is critical for the CNN to learn features effectively across all images.
c. Enhanced Performance: Data preprocessing, especially rescaling and resizing, has been shown to improve the performance of CNN models in tasks like image classification. By providing a

uniform input format, the model can focus more effectively on learning meaningful patterns and features, rather than being distracted by irrelevant variations in the input data.

Compared to other techniques, such as manual feature extraction or unmodified raw images, preprocessing significantly enhances model generalization and accuracy. Without preprocessing, the model would face difficulty in processing images with varying sizes and pixel values, leading to poor performance and slower training times.

3.3. Developing Two Distinct CNN Architectures

This research develops two CNN models with different architectures to obtain the best. **Model CNN-1FC**, consists of 4 convolutional layers, 4 max-pooling layers, 1 flattened layer, 1 dropout layer, and 1 fully connected layer, illustrated in Fig. 4.

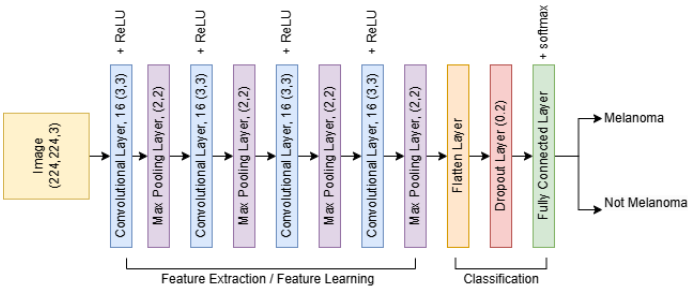


Figure 4. The Architecture of the Model CNN-1FC

Meanwhile, **Model CNN-2FC**, adds one more fully connected layer, resulting in a total of 2 fully connected layers., as in Fig. 5.

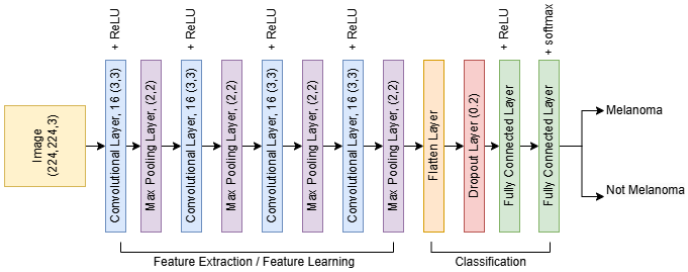
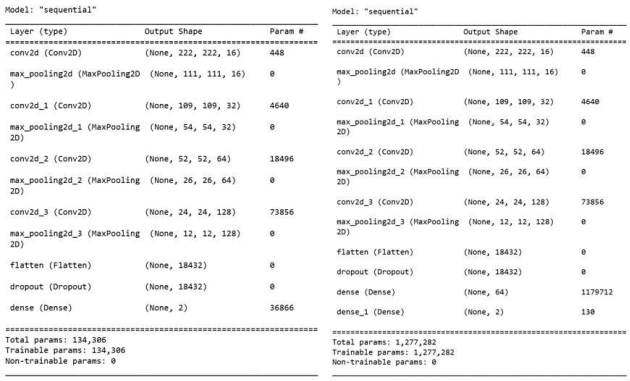


Figure 5. The Architecture of ModelCNN-2FC

The difference in the number of layers used in Model CNN-1FC and Model CNN-2FC affects the total features learned by both models. Model CNN-1FC learns 134,306 features in the data, while the Model CNN-2FC learns 1,277,282 features in the data, as shown in Fig. 6.



Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 16)	448
max_pooling2d (MaxPooling2D)	(None, 111, 111, 16)	0
conv2d_1 (Conv2D)	(None, 109, 109, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 32)	0
conv2d_2 (Conv2D)	(None, 52, 52, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 26, 26, 64)	0
conv2d_3 (Conv2D)	(None, 24, 24, 128)	73856
max_pooling2d_3 (MaxPooling2D)	(None, 12, 12, 128)	0
flatten (Flatten)	(None, 18432)	0
dropout (Dropout)	(None, 18432)	0
dense (Dense)	(None, 2)	36866

