

A CNN-Based Information System for Balinese Dance Classification with Hyperparameter Optimization

Nengah Widya Utami¹, Made Adi Paramartha Putra¹,
I Gede Juliana Eka Putra¹, Gabriel Avelino Sampedro²

¹ Faculty of Information Technology and Design, Primakara University, Indonesia

² Department of Computer Science, University of the Philippines Diliman, Quezon City, Philippines

Article Info

Article history:

Received May 22, 2025

Revised Jul 09, 2025

Accepted Aug 15, 2025

Keywords:

Convolutional Neural
Netowrk
Balinese Dance
Hyperparameter Optimization
Information System

ABSTRACT

Balinese dance classification presents challenges due to limited datasets, complex postures, and the lack of real-world implementation. Existing studies often focus on model development while overlooking deployment aspects. This research proposes a lightweight Convolutional Neural Network (CNN) designed for Balinese dance classification and compares it with MobileNetV2, ResNet50, and VGG16 using consistent training settings. Data augmentation was applied to enhance generalization, and training epochs were optimized based on model convergence. The proposed CNN achieved a validation accuracy of 99.00%, with a precision of 92.55%, recall of 89.88%, and F1-score of 91.1%, using only 590 thousand trainable parameters and the fastest inference time of 476 milliseconds. Although others pretrained model, MobileNetV2 slightly outperformed in some metrics, the proposed model offered a better tradeoff between performance and efficiency. The trained model was deployed in a web-based application, demonstrating practical usability. This work supports the preservation of Balinese dance through accessible and efficient AI integration.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Nengah Widya Utami,
Faculty of Information Technology and Design,
Primakara University,
Jalan Tukad Badung No. 135, Renon, Denpasar Selatan, Kota Denpasar, Bali 80226 Indonesia.
Email: widya@primakara.ac.id

1. INTRODUCTION

Bali is widely renowned not only for its stunning natural landscapes but also for its rich and diverse artistic and cultural heritage. Among the various expressions of Balinese art, traditional performing arts hold a central role in the daily lives of the Balinese people. One of the most prominent forms of this artistic expression is traditional Balinese dance, a creative outcome shaped by deep-rooted cultural norms and rituals. These performances are not only aesthetic showcases but also symbolic and often ritualistic acts that follow both tangible and intangible rules embedded in tradition [1]. Balinese dance has long served as a cultural hallmark that is actively preserved across generations. Despite its popularity, especially among Balinese youth, there remains a gap in understanding the symbolic meanings embedded in these dances, particularly those conceived by now-deceased master choreographers [2]. Each dance type is distinguished by various stylistic markers such as costume, makeup, and movement patterns. Visual elements like makeup and attire allow observers to infer the dance's theme and style, while choreographic patterns further reflect its distinctive identity [3].

Movement, as the primary expressive medium in dance, plays a crucial role in conveying aesthetic and narrative elements. Traditional Balinese dance comprises an intricate arrangement of movements that form a cohesive artistic performance [4]. The mastery of basic movement techniques is fundamental for learners to fully grasp and execute comprehensive dance compositions. These movements can be categorized into several types: *gerak maknawi* (gesture), *gerak murni* (pure movement), *gerak penguat ekspresi* (expressive reinforcement), and *gerak berpindah tempat* (locomotion). All are united under core principles of *agem*, *tandang*, *tangkis*, and *tangkep*, which serve as the aesthetic foundation of Balinese dance [5].

In the era of rapid technological advancement, particularly with the rise of information technology, there has been a concerted effort to digitally preserve and promote Balinese dance. This includes disseminating performances on social media platforms like YouTube, TikTok, Instagram, and Facebook, as well as archiving them in cloud storage systems such as Google Drive. Several studies have already explored technology-assisted preservation methods, including Android-based learning applications [6], 3D movement simulation apps [7], and augmented reality-enhanced dance learning tools [8].

As we transition into the era of the Fourth Industrial Revolution (Industry 4.0), Artificial Intelligence (AI) offers a novel and promising avenue for cultural preservation. Recent research has employed AI techniques, particularly Convolutional Neural Networks (CNN), to analyze and detect movement patterns in Balinese dance. These studies demonstrate the potential of AI-based systems to contribute significantly to the documentation and transmission of this cultural heritage [9]–[11]. In the past few years, numerous studies have explored the integration of AI in recognizing dance movement patterns, particularly in video-based analysis. These existing works can be grouped into several categories. First, several studies focus on algorithm evaluation for dance recognition, which utilize CNN, optical flow, and attribute mining to achieve high recognition accuracy [12], [13]. However, these studies largely remain limited to the implementation and evaluation of algorithms without developing accessible systems for end users. Furthermore, they still face challenges in addressing complex issues like occlusion, disjointed movement, and significant displacement.

Moreover, studies by [14], delve into pattern similarity detection in Balinese dance choreography, utilizing CNN architectures like AlexNet and distance-based metrics such as Euclidean distance. These works are insightful but constrained by limited datasets and the lack of automated pose detection techniques, which limits their practical application. Third, dataset development efforts, provide valuable video data classified by gender, type of dance, and camera angles. Nonetheless, these datasets are not yet paired with intelligent applications, making them underutilized in applied learning contexts [2]. Additionally, there are efforts in visual representation and segmentation, such as semantic segmentation of Pendet dance images using the MultiRes U-Net architecture. This approach, while innovative, is limited to a single dance type and focused only on model evaluation [15]. Meanwhile, immersive learning and simulation tools using AR/VR and 3D modeling, offer future potential in digital dance education but remain at a conceptual or early developmental stage and lack AI-based intelligent recognition capabilities. Given these limitations, this study offers the following contributions:

1. We investigated the performance of several CNN-based models in classifying Balinese Dance, with hyperparameter optimization.
2. We developed an smart application that enabled users to automatically detect the identity and movement patterns of Balinese dances through video.
3. We integrated skeleton tracking algorithms, specifically Mediapipe for real-time pose detection and automatic annotation of dance movements, addressing the lack of automation in prior works.
4. We designed a user-friendly platform to support self-learning for the general public, Balinese youth, and international audiences interested in Balinese cultural heritage.

The remainder of this work detailed as follows: Section II highlight the reserach method adopted in this work, whereas the results and discussion covered in Section III. Finally, the conclusion and future work direction pointed in Section V.

