Optimizing South Kalimantan Food Image Classification Through CNN Fine-Tuning

Muhammad Ridha Maulidi, Fatma Indriani, Andi Farmadi, Irwan Budiman, Dwi Kartini
Department of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Indonesia

ARTICLE INFO

Article history:

Received November 15, 2024
Revised December 10, 2024
Published January 16, 2025

Keywords:

CNN;
Fine-Tuning;
Image Classification;
Traditional Food;
South Kalimantan

ABSTRACT

South Kalimantan's rich culinary heritage encompasses numerous traditional dishes that remain unfamiliar to visitors and digital platforms. While Convolutional Neural Networks (CNNs) have demonstrated remarkable success in image classification tasks, their application to regional cuisine faces unique challenges, particularly when dealing with limited datasets and visually similar dishes. This study addresses these challenges by evaluating and optimizing two pre-trained CNN architectures—EfficientNetB0 and InceptionV3—for South Kalimantan food classification. Using a custom dataset of 1,000 images spanning 10 traditional dishes, we investigated various fine-tuning strategies to maximize classification accuracy. Our results show that EfficientNetB0, with 30 fine-tuned layers, achieves the highest accuracy at 94.50%, while InceptionV3 reaches 92.00% accuracy with 40 layers fine-tuned. These findings suggest that EfficientNetB0 is particularly effective for classifying regional foods with limited data, outperforming InceptionV3 in this context. This study provides a framework for efficiently applying CNN models to small, specialized datasets, contributing to both the digital preservation of South Kalimantan's culinary heritage and advancements in regional food classification. This research also opens the way for further research that can be applied to other less documented regional cuisines. The framework presented can be used as a reference for developing automated classification systems in a broader cultural context, thus enriching the digital documentation of traditional cuisines and preserving the culinary diversity of the archipelago for future generations.

Corresponding Author:

Fatma Indriani, Departement of Computer Science, Lambung Mangkurat University, Jalan A. Yani Km 36, Banjarbaru 70714, Indonesia
Email: f.indriani@ulm.ac.id

1. INTRODUCTION

Traditional food is a product of a specific region, made using traditional methods that rely on simple equipment. It typically uses local ingredients at relatively low cost, does not require special skills, and recipes are passed down through generations [1]. South Kalimantan is one of Indonesia's regions rich in culinary heritage, where citizens consume traditional food daily. Moreover, tourists visiting South Kalimantan often seek out local specialties such as Masak Habang and Soto Banjar.

Despite the popularity of South Kalimantan's traditional foods, digital resources, and structured documentation on these dishes are limited. This lack of accessible information can make it difficult for tourists or newcomers to identify and appreciate the unique flavors and cultural significance of these foods. The lack of a representative dataset is a significant challenge in building a reliable model to classify these food types accurately. Image classification models, particularly those using advanced deep learning methods, could offer a solution by providing accurate and immediate food identification. Although various deep learning architectures have shown promising performance in image classification, to date, no study has specifically evaluated the effectiveness of transfer learning architectures for identifying traditional foods of South Kalimantan due to the lack of sufficient datasets. To address these challenges., this research will develop a deep learning-based image classification model using transfer learning architecture by building a representative dataset and exploring the effectiveness of models such as EfficientNetB0 and InceptionV3. This method is expected to overcome data limitations while providing reliable results in detecting typical South Kalimantan food. This approach can bridge the gap between South Kalimantan's rich culinary heritage and the growing interest in regional cuisine while contributing to cultural preservation through modern technology.

A promising approach for traditional food identification is through Convolutional Neural Networks (CNNs), widely used for image processing tasks in fields like computer vision, speech recognition, and natural language processing [2], [3]. To improve CNN performance, various pre-trained architectures—such as EfficientNet and Inception—have been developed, offering strong initial models for specialized tasks through transfer learning [4]. The EfficientNet models represent a series of pre-trained convolutional neural networks (CNNs) specifically engineered for transfer learning applications within the domain of image classification. These models were conceived by researchers from Google AI, namely Mingxing Tan and Quoc V. Le, and were first introduced to the academic community in May 2019. The models are publicly accessible for scholarly and practical use through designated GitHub repositories, facilitating further research and development in the field of deep learning [5], [6]. GoogleNet, also known as Inception, is a deep convolutional neural network (CNN) architecture developed by Christian Szegedy and his collaborators at Google in 2014. This innovative architecture demonstrated remarkable performance by achieving a top-5 accuracy of 93.3% in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [7], [8]. Transfer learning, particularly fine-tuning, enables pre-trained models to adapt to new datasets by retraining select layers [9]. This technique is particularly effective for tasks with limited data, making it promising for classifying specific regional foods like those from South Kalimantan.