Total params: 134,306
Trainable params: 134,306
Non-trainable params: 0

(A) First Model

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 16)	448
max_pooling2d (MaxPooling2D)	(None, 111, 111, 16)	0
conv2d_1 (Conv2D)	(None, 109, 109, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 32)	0
conv2d_2 (Conv2D)	(None, 52, 52, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 26, 26, 64)	0
conv2d_3 (Conv2D)	(None, 24, 24, 128)	73856
max_pooling2d_3 (MaxPooling2D)	(None, 12, 12, 128)	0
flatten (Flatten)	(None, 18432)	0
dropout (Dropout)	(None, 18432)	0
dense (Dense)	(None, 64)	1179712
dense_1 (Dense)	(None, 2)	130

Total params: 1,277,282
Trainable params: 1,277,282
Non-trainable params: 0

(B) Second Model

Figure 6. Model Summary

3.4. Training the Two CNN Models

The next stage is training the two CNN models using the processed data. Before starting the training process, the models are compiled to configure training parameters, which include selecting optimization functions, loss functions, and model performance evaluation metrics. The optimization function plays a role in directing the model learning process toward optimal values. The loss function is used to measure the difference between values predicted by the models and actual values. Meanwhile, evaluation metrics are used to monitor and evaluate the models' performance during the training process.

This research applies different optimization functions to both models: Stochastic Gradient Descent (SGD) for the first model and Adamax for the second model. SGD was chosen for its simplicity and common usage. Adamax was chosen for its flexible adaptation mechanism, which potentially accelerates and stabilizes the learning process. Both models use the categorical cross-entropy loss function, suitable for classification problems with more than two classes, and accuracy metrics. To control the learning rate during training and address overfitting risks, the learning_rate_reduction variable is implemented using the ReduceLROnPlateau class from the Keras library. This class automatically adjusts the learning rate based on validation accuracy metric monitoring. If validation accuracy shows no improvement for two consecutive iterations, the learning rate will be reduced by 50% from its previous value. Both CNN models are trained for 100 epochs using 10,682 training data and 3,562 validation data that have been processed.

3.5. Evaluating the Training Results of the Two CNN Models

The evaluation of the two CNN models' training results, as presented in Table 3, is performed by analyzing accuracy and loss values on both the training and validation datasets. The learning curve is a graph showing changes in model accuracy and loss during the training process, consisting of accuracy graphs and loss graphs [18]. Learning curves also function to identify whether the model is in an underfitting, good fit, or overfitting condition [19]. In addition, learning curves are used to visually observe how accuracy and loss change as the number of training epochs increases. This evaluation helps measure each model's ability to learn patterns in the data and its capacity to generalize to unseen data.

While this study focuses on comparing two CNN models with simple architectures, it is important to acknowledge the potential benefits of experimenting with more complex, state-of-the-art CNN architectures. Popular CNN models such as LeNet, AlexNet, VGG, GoogLeNet, and ResNet have been extensively used in image classification tasks, including melanoma detection, and could provide significant improvements in performance. These architectures typically offer deeper networks, better feature extraction capabilities, and improved generalization compared to simpler CNN models.

Given their proven success in image classification, the use of these more advanced models could lead to more accurate and robust melanoma detection. Future work could explore the performance of these models in comparison to the simpler architectures used in this study. This would provide a more comprehensive evaluation of the CNN models' performance and could potentially enhance the results for melanoma detection.

In this study, however, the focus was to compare two basic CNN models to establish a baseline performance for melanoma detection. More complex architectures could be explored in subsequent experiments to better understand their impact on model accuracy, training time, and computational efficiency.

Table 3. Training Results of Model CNN-1FC

Epoch	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
10	0.7632	0.7670	0.4904	0.4906
20	0.8351	0.8321	0.3859	0.3829
30	0.8406	0.8380	0.3710	0.3713
40	0.8409	0.8431	0.3696	0.3700
50	0.8426	0.8403	0.3688	0.3706
60	0.8423	0.8450	0.3669	0.3681
70	0.8415	0.8445	0.3654	0.3661
80	0.8482	0.8419	0.3637	0.3649
90	0.8459	0.8414	0.3614	0.3640
100	0.8457	0.8450	0.3615	0.3623