2. RESEARCH METHOD

In this proposed work, we developed an application that incorporates an AI model to identify Balinese traditional dances and concurrently displays specific information about each respective dance. To achieve this objective, various steps were meticulously incorporated throughout the research process. Figure 1 illustrates the overall methodology employed in this research.

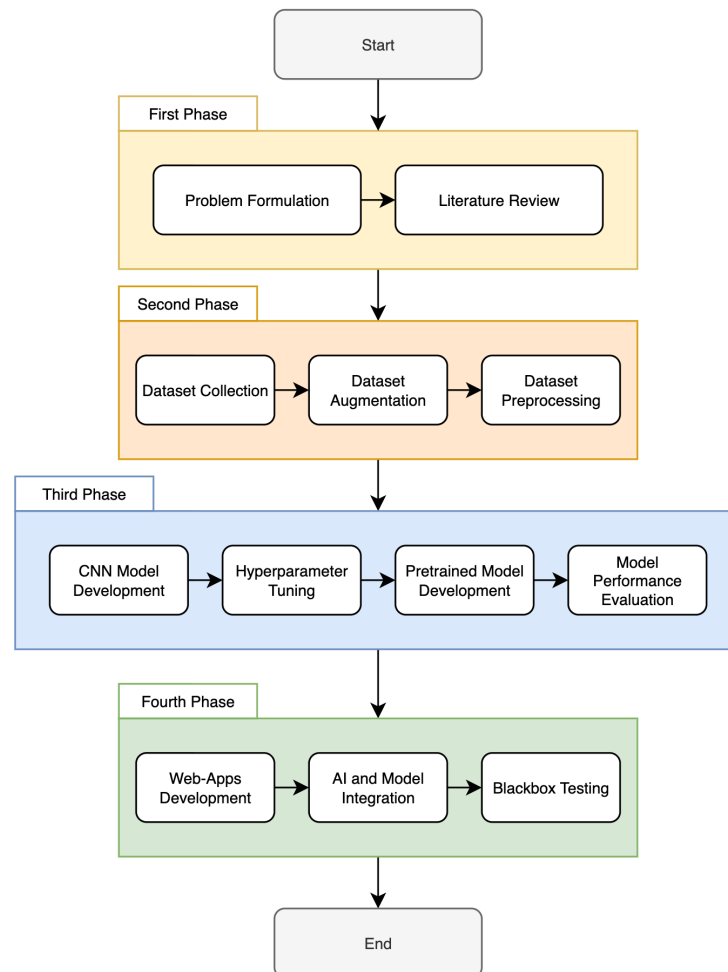


Figure 1. Overview of the methodology used in this research.

In general the process can be divided into three four phases. Each phase completion is mandatory before exploring the subsequence phase. In the initial phase, we start with problem formulation and literature review to ensure the current research work in this field. The next phase covers the dataset collection and preprocessing. In this stage we captures several dataset, while also utilizes our previous work data related to Balinese traditional dance. In the third phase, after the data is carefully collected and preprocessed, the model development process were conducted. In this stage, we created a CNN-based model while also evaluates pre-trained model, such as MobileNet [16], ResNet [17], and VGG [18] variations to shows the best performing model for our work. To further enhance the model performance, hyperparameter optimization were also conducted in this stage. Finally, in the last stage we develop the application for web-based application that utilized the best model obtained from the third phase. To ensure the system

performance is satisfactory, evaluation process were also incorporated in this final phase of development.

2.1. Dataset Collection and Preprocessing

Referring to the Balinese traditional dataset, the availability of publicly accessible data is relatively limited. Although there are numerous dance performances held in Bali for tourism purposes, the amount of data collected for AI model training remains insufficient. To address this issue, we conducted a data collection process that can be categorized into two types. The first type involved collecting images through online searches. Using this method, we were able to secure data for nine classes of Balinese traditional dances. While the number of classes is relatively high, the total number of samples remains inadequate. Therefore, the second type of data collection was conducted by capturing dance performances directly, organized by our research team. The camera setup and configuration used for this process are illustrated in Figure 2.

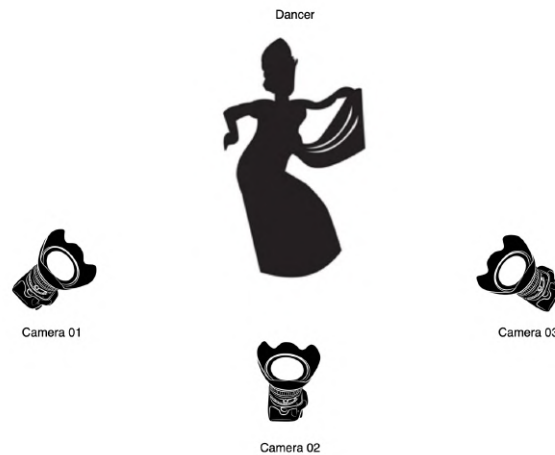


Figure 2. Dataset collection method using a multi-point-of-view approach.

Based on the given figure, three different angles were used to capture each movement of the dancer. This approach was adopted to ensure better data generalization. The dataset collection was carried out for all nine classes investigated in this study. Table 1 details the total number of data samples obtained from both image searches and direct captures to better illustrate the condition of the dataset.

Table 1. Total number of datasets used in this study.

Class	Online Collection	Manual Capture	Augmented Data
<i>Baris</i>	100	189	1156
<i>Barong</i>	100	191	1164
<i>Condong</i>	100	198	1192
<i>Janger</i>	100	194	1176
<i>Kecak</i>	100	148	992
<i>Pendet</i>	100	197	1188
<i>Penyambutan</i>	100	206	1224
<i>Rejang</i>	100	200	1200
<i>Truna</i>	100	197	1188
Total	900	1720	10480

Table 1 shows that the total number of images collected in this study was 2,620. The first dataset collection method yielded 900 images retrieved from various online sources. Additionally, the second method provided 1,720 images, obtained from nine different dance videos, each recorded from three different camera angles with an average duration of one minute per angle. For reproducibility, the dataset samples have been made publicly available and can be accessed via GitHub at <https://github.com/widya-utami/dataset-taribali>. In total, 27 videos were collected from the three camera perspectives. These videos were extracted at a rate of one frame per second. To further illustrate the diversity present in the dataset, which reflects real-world variations such as lighting conditions, occlusions, body shapes, costumes, and backgrounds, Figure 3 presents

representative sample images used in this work. Since the total of 2,620 images was still insufficient, it was necessary to include additional images in the dataset. To address this, a data augmentation process was applied. This process generates new data by modifying certain aspects of existing images, significantly increasing the total number of samples. As shown in Table 1, the final dataset size after augmentation was 10,480 images. This was achieved using four types of image modifications: flipping, zooming, rotation, and contrast adjustment.