Numerous researchers have investigated food image classification. An ensemble technique combining ResNetX-101 and DenseNet-161 architectures achieved an accuracy of 90.02% on the UEC FOOD-100 dataset, as reported in a study [10]. In another study, [11] evaluated three distinct datasets (Food101, Sushi-50, and THFOOD-50), achieving 91.76% accuracy. Similarly, research conducted by [12] utilized the VGG-16 architecture to classify the Food101 dataset, attaining 85.07% accuracy. Research by [13] on the Food101 dataset using Particle Swarm Oprimizon on the NutriFoodNet architecture resulted in an accuracy of 98.5%. Research done by [14] resulted in an accuracy of 85.40% using Laplace Pyramid and LMB-net to classify Chinese food. Research [15] produced an accuracy of 93.00% using SwinTransformer to classify typical Chinese dimsum.

Research conducted by [16] achieved 94% accuracy using the VGG16 architecture to classify ten classes from the Food101 dataset. Similarly, [17] applied the VGG16 architecture for traditional Bengali food classification. In another study, [18] investigated traditional Indonesian food classification using a large dataset of 24,427 images spanning 160 different Indonesian dishes. Their research evaluated various pre-trained CNN models, with the EfficientNetV2-L architecture achieving the highest accuracy of 85.44%.

However, a significant knowledge gap exists in the systematic application of CNN-based transfer learning to South Kalimantan's traditional food classification. This gap presents an opportunity to explore how pre-trained models like EfficientNetB0 and InceptionV3 can be optimized through fine-tuning strategies, particularly in scenarios with limited data availability. Understanding the impact of fine-tuning depth on model performance remains crucial for developing effective classification systems for regional cuisines.

Our study addresses these challenges through two key contributions: (1) the compilation of a novel dataset comprising 1,000 images representing 10 distinct South Kalimantan dishes. By compiling this dataset, the study helps digitally preserve and document South Kalimantan's unique culinary traditions, making this cultural heritage more accessible to a global audience and ensuring its recognition for future generations. The dataset also serves as a valuable resource for future studies, educational purposes, and cultural initiatives. (2), a comprehensive evaluation of EfficientNetB0 and InceptionV3 architectures under various fine-tuning conditions. This research evaluates and compares these two state-of-the-art deep learning architectures to determine the most effective strategy for classifying images of South Kalimantan's traditional foods. By addressing challenges such as limited datasets and exploring transfer learning techniques, this study enhances the accuracy and reliability of food recognition models.

2. METHODS

The research started with collecting a dataset comprising 1,000 images of traditional South Kalimantan cuisine, covering ten different food types. After gathering the dataset, the next step was pre-processing, which involved preparing the data to ensure it was suitable for model training. Subsequently, a Convolutional Neural Network (CNN) model was developed using transfer learning architectures, specifically EfficientNetB0 and

InceptionV3. The model was then optimized by fine-tuning various layers and evaluated to measure its performance in classifying traditional food images. The research stages are presented in Fig. 1.

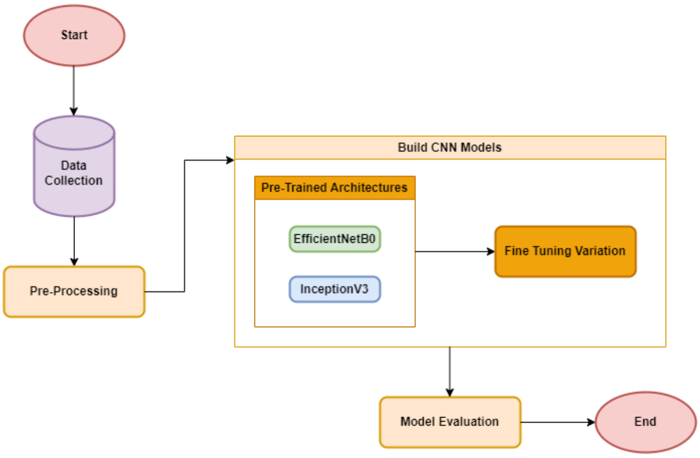


Fig. 1. Research Flow Diagram

2.1. Data Collection

The dataset used in this research was collected from various sites on the internet. The data is obtained by using the name of each traditional food as a keyword on the Google search engine. For example, for the Soto Banjar class, the keyword used was "Soto Banjar", and the same method was applied to data collection for the other classes. The images were manually selected to include a variety of perspectives, lighting conditions, and presentation styles, capturing different ways the dishes were prepared and served. Each class of dataset contains an equal allocation of 100 images, ensuring a balanced representation of the dataset. To ensure the dataset accurately represents South Kalimantan's traditional foods, a manual validation process was performed by involving local experts and individuals familiar with South Kalimantan's traditional foods. They reviewed each image to confirm that it accurately represented the intended dish, identifying and excluding any misclassified or unrelated images. This step was crucial to maintaining the reliability and cultural accuracy of the dataset. Table 1 displays a sample and description from each class of the dataset.