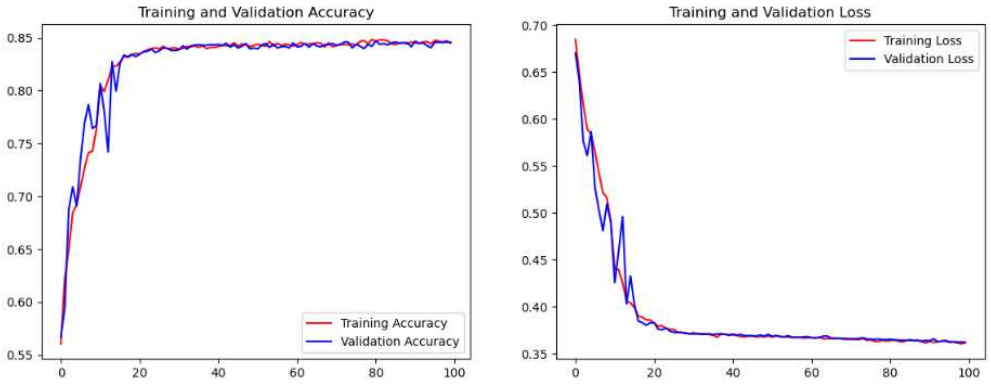


Figure 7. Learning Curve of The Model CNN-1FC

The Model CNN-1FC training results as in Fig. 7 show the model can learn patterns in training data well and demonstrates good generalization ability toward validation data. At epoch 100, the model achieves a fairly high accuracy of 0.8457 on training data and 0.8450 on validation data. Additionally, the learning curve shows accuracy improvement in both training and validation data, as well as decreased loss values in both types of data. This pattern indicates that the model has stable performance in training and validation processes.

Table 4. Training Results of The Model CNN-2FC

Epoch	Training Accuracy	Validation Accuracy	Training Loss	Validation Loss
10	0.9171	0.9208	0.2132	0.2014
20	0.9338	0.9281	0.1671	0.1896
30	0.9406	0.9371	0.1499	0.1592
40	0.9437	0.9312	0.1428	0.1655
50	0.9443	0.9287	0.1370	0.1667
60	0.9455	0.9385	0.1361	0.1569
70	0.9478	0.9391	0.1273	0.1540
80	0.9499	0.9385	0.1243	0.1528
90	0.9516	0.9413	0.1202	0.1490
100	0.9522	0.9410	0.1173	0.1486

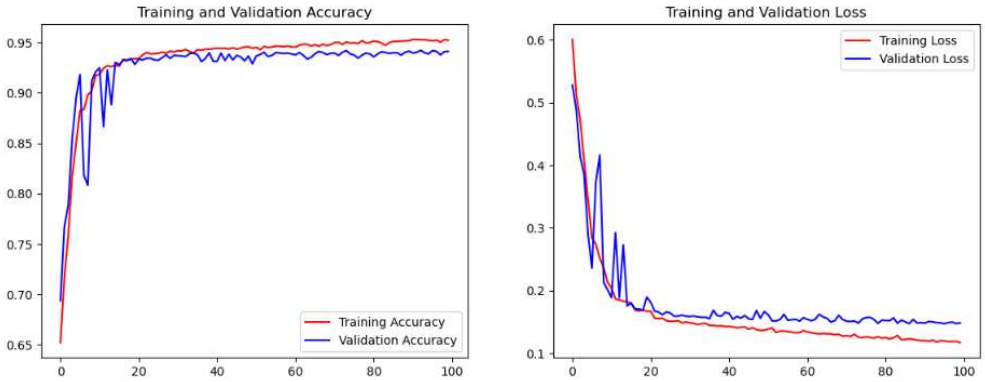


Figure 8. Learning Curve of The Model CNN-2FC

Model CNN-2FC in Fig. 8 also shows good model ability in learning patterns in training data and generalizing validation data well. At epoch 100, the model achieves a very high accuracy of 0.9522 on training data and 0.9410 on validation data. Additionally, the learning curve shows significant accuracy improvement in training and validation data, accompanied by a significant decrease in loss values. This pattern indicates that the model does not experience underfitting or overfitting problems. Therefore, in this research, no further adjustments to model hyperparameters were made.