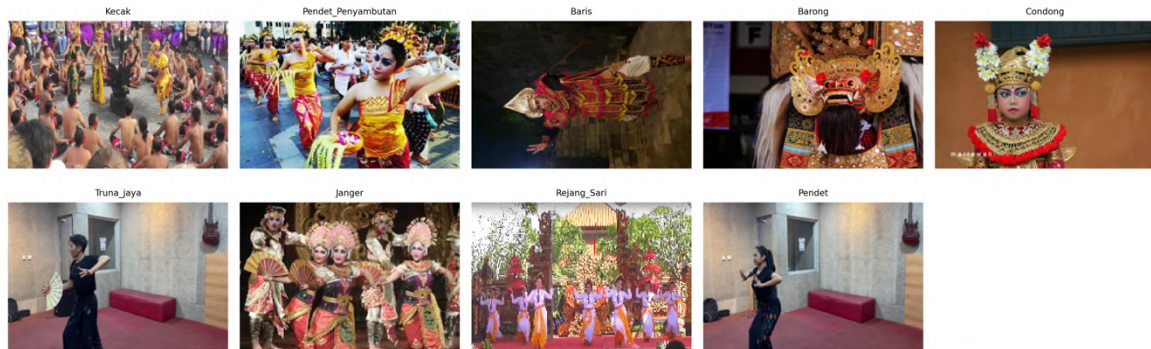


Figure 3. Sample of the dataset used in this work, collected from the web and through manual capture.

Furthermore, the dataset was split into training, validation, and testing sets based on the augmented data. We allocated 70% of the data for training, 20% for validation, and 10% for testing. This distribution ensures that the model trains effectively while being validated against a diverse set of samples. Before training, data preprocessing was performed to standardize the images. All images were resized to 128×128 pixels to maintain consistency, and rescaling was applied to normalize the dataset and further enhance training performance. Lastly, one-hot encoding was applied to convert the class names into numerical values for better interpretation.

2.2. Model Development and Optimization

The next steps carried out in this research are model development and optimization. Considering the image-based input used in this study, the model was developed using a CNN-based approach. The ability of CNN architectures to perform automatic feature extraction is highly effective for image classification tasks.

However, despite their strong performance, CNNs require careful tuning to achieve optimal results. This includes selecting the appropriate number of filters, kernel sizes, activation functions, and other layer parameters. In this study, we developed a custom CNN model that incorporates several Conv2D layers with specific configurations of filters and kernels. The proposed CNN model is illustrated in Figure 4.

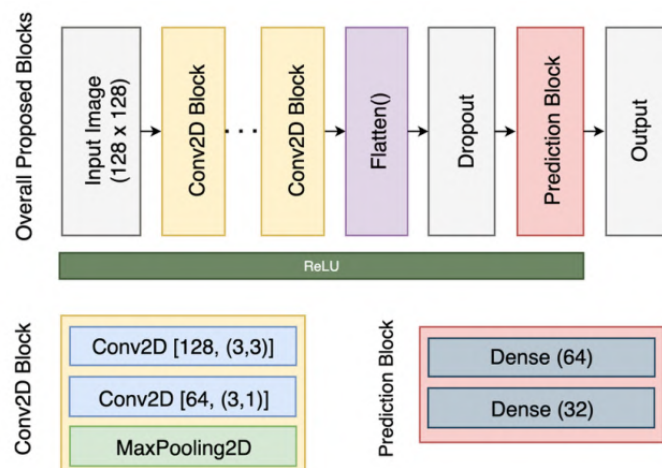


Figure 4. The architecture of proposed CNN model.

The proposed CNN model incorporates four CNN blocks, each consisting of two convolutional layers with a relatively low number of filters and small kernel sizes to speed up the feature extraction process, namely Conv2D (1,3) and Conv2D (3,1). After being processed by all blocks, the feature maps are passed through MaxPooling2D layers to reduce their spatial dimensions. Following the four CNN blocks, the model includes a flattening layer, a dropout layer with a rate of 0.5, two dense layers with 64 and 32 units respectively, and an output layer. These steps are essential to convert the extracted feature maps into a single array format suitable for prediction. It is worth noting that the activation function used in this work is the Rectified Linear Unit (ReLU), while the loss function is categorical cross-entropy with Adam optimizer. The softmax function is applied in the output layer to handle multi-class classification.

Relying solely on the proposed CNN architecture is not sufficient to achieve optimal performance; therefore, optimization techniques are necessary. In this work, we fine-tuned the model parameters using hyperparameter tuning. Three parameters were considered: learning rate, batch size, and the number of units in the dense layer. Grid search was applied to explore all possible configurations of these parameters. Although this process is time-consuming, the performance gains make the trade-off worthwhile.

Table 2. The value settings for hyperparameter tuning process.

Parameter	Minimal Value	Middle Value	Maximum Value
Learning Rate	0.001	0.01	0.1
Batch Size	8	16	32
Dense Unit	32	64	128

The list of parameters along with their value variations is presented in Table 2. Each parameter was assigned three different values, resulting in a total of 27 possible combinations. It is worth mentioning that the best model was selected based on the F1-score and model complexity.

To demonstrate the superior performance of the proposed model, several pretrained models were also considered in this study. Specifically, MobileNetV2, ResNet50, and VGG16 were the three pretrained models used. During model loading, the base models were frozen, and only the prediction layers were used for training process.

Furthermore, to compare the performance of the proposed CNN model with the three pretrained models, several evaluation metrics were used to identify the best-performing model. The evaluation began with model accuracy, calculated using Equation 1, followed by precision and recall, estimated using Equations 2 and 3, respectively.

$$Accuracy (\%) = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \times 100 \quad (1)$$

$$Precision (\%) = \frac{T_p}{T_p + F_p} \times 100 \quad (2)$$

$$Recall (\%) = \frac{T_p}{T_p + F_n} \times 100 \quad (3)$$

where T_p denotes true positive, T_n represent true negative, followed by F_p and F_n which refers to false positive and false negative, respectively. In addition to accuracy, precision, and recall, we also include the F1-score, which is calculated based on the precision and recall values, as shown in Equation 4.

$$F1 - score (\%) = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

In addition to performance, model complexity is also an important consideration when developing an application for Balinese dance classification. This is crucial because the inference

time required to process a single image must be minimized for practical use. Therefore, the inference time of each model was also investigated in this study.

2.3. Application Development

Most of previous studies covers traditional dance classification only focused on developing model and optimization process, while leaves the application side ignored. To provide and reach more impact of the development of the proposed AI model, we also developed a web-based application that can be used to classify 9 different classes of balinese dance.

We utilized waterfall methodology as the Software Development Life Cycle (SDLC) to develop the web-based application proposed in this work. The framework consist of several stages, including requirements, design, development, testing, deployment and maintenance.

First, for the requirements of the application, we consider the system able to support 9 different balinese dance, equipped with responsive design for different devices and ensure to provide high classification accuracy. For the next steps, the system were designed with simple landing page, where the user is able to start the detection immediately wihtout any requirement to logged in to the system. The server used in this work utilizes Flask framework to perform the backend process of the classification task. The development of the application conducted using NuxtJS framework for the frontend side, whereas Python for backend side.

Furthermore, after development phase completed, we move forward to the deployment phase which focused on evaluate the system in terms of functional and performance testing to ensure user satisfactory. After all evaluation passed, the developed application were then deployed to make it publicly available. We also conduct weekly maintenance to ensure the server is running well. To better illustrate the workflow of the website, the overall steps of user interaction and system response depicted in Figure 5.

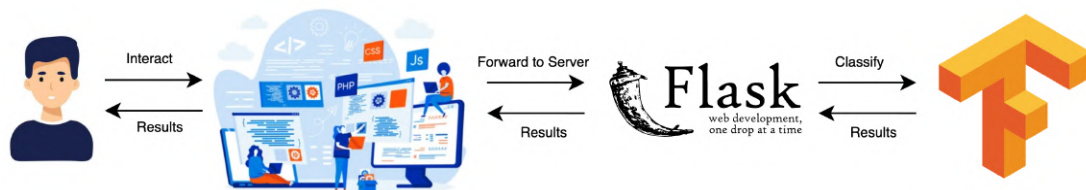


Figure 5. Workflow of the web-based application showing user interaction and system response.

3. RESULTS AND DISCUSSION

This section presents the results of the proposed work by considering various aspects, starting with the performance comparison between the proposed model and pretrained models, evaluated using multiple metrics. Additionally, the web-based application developed as the interaction platform was also assessed using black-box testing, focusing on its core functionalities.

3.1. Model Performance

The initial performance evaluation was conducted by assessing the proposed CNN-based model. The model was first trained on a limited dataset prior to the application of data augmentation. Subsequently, data augmentation techniques were applied, and the model was retrained to demonstrate the performance improvements resulting from the increased number of training samples. It is important to note that the dataset configuration remained consistent before and after augmentation to clearly highlight the benefits of the data augmentation process.

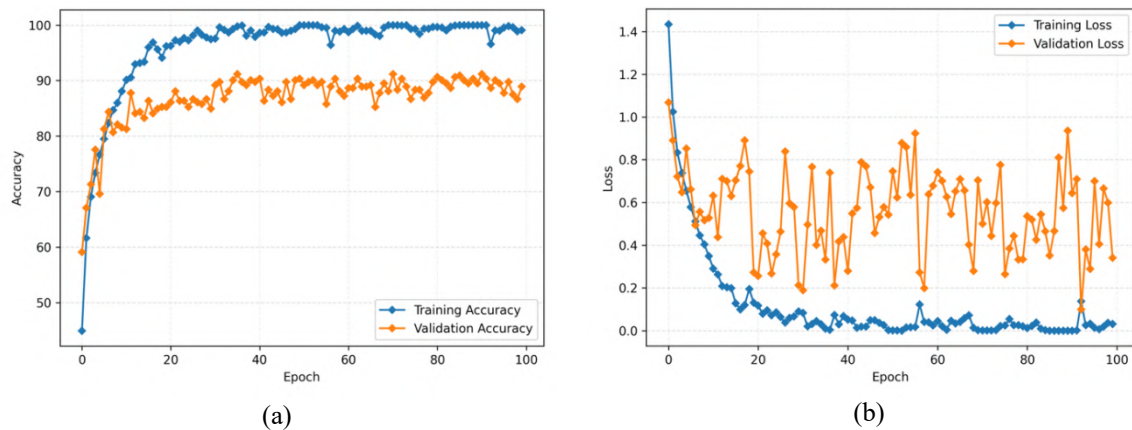


Figure 6. The performance of the proposed CNN-based model during training process
(a) Training accuracy over epochs, (b) Training loss over epochs

Figure 6(a) presents the training and validation accuracy of the model over 100 epochs. The training accuracy, represented by the blue line, improves rapidly, surpassing 95% within the first 20 epochs and reaching nearly 100% by epoch 30, demonstrating the model's strong learning capacity and ability to effectively fit the training data. The validation accuracy, shown by the orange line, also increases consistently during the initial epochs and reaches a stable plateau around 88 to 90 percent, reflecting strong generalization performance on unseen data. The narrow gap between training and validation accuracy during the early and middle stages of training indicates effective learning dynamics. Although a slight divergence is observed in the later epochs, the validation accuracy remains high, suggesting that the model maintains good generalization throughout training.

Additionally, Figure 6(b) shows the evolution of the training and validation loss. The training loss, indicated by the blue line, exhibits a steady and smooth decline, approaching near-zero levels, confirming the model's excellent capacity to minimize the objective function on the training set. The validation loss, represented by the orange line, displays greater variability but remains within a moderate range and does not show a consistent upward trend, indicating that the model does not suffer from severe overfitting. Occasional spikes in the validation loss may be attributed to batch-level variance or noise in the validation set, which is common in complex datasets. Overall, the relatively stable validation loss supports the model's robust performance across epochs. Based on the findings in Figure 5, the accuracy and loss curves demonstrate that the model learns effectively from the training data and achieves high validation accuracy, indicative of good generalization. The slight fluctuations in validation loss are not uncommon and do not undermine the model's overall robustness. These results validate the efficacy of the training process and confirm the suitability of the model architecture for the task.