3.6. Testing the Two CNN Models

The next stage is testing the two CNN models using 3,561 processed test data. This testing stage is conducted to evaluate the extent of the model's performance in predicting new data and representing their performance in real-world applications.

3.7. Comparing the Two CNN Models' Performance

Fig. 9 shows that the Model CNN-1FC successfully classified 1,504 melanoma data into the melanoma category and 1,436 non-melanoma data into the non-melanoma category correctly. However, the model misclassified 344 melanoma data into the non-melanoma category and 277 non-melanoma data into the melanoma category. Based on these classification results, other evaluation metrics were calculated. The precision value obtained for the melanoma class is 0.81, and for the non-melanoma class is 0.84. The recall value produced by the model is 0.84 for the melanoma class and 0.81 for the non-melanoma class. Furthermore, the calculated F1-score is 0.83 for the melanoma class and 0.82 for the non-melanoma class. These results indicate that the first model performs reasonably well, though there is still room for improvement, particularly in reducing false positives and false negatives.

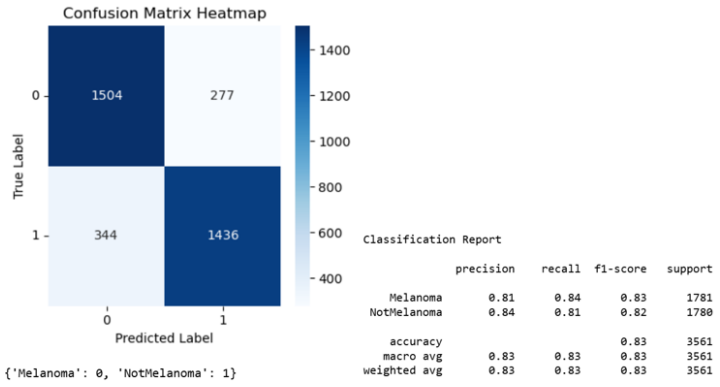


Figure 9. Testing Results of the First CNN Model

In contrast, the Model CNN-2FC achieved a better classification performance. It successfully classified 1,607 melanoma data into the melanoma category and 1,711 non-melanoma data into the non-melanoma category, as shown in Fig. 10. However, it misclassified only 69 melanoma data as non-melanoma and 174 non-melanoma data as melanoma. Based on these classification results, other evaluation metrics were also calculated. The precision value for the melanoma class was 0.96, and for the non-melanoma class was 0.91. The recall value for the melanoma class was 0.90, and for the non-melanoma class, it was 0.96. The calculated F1-score for both classes was 0.93. These results indicate that the second model outperforms the first model in terms of both precision and recall, suggesting a better ability to generalize to new data.

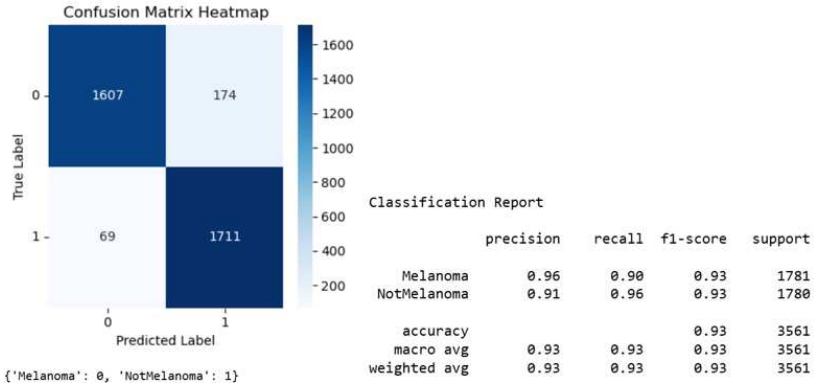


Figure 10. Testing Results of the Second CNN Model

The improved performance of the second model can likely be attributed to two main factors:

a. Addition of an Extra Fully Connected Layer: The second model includes an additional fully connected layer compared to the first model. This deeper architecture allows the model to learn more complex patterns and interactions within the data, which can lead to improved classification accuracy. The increased number of parameters enables the model to capture more detailed features from the input images, resulting in a better ability to distinguish between melanoma and non-melanoma classes.
b. Optimizer Choice (Adamax): The second model used the Adamax optimizer, which is a variant of the Adam optimizer that adapts learning rates based on the weight of each parameter. This optimizer can sometimes offer faster convergence and better generalization, especially in cases where the model has more complex architectures. In this case, the Adamax optimizer likely contributed to the better performance by effectively adjusting the learning rates during training, leading to improved accuracy and reduced loss.