To further demonstrate the performance of the proposed model in comparison with widely adopted architectures, we conducted an evaluation against three popular pretrained models: MobileNetV2, ResNet50, and VGG16. All models were implemented using the TensorFlow model library, and trained under identical conditions to ensure a fair comparison. The same dataset was used across experiments, and all models were trained for 100 epochs. In order to accelerate the convergence and reduce computational cost, transfer learning was applied to the pretrained models by freezing the early convolutional layers, while fine-tuning was performed on the deeper layers.

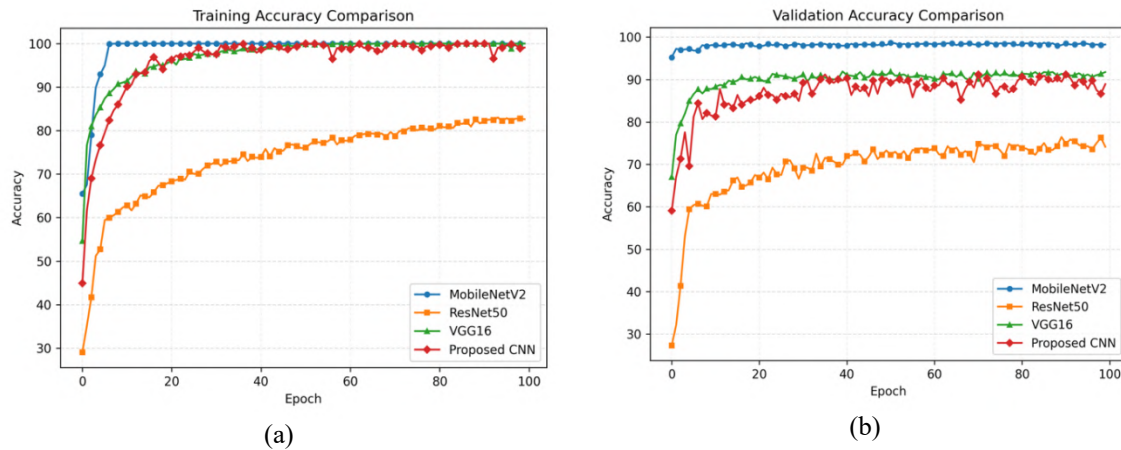


Figure 7. Performance comparison among the proposed CNN and pretrained model before the data augmentation process

(a) Training accuracy, (b) Training validation accuracy

Figure 7(a) presents the training accuracy comparison. As seen, MobileNetV2 reaches perfect training accuracy of 100% rapidly, stabilizing early in the training process. The proposed CNN model follows closely, also converging to near-100% accuracy by epoch 30. While MobileNetV2 converges slightly faster, the proposed CNN achieves nearly identical training performance, indicating that it is highly effective at learning from the data. Both models outperform VGG16, which converges more slowly, and ResNet50, which shows the weakest training performance, plateauing below 85%.

The validation accuracy shown in Figure 7(b) provides further insight into generalization. MobileNetV2 maintains a consistent validation accuracy of approximately 98%, the highest among all models. Importantly, the proposed CNN achieves validation accuracy in the range of 87% to 91%, closely trailing MobileNetV2. While a small gap remains, this still demonstrates strong generalization and robustness. Compared to VGG16, which yields similar performance to the proposed model, and ResNet50, which lags behind at around 75%, the proposed CNN clearly stands out as a competitive alternative to more complex pretrained architectures.

It is noteworthy to mention that the proposed CNN achieves this level of performance without the need for extensive pretrained weights, highlighting its effectiveness and efficiency in learning meaningful features from scratch. The proposed model demonstrates training and validation behavior that closely mirrors MobileNetV2, reinforcing its suitability for the task at hand. Additionally, despite the fact that MobileNetV2 shows slightly higher validation accuracy, the proposed CNN delivers comparable performance, both in learning speed and final accuracy. This positions it as a strong candidate in scenarios where model simplicity, customization, or independence from large-scale pretrained weights is desirable.

Although the proposed model initially exhibited promising results, it is important to acknowledge that these outcomes may lack full generalizability due to the relatively limited size of the original training dataset. Insufficient data can lead to overfitting, where a model fits the training data well but performs poorly on unseen samples. To address this concern and improve the model's generalization capability, a data augmentation strategy was employed. This process artificially increases the diversity and volume of the training set by applying random transformations to the input images, thereby helping the model learn more robust features.

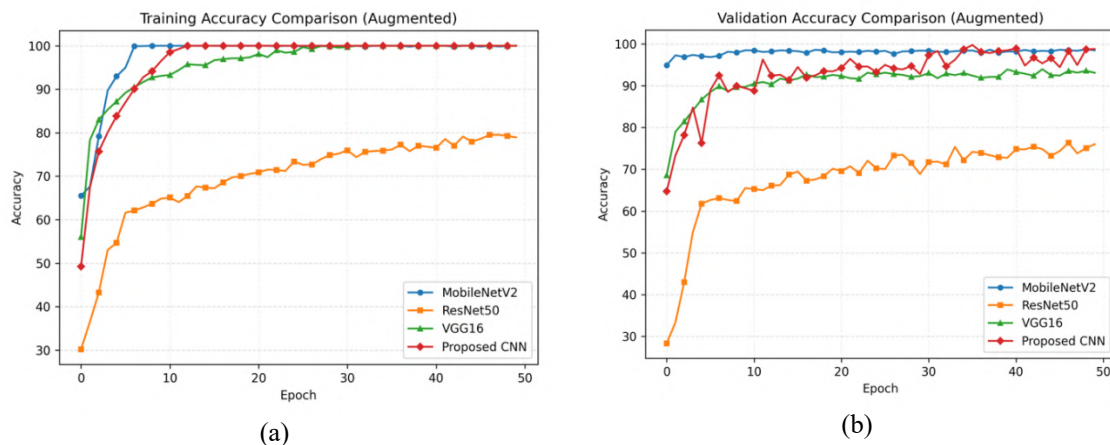


Figure 8. Performance comparison among the proposed CNN and pretrained model after the data augmentation process

(a) Training accuracy, (b) Training validation accuracy

Figure 8 displays the training and validation accuracy comparisons after the application of data augmentation, across four different models: MobileNetV2, ResNet50, VGG16, and the proposed CNN. Subfigure 8(a) illustrates the training accuracy, while the Subfigure 8(b) shows the validation accuracy over 50 epochs. It is important to note that the number of training epochs was reduced from 100 to 50 in this experiment, as preliminary results indicated that most models reached convergence well before the 50th epoch. This change also helps prevent unnecessary computation and reduces the risk of overfitting in later epochs.

From both subfigure, we can observe that data augmentation has positively impacted the training stability and generalization of all models. The proposed CNN achieves nearly 100% training accuracy and maintains validation accuracy consistently between 95% and 98%, indicating strong generalization. Notably, its performance closely aligns with that of MobileNetV2, a widely adopted lightweight pretrained model. Both models converge rapidly and maintain high validation accuracy throughout the training process. In contrast, ResNet50 continues to show lower overall performance, plateauing below 80% validation accuracy, while VGG16 stabilizes around 93%. This highlights the superior learning dynamics of both MobileNetV2 and the proposed CNN under the augmented training setup. The results shows that the use of data augmentation has not only enhanced model robustness but also validated the effectiveness of the proposed CNN architecture. Despite its simplicity and being trained from scratch, it achieves comparable, and at times superior, performance to MobileNetV2, while clearly outperforming VGG16 and ResNet50. These results affirm the model's suitability for the task, especially under data-constrained scenarios.

Table 3. Comprehensive model performance before and after the data augmentation process.

Model	Before Augmentation			After Augmentation		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
MobileNetV2	91.76	90.12	90.93	91.52	91.15	91.33
ResNet50	78.13	78.58	78.35	79.76	80.04	79.90
VGG16	88.92	86.67	87.78	89.84	88.42	89.12
Proposed CNN	90.48	89.64	90.06	92.55	89.88	91.19

In addition to overall accuracy, it is crucial to evaluate model performance using additional metrics such as precision, recall, and F1-score, especially when addressing classification problems. These metrics provide a more comprehensive view of the models' effectiveness in identifying relevant patterns across different scenarios. Table 3 presents the comparison of these metrics for four models, namely MobileNetV2, ResNet50, VGG16, and the proposed CNN under condition of before and after the data augmentation process.

To further evaluate the model performance beyond overall accuracy, precision, recall, and F1-score metrics were analyzed before and after applying data augmentation. As shown in Table 3, the proposed CNN demonstrated competitive performance, achieving an F1-score of 90.06% before augmentation, which is comparable to the pretrained MobileNetV2 that recorded 90.93%. In

contrast, VGG16 and ResNet50 showed lower performance, particularly ResNet50, which scored only 78.35%. These results suggest that the proposed model is well-optimized and effective even without data augmentation. After applying the augmentation process, all models showed performance improvements. The proposed CNN achieved a precision of 92.55%, recall of 89.88%, and F1-score of 91.19%, closely aligning with MobileNetV2, which recorded an F1-score of 91.33%. This indicates that the proposed model can generalize well, maintaining high accuracy while benefiting from a relatively simple architecture. The results validate the augmentation strategy and highlight the capability of the proposed CNN to match the performance of more complex pretrained models.

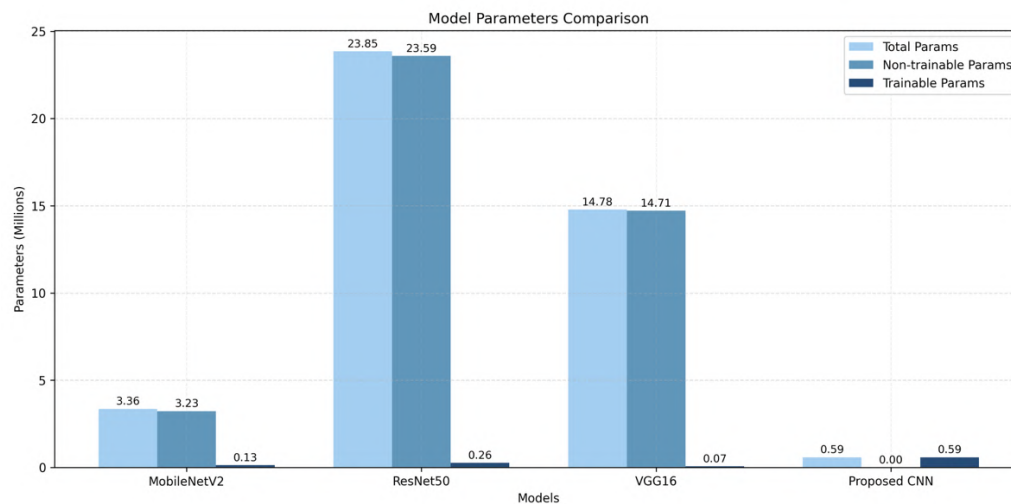


Figure 9. The comparison of total model parameters from different models investigated in this work.

While accuracy and performance metrics are essential indicators of a model's effectiveness, practical deployment also requires careful consideration of model complexity, particularly in applications where real-time or resource-constrained environments are involved. In such cases, a balance must be struck between achieving high accuracy and maintaining low computational overhead to ensure fast inference times. Therefore, evaluating the model complexity becomes an important aspect of assessing the overall utility of a deep learning model. Model complexity can be determined by the number of trainable and non-trainable parameters. These parameters directly influence the computational cost during training and, to some extent, during inference. Trainable parameters are those that are updated through backpropagation, affecting the learning capacity and the resources needed for training. Non-trainable parameters, on the other hand, typically result from layers or weights that are frozen (e.g., during transfer learning) and remain unchanged throughout training.

Figure 9 details the highest model complexity required by ResNet50 model with 23.85 millions parameters in total, followed by VGG16 with 14.78 millions parameters. These models require more memory and computational power, leading to longer training times and increased risk of overfitting, particularly with limited data. The other pretrained model evaluated in this work, MobileNetV2 maintains a much smaller parameter footprint with 3.36 millions, making it more efficient for both training and deployment. However, even MobileNetV2 includes a noticeable number of non-trainable parameters due to the use of pretrained weights with frozen layers. The proposed CNN stands out by having a relatively small total parameter count of 0.59 millions, with all parameters being trainable. This indicates that the model was designed from scratch, allowing for full adaptation to the dataset while keeping the architecture simple and efficient. The minimal number of non-trainable parameters also reflects the model's independence from transfer learning, giving it flexibility and lower memory requirements during training.