The following is a performance comparison chart of two CNN model architectures, namely CNN-1FC and CNN-2FC. This chart presents a visualization of the accuracy and loss of each model during the training process, aiming to illustrate the learning effectiveness and overall performance of both models. The blue and green lines represent accuracy (left axis): Model CNN-2FC consistently shows higher accuracy compared to CNN-1FC. The red and dashed orange lines represent loss (right axis): Model CNN-2FC has lower loss values, indicating more effective learning. Overall, Model CNN-2FC has proven to deliver better performance in terms of both accuracy and loss efficiency compared to CNN-1FC illustrated in Fig. 11.

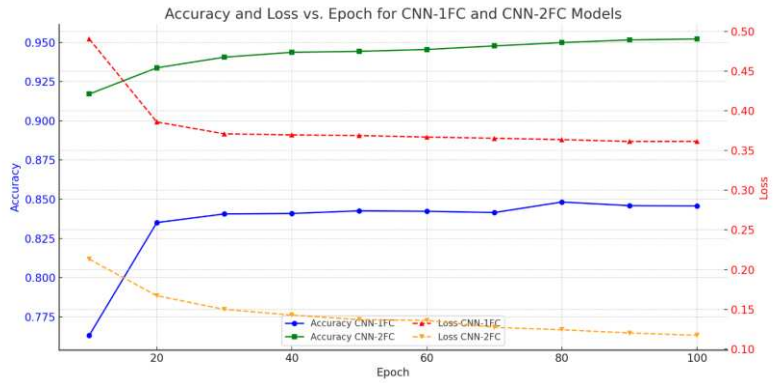


Figure 11. Accuracy and Loss Values of Two CNN Models

3.8. Drawing Conclusions from the Comparison Results

Based on a comprehensive evaluation comparing the two CNN models' performance, it can be concluded that both models have a good ability to predict new data. The second CNN model shows better performance compared to the first model, achieving 0.9318 accuracies and 0.1636 loss. Based on these results, the second CNN model was chosen to be implemented in the website. The implementation process requires saving the model so that weight and bias parameters, which are important information for the model, can be maintained and reused. In this research, the model is saved in .h5 file format.

3.9. Developing a Website And Implementing the Best CNN Model

Before developing the website and implementing the best CNN model, a thorough planning phase is essential. This stage ensures that the system built will not only meet user needs but also maintain high reliability and efficiency. The website is designed to provide an intuitive user interface, enabling users to upload or capture images for melanoma detection. The model is implemented using the best-performing CNN architecture identified in the previous stages, ensuring optimal accuracy in predictions.

To ensure smooth operation, the system was developed using Python, TensorFlow, and Streamlit. The deployment of the website was facilitated through Streamlit Cloud, allowing the model to be accessed and used on various devices. The focus of this development is on creating a user-friendly, efficient, and accessible platform for melanoma detection, leveraging the power of deep learning models while prioritizing ease of use.

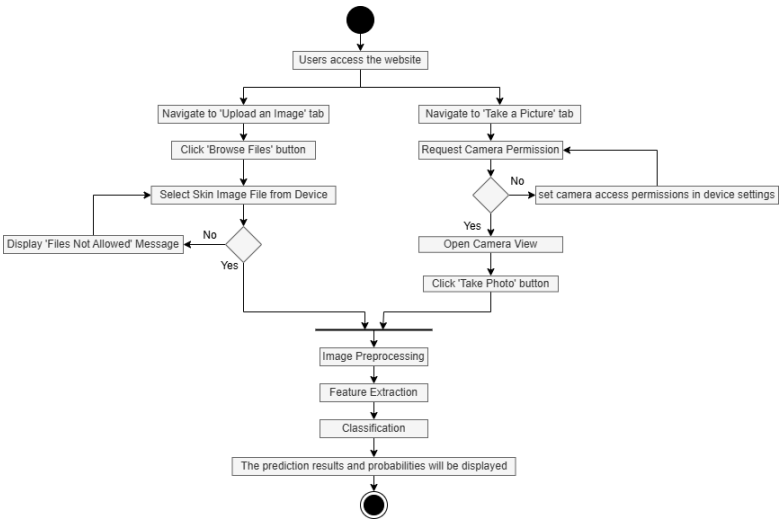


Figure 12. Activity Diagram

In this research, system design is carried out using UML diagrams, specifically activity diagrams, which are used to model system activity flows systematically and clearly [20]. The activity diagram in Fig. 12 illustrates interactions between users, the system, and other internal components to facilitate an understanding of image-based melanoma skin cancer detection system functionality. After accessing the website, users are given options to choose between an image upload feature or a direct image capture feature. The website allows users to either upload or capture images for melanoma detection. Users can upload an image from their device, and the system will validate the file format before processing it. Alternatively, users can take a picture using the website's camera functionality, with appropriate permissions requested for camera access. Once an image is selected or captured, it undergoes preprocessing, feature extraction, and classification using the trained CNN model. Subsequently, on the same web page, the system will display the model's prediction results along with their probability values. In this research, the website is built with Python programming language with Streamlit framework. The website interface design is shown in Fig 13.

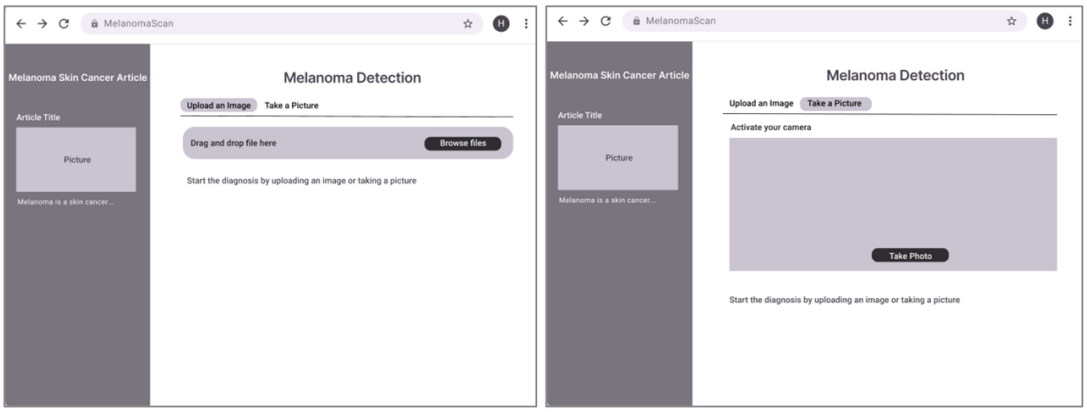


Figure 13. Website Interface Design

3.10. Testing the Website in a Local Environment

After developing the website and implementing the CNN model on the web, the next stage is functionality verification in the local environment. This process aims to ensure the performance of all features meets specifications and fulfills previously established objectives before being used by the public.

3.11. Deploying the Website

The final stage is deploying the website so it can be accessed by users generally through an appropriate platform. This research utilizes Streamlit Cloud as the deployment medium. The website display on a mobile device is shown in Fig 14 and on a desktop device is shown in Fig 15.