To better illustrate the performance of the proposed model compared with the other three pretrained models, the training and inference time were also investigated. Table 4 compares the training and inference times of four models: MobileNetV2, ResNet50, VGG16, and the proposed

CNN, measured both before and after the data augmentation process. The results clearly highlight the efficiency of the proposed CNN.

Table 4. Model performance before and after the data augmentation process.

Model	Before Augmentation (100 Epoch)		After Augmentation (50 Epoch)	
	Training Time (ms)	Inference Time (ms)	Training Time (ms)	Inference Time (ms)
MobileNetV2	2,696,000	750	1,470,000	750
ResNet50	2,862,000	1220	1,522,500	1220
VGG16	2,707,000	969	1,417,500	969
Proposed CNN	2,582,000	476	1,435,000	476

Before augmentation, the proposed CNN had the lowest inference time at 476 milliseconds, significantly faster than MobileNetV2 at 750 milliseconds, VGG16 at 969 milliseconds, and ResNet50 at 1220 milliseconds. Despite its simplicity, the proposed model also recorded the lowest training time among the models, indicating its computational efficiency during training. After augmentation, where the number of epochs was reduced to 25 due to faster convergence, the proposed CNN maintained its speed advantage with no increase in inference time and a notably reduced training time of 1,435,000 milliseconds, still lower than all other models. This reflects the model's ability to learn efficiently while maintaining fast deployment capabilities.

Based on the results, the proposed CNN is identified as the most suitable model for this work, primarily due to its efficient tradeoff between accuracy and computational complexity. While MobileNetV2 offers slightly higher accuracy in certain metrics, the proposed CNN consistently delivers competitive performance with significantly lower inference time and fewer trainable parameters, making it more efficient for real-time and resource-constrained environments. If inference speed, simplicity, and deployment efficiency are the main concerns, the proposed CNN stands out as the best choice.

Following the evaluation, the selected models including the proposed CNN and benchmark architectures were converted into .h5 format using TensorFlow. This step is essential for seamless integration into the next phase of web-based application development, where these models will be deployed within a real-time information system, enabling practical usage of the trained models for intelligent decision-making tasks.

3.2. Application Performance

In addition to evaluating model performance, this study also developed a web-based application to test the CNN-based architecture proposed in this work. As previously described, the development process followed the waterfall methodology. The requirements were gathered before proceeding to the design, development, and subsequent stages. The web-based application was successfully developed, with a snapshot of the web interface shown in Figure 10.

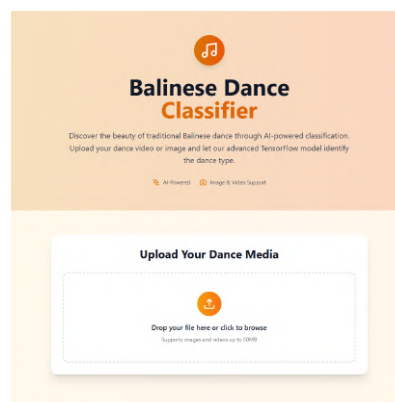


Figure 10. Landing page of the proposed web-based application for Balinese dance classification.

The landing page was designed to be as simple as possible to enhance user experience while using the proposed services. Users are allowed to upload both images and videos as model

input. For image input, the prediction results are displayed directly below the upload button. For video input, the system first checks the video duration. If the duration is equal to or longer than 5 seconds, up to five frames are randomly extracted. These frames are then used as image inputs for the model to generate predictions. The final classification result is determined based on the class with the highest frequency among the predicted frames. This method provides users with greater flexibility to classify various types of Balinese traditional dance.

To ensure the usability of the proposed web-based application, we conducted black-box testing to verify that all functionalities operate correctly. Since the core function of the website is to upload single images or videos for classification, the black-box testing focused on key aspects: upload success and prediction results for both image and video inputs.

Table 5. Performance Evaluation of the Proposed Web-Based Application

Test Category	Test Description	Expected Results	Status
UI/UX	Verify page loads completely	Page loads with all elements visible, no broken images/styles	Valid
UI/UX	Test system responsiveness	Layout adapts properly, where all elements remain readable and functional	Valid
Functionality	Test file selection via browse button	File dialog opens allowing image selection	Valid
Functionality	Verify classification result display	Result shows predicted dance type and confidence percentage	Valid
Input Validation	Test accepted image formats	All standard image formats are accepted	Valid
Input Validation	Test large image file handling	System handles large files appropriately	Valid
Performance	Test classification accuracy	Classification results only show results when accuracy is $\geq 80\%$	Valid
Performance	Test classification processing speed	Classification completes within 10 seconds	Valid

Table 5 outlines the test cases were categorized into four main areas, namely User Interface and User Experience (UI/UX), Functionality, Input Validation, and Performance. Each category encompassed a set of specific test descriptions, each paired with clearly defined expected results to ensure objective evaluation. The UI/UX category focused on assessing the interface's visual consistency and responsiveness under various usage scenarios. Functionality tests examined the system's operational behavior, such as whether users could successfully select files and receive classification results. Input validation evaluated the system's ability to handle different image formats and file sizes, while the performance category assessed how quickly and accurately the model processed input data. All functionalities were tested thoroughly and achieved their expected outcomes, with every test resulting in a "Valid" status, which signifies that the component functioned correctly and met the specified criteria.

The UI/UX tests confirmed that the application consistently maintained a clean and readable layout across devices and browsers, with no broken images or styling issues. The layout adapted responsively, ensuring all interface elements remained functional and visually accessible. Functionality tests validated that users could easily select image files using the provided browse button and that the classification results were displayed clearly, showing both the predicted dance type and the associated confidence score. Input validation confirmed that the system successfully accepted all standard image formats and handled large files without failure or slowdown. In terms of performance, the classification model demonstrated reliable accuracy, only presenting results when the prediction confidence reached or exceeded 80 percent. Moreover, the processing speed was sufficient to complete classification tasks within 10 seconds. Overall, this evaluation underscores the stability, efficiency, and user-friendliness of the proposed web-based application, making it suitable for practical deployment.

4. CONCLUSION

Balinese dance classification remains a challenging task due to limited available datasets, the complexity of traditional dance postures, and the lack of real-world implementations. Most previous research has focused primarily on developing high-performance models while neglecting deployment aspects, which limits the practical impact of these solutions. To address this gap, this study proposed a lightweight convolutional neural network specifically designed for Balinese

dance posture classification. The model was trained and evaluated alongside well-known pretrained architectures, including MobileNetV2, ResNet50, and VGG16, using the same dataset and training configuration. Additionally, a data augmentation strategy was applied to improve generalization, and the number of epochs was optimized based on convergence behavior.