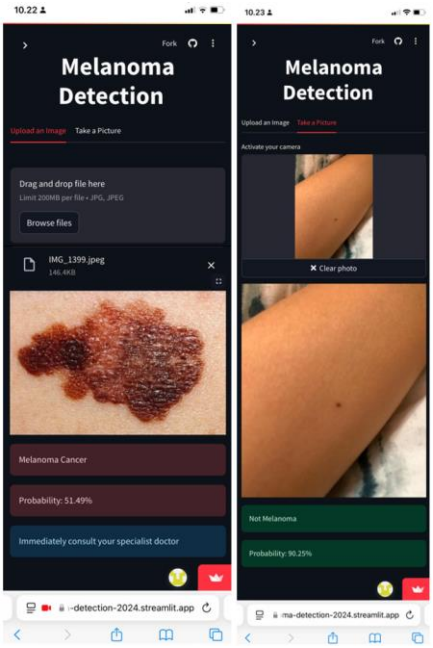


Figure14. Website Application Display on Mobile

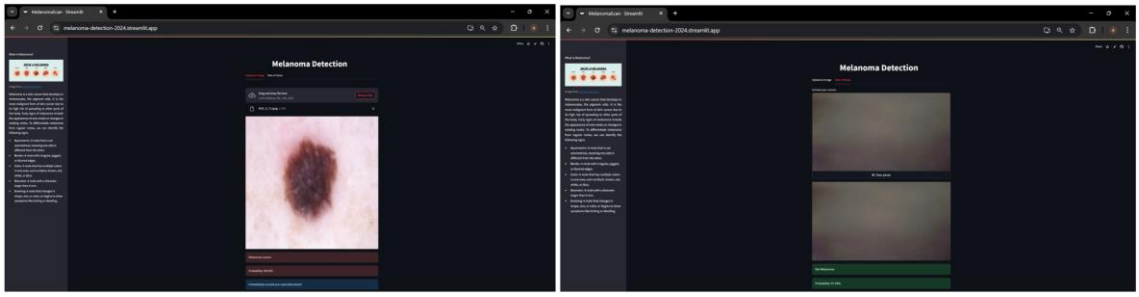


Figure 15. Website Application Display on Desktop

4. CONCLUSION

This research demonstrates the potential of Convolutional Neural Networks (CNNs) for digital image-based melanoma detection, with the second CNN model outperforming the first in terms of accuracy and loss. This model has been successfully integrated into a web-based platform, providing a user-friendly interface for melanoma detection through image upload or capture. The platform, hosted on Streamlit Cloud, offers accessibility across various devices, making it a valuable tool for early melanoma detection.

The results of this study highlight the promising role of deep learning in clinical settings, particularly for assisting with early skin cancer diagnosis. However, there are some limitations, including the reliance on high-quality images and the challenge of generalizing the model to diverse real-world datasets. Future research can focus on expanding the dataset to include more diverse images, improving model robustness, and integrating additional features such as saving prediction results and providing referrals to dermatologists or nearby healthcare facilities.

Further investigation into combining CNNs with other diagnostic tools and exploring the model's performance in real-world clinical environments could enhance its clinical value and applicability in assisting healthcare professionals in melanoma diagnosis.

REFERENCES

[1] A. R. Marietha, "18.000 Kasus Kanker Kulit Terjadi di Tanah Air, Berikut Daftar Tabir Surya Paling Favorit Indonesia," GoodStats, Feb. 26, 2024. [Online]. Available: https://goodstats.id/article/globocan-2020-18000-kasus-kanker-kulit-terjadi-di-indonesia-berikut-daftar-tabir-surya-paling-favorit-indonesia-JgXU9.

[2] R. R. Saputro, A. Junaidi, and W. A. Saputra, "Klasifikasi Penyakit Kanker Kulit Menggunakan Metode Convolutional Neural Network (Studi Kasus: Melanoma)," *Journal of Data Science, Information Technology, and Data Analytics,* vol. 2, no. 1, pp. 52–57, 2022, doi: 10.20895/dinda.v2i1.349.

[3] G. N. A. A. Paramartha, I. W. Niryana, and P. A. T. Adiputra, "Karakteristik Pasien Melanoma Maligna di Subbagian Bedah Onkologi RSUP Sanglah Tahun 2015-2016," *Intisari Sains Medis,* vol. 10, no. 2, Jun. 2019, doi: 10.15562/ism.v10i2.242.

[4] Q. Aina Fitroh and S. 'Uyun, "Deep Transfer Learning untuk Meningkatkan Akurasi Klasifikasi pada Citra Dermoskopi Kanker Kulit," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi,* vol. 12, no. 2, pp. 78–84, 2023.

[5] R. Yohannes and M. E. Al Rivan, "Klasifikasi Jenis Kanker Kulit Menggunakan CNN-SVM," *Jurnal Algoritme,* vol. 2, no. 2, pp. 133–144, 2022, doi:10.35957/algoritme.v2i2.2363.

[6] D. Gunawan and H. Setiawan, "Convolutional Neural Network dalam Analisis Citra Medis," *Jurnal Konvergensi Teknologi dan Sistem Informasi,* vol. 2, no. 2, pp. 376–3990, 2022, doi: 10.24002/konstelasi.v2i2.5367.