Experimental results show that the proposed CNN model achieved a validation accuracy of 99.00%, with a precision of 92.55 %, recall of 89.88 %, and F1-score of 91.19 %. While MobileNetV2 performed slightly better in some metrics, the proposed model maintained comparable accuracy with significantly lower complexity, using only 590 thousand trainable parameters and achieving the fastest inference time of 476 milliseconds, outperforming ResNet50 and VGG16 by a wide margin. Furthermore, unlike prior work, this study successfully deployed the trained models into a web-based application, converting them into .h5 format for real-time use. The evaluation performed using blackbox testing also shows that the web-based application developed in this work, successfully provides Balinese dance classification with high accuracy and fast inference time. This practical implementation demonstrates the feasibility of using lightweight deep learning models in cultural heritage applications, offering a meaningful contribution toward preserving and promoting traditional Balinese dance through accessible technology.

We acknowledge that this method provides a simplified solution for video-based dance recognition, it does not fully capture the temporal dynamics inherent in dance movements. This limitation may affect the model's ability to understand motion continuity and complex transitions over time. As a direction for future research, we suggest exploring deep learning architectures that are specifically designed to model temporal dependencies, such as 3D Convolutional Neural Networks (3D CNNs), Long Short-Term Memory networks (LSTMs), and Transformer-based models for video analysis. So that the future work will focus on expanding the dataset, incorporating more dance categories, and enhancing the usability and intelligence of the deployed system.

ACKNOWLEDGEMENTS

This research was supported by Directorate of Research and Community Services, Primakara University, Bali, Indonesia.

REFERENCES

- [1] M. W. A. Kesiman, I. M. D. Maysanjaya, I. M. A. Pradnyana, I. M. G. Sunarya, and P. H. Suputra, "Profiling Balinese Dances with Silhouette Sequence Pattern Analysis," in *2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)*, 2020, pp. 423–428.
- [2] I. P. P. B. Lantara, I. P. D. Payana, G. A. S. Pratama, W. E. A. Munayana, K. S. Nopiani, and I. N. R. Hendrawan, "GETARI: dataset untuk Klasifikasi Gerakan Dasar Tari Bali Perempuan," *J. Nas. Pendidik. Tek. Inform. JANAPATI*, vol. 11, no. 3, pp. 290–300, 2022.
- [3] N. M. R. Maryati, M. W. A. Kesiman, and I. M. G. Sunarya, "Deteksi Persamaan Pola Gerakan pada Koreografi Tari Bali," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 5, pp. 1015–1026, 2023.
- [4] N. M. P. Erawati, "Mengenal Ragam Gerak Dan Jalinan Estetika Tari Bali," *Widyadari*, vol. 19, no. 2, 2018.
- [5] K. I. Parasari, I. K. Sariada, and N. N. M. Suryani, "Edukasi Dasar Tari Bali Perempuan oleh Ni Ketut Arini," *J. Igel J. Danc.*, vol. 3, no. 2, pp. 131–137, 2023.
- [6] N. M. D. Widiastuti, "Inovasi aplikasi media pembelajaran tari Bali berbasis android," *Mudra J. Seni Budaya*, vol. 33, no. 2, pp. 287–295, 2018.
- [7] I. Perbawa, M. Sudarma, and I. M. A. Suyadnya, "Aplikasi Pembelajaran Gerakan Dasar Tarian Bali dengan Simulasi Model 3D Berbasis Android," *J. SPEKTRUM*, vol. 1, no. 1, pp. 40–45, 2014.
- [8] A. A. G. B. Ariana, I. K. A. G. Wiguna, I. P. Y. Indrawan, A. A. Iswara, I. P. H. Permana, and N. L. K. R. I. D., "Balinese Dance Attributes Utilizing Augmented Reality with Marker Based Tracking Method," in *2024 IEEE International Symposium on Consumer Technology (ISCT)*, 2024, pp. 30–34.
- [9] J. Fesl, J. Jelínek, K. Horníčková, Z. Nevařilová, M. Konopa, and M. Feslová, "AI-based

- System for Cultural Heritage Objects Identification from Real Photos,” in *2022 12th International Conference on Advanced Computer Information Technologies (ACIT)*, 2022, pp. 476–479.
- [10] J. Mitric, I. Radulovic, T. Popovic, Z. Scekic, and S. Tinaj, “AI and Computer Vision in Cultural Heritage Preservation,” in *2024 28th International Conference on Information Technology (IT)*, 2024, pp. 1–4.
- [11] S. Tiribelli, S. Pansoni, E. Frontoni, and B. Giovanola, “Ethics of Artificial Intelligence for Cultural Heritage: Opportunities and Challenges,” *IEEE Trans. Technol. Soc.*, vol. 5, no. 3, pp. 293–305, 2024.
- [12] A. K. Sharma and H. Foroosh, “Slim-CNN: A Light-Weight CNN for Face Attribute Prediction,” in *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, 2020, pp. 329–335.
- [13] A. Ammar, A. Chebbah, H. B. Fredj, and C. Souani, “Comparative Study of latest CNN based Optical Flow Estimation,” in *2022 International Conference on Intelligent Systems and Computer Vision (ISCV)*, 2022, pp. 1–6.
- [14] I. Jaya, M. W. A. Kesiman, and I. M. G. Sunarya, “Detecting the same pattern in choreography balinese dance using convolutional neural network and analysis suffix tree,” *J. Ilm. Tek. Elektro Komput. Inform.(JITEKI)*, vol. 8, no. 3, pp. 410–421, 2022.
- [15] H. Ramdan, M. A. Soeleman, P. Purwanto, B. Imran, and R. A. Pramunendar, “Semantic segmentation of pendet dance images using multires U-Net architecture,” *Ilk. J. Ilm.*, vol. 14, no. 3, pp. 329–338, 2022.
- [16] A. G. Howard *et al.*, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv Prepr. arXiv1704.04861*, 2017.
- [17] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [18] H. Kim, H. Lee, J.-S. Kim, and S.-H. Ahn, “Image-based failure detection for material extrusion process using a convolutional neural network,” *Int. J. Adv. Manuf. Technol.*, vol. 111, no. 5, pp. 1291–1302, 2020.