[7] A. A. Soebroto, I. Cholissodin, Sutrisno, U. Hassanah, and Y. I. Febiola, *AI, Machine Learning & Deep Learning (Teori & Implementasi).* Malang, Indonesia, 2019.

[8] H. Suryalim, K. R. R. Wardani, and H. Heryantro, "Analisis Optimizer pada Convolutional Neural Network untuk Meningkatkan Akurasi Pengenalan Wajah," Skripsi, 2022.

[9] J. Pardede and D. A. L. Putra, "Implementasi DenseNet Untuk Mengidentifikasi Kanker Kulit Melanoma," *Jurnal Teknik Informatika dan Sistem Informasi,* vol. 6, no. 3, pp. 425-433, 2020, doi: 10.28932/jutisi.v6i3.2814.

[10] M. F. F. Rozi and S. Mulyono, "Deteksi Kanker Kulit Melanoma Berbasis Android Menggunakan Convolutional Neural Network Arsitektur MobileNET v2," *Jurnal Transistor Elektro dan Informatika (TRANSISTOR EI),* vol. 5, No. 2, pp. 89-94, 2023.

[11] H. M. Romario, E. Ihsanto, and T. M. Kadarina, "Sistem Hitung Dan Klasifikasi Objek Dengan Metode Convolutional Neural Network," *Jurnal Teknologi Elektro,* vol. 11, no. 2, pp. 108, 2020, doi: 10.22441/jte.2020.v11i2.007.

[12] Purwono, A. Ma'arif, W. Rahmaniar, H. I. K. Fathurrahman, A. Z. K. Frisky, and Q. M. U. Haq, "Understanding of Convolutional Neural Network (CNN): A Review," *International Journal of Robotics and Control Systems*, vol. 2, no. 4, pp. 739–748, 2022, doi: 10.31763/ijrcs.v2i4.888.

[13] A. Yusuf, R. Cahya Wihandika, and C. Dewi, "Klasifikasi Emosi Berdasarkan Ciri Wajah Menggunakan Convolutional Neural Network," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* vol. 3, no. 11, pp. 10595–10604, 2019, [Online]. Available: http://j-ptiik.ub.ac.id.

[14] K. P. R. Wardani and L. Leonardi, "Klasifikasi Penyakit pada Daun Anggur menggunakan Metode Convolutional Neural Network," *Jurnal Tekno Insentif,* vol. 17, no. 2, pp. 112–126, 2023, doi: 10.36787/jti.v17i2.1130.

[15] A. Peryanto, A. Yudhana, and D. R. Umar, "Rancang Bangun Klasifikasi Citra Dengan Teknologi Deep Learning Berbasis Metode Convolutional Neural Network," *Jurnal Format,* vol. 8, no. 2, pp. 138–147, 2019. [Online]. Available: https://www.mathworks.com/discovery/convolutional-neural-network.html

[16] N. Agustina Purwitasari and M. Soleh, "Implementasi Algoritma Artificial Neural Network Dalam Pembuatan Chatbot Menggunakan Pendekatan Natural Language Processing," *Jurnal IPTEK,* vol. 6, no. 1, pp. 14–21, 2022, doi:10.31543/jii.v6i1.192.

[17] R. Magdalena, S. Saidah, N. K. C. Pratiwi, and A. T. Putra, "Klasifikasi Tutupan Lahan Melalui Citra Satelit SPOT-6 dengan Metode Convolutional Neural Network (CNN)," *Jurnal Edukasi dan Penelitian Informatika,* vol. 7, no. 3, pp. 335–339, 2021, doi:10.26418/jp.v7i3.48195.

[18] W. Astriningsih, "Identifikasi Multi Aspek Dan Sentimen Analisis Pada Review Hotel Menggunakan Deep Learning," Skripsi, 2023.

[19] M. A. Wirya, "Deteksi Penyakit Alzheimer Pada Citra Magnetic Resonance Imaging Menggunakan Ml Dengan Metode CNN," Skripsi, 2023.

[20] L. P. Sumirat, D. Cahyono, Y. Kristyawan, and S. Kacung, *Dasar-Dasar Rekayasa Perangkat Lunak.* Malang, Indonesia: Mazda Media, 2023.