2.2. Pre-Processing

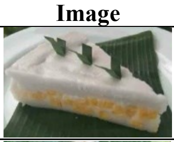
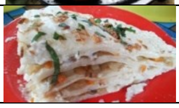
The collected images come from various file formats, including .png, .aviv, .jfif, and others. However, many of these images include visual noise, such as irrelevant objects like complex backgrounds, shadows, and non-food elements. Such issues can hinder the model's ability to recognize key features of the data accurately. To address these issues, pre-processing techniques were employed to refine the images and minimize noise. Pre-processing strategies are essential for optimizing raw data quality, thereby enhancing the input fed into the network. This enhancement contributes to improved robustness and accuracy of the model [19].

In this study, the pre-processing stage will be performed in several steps. The first step is to convert all collected images with various extension formats into the Joint Photographic Experts Group format (.jpg/.jpeg). Then, the image is cleaned of unnecessary objects by manually cropping the image from those objects. This process aims to ensure that each image is more focused on the class characteristics to be displayed, thereby improving the data quality and the model's performance in classification tasks. Fig. 2 shows the images before and after removing unnecessary objects.

The next step is to divide the data into three distinct subsets: training, validation, and testing. In this process, 70% of the data is allocated to the training set, 10% to the validation set, and 20% to the testing set. These proportions are widely recognized in deep learning as standard practices for effective model evaluation and validation [20]. The program will automatically perform the data splitting. Each set of images will then be normalized using a method appropriate for the pre-trained model that will be employed. Image normalization is the process of adjusting the intensity levels in order to standardize them across different cases [21].

Then, the augmentation process will be applied to the training data. The goal of data augmentation is to create more samples of the class while keeping the underlying category constant [22]. The augmentation techniques used include random rotation of up to 10 degrees, horizontal flipping to create variations in image orientation, and random contrast adjustment with a change rate of up to 10%. Image augmentation is performed only on the training data to prevent data leakage that could affect the model's validity. Data leakage refers to the inadvertent and erroneous incorporation of information from the validation dataset during the model development process. This phenomenon undermines the integrity of the predictive model by allowing insights from the validation set to unduly influence training, ultimately compromising the model's ability to generalize to unseen data [23]. This issue can compromise the integrity of the model assessment and lead to misleading performance metrics. Fig. 3 illustrates the preprocessing stages that have been previously explained.

Table 1. Ten Classes of the Dataset with Descriptions and Sample Images

Image	Description
	**Amparan Tatak** is a traditional banana-based food from South Kalimantan. It has a soft texture and a distinctive sweet flavor. It is usually made from a mixture of crushed bananas, rice flour, coconut milk, and sugar, then steamed until cooked.
	**Apam Barabai** is a traditional cake from South Kalimantan, specifically from the Barabai area, Hulu Sungai Tengah Regency. It has a soft and hollow texture and a mild, sweet flavor. The main ingredients of Apam Barabai are rice flour, brown sugar, and coconut milk, which are fermented to produce a distinctive aroma and unique texture.
	**Bingka** is a traditional cake from South Kalimantan, known for its soft, dense texture and sweet flavor. It is made from simple ingredients, including wheat flour, coconut milk, sugar, and eggs, which combine to create a rich and savory taste. The cake is typically shaped like a flower and has an enticing aroma of coconut milk.
	**Ipau** is a specialty from South Kalimantan that resembles lasagna in Western cuisine. It is made up of layers of thin skin created from wheat flour and eggs, filled with minced meat, vegetables, and a variety of specialty spices. Once assembled, Ipau is baked or steamed until fully cooked.
	**Ketupat Kandangan** is a traditional dish from the Kandangan area of Hulu Sungai Selatan Regency. This dish features ketupat (which is rice cooked in woven coconut leaves) served with a thick and spiced coconut milk sauce. Typically, Ketupat Kandangan is served with haruan (cork) fish cooked with spices and coconut milk, resulting in a rich and savory fl avor. A unique aspect of this dish is its consumption; locals often eat the ketupat directly with their hands, reflecting their cultural eating traditions.
	**Lontong Orari** is a dish from South Kalimantan that offers a unique variation of lontong, one of Indonesia's traditional foods. The dish features lontong (made from rice cooked in banana leaves) served with a thick and richly spiced coconut milk gravy. It is typically accompanied by side dishes such as boiled eggs, pieces of meat or chicken, and sambal, which adds a spicy flavor.
	**Manday** is a traditional dish made from the fermented flesh of young cempedak (also known as jackfruit). This fermentation process imparts a unique flavor to Manday, which is sour, savory, and slightly spicy when adding chilies. After fermentation, Manday is typically prepared by frying, grilling, or cooking it in coconut milk, resulting in a rich and flavorful dish. It is a favored side dish among the Banjar people, enjoyed as a daily meal and complementing traditional events.
	**Masak Habang** is a dish known for its deep red gravy, which comes from dried red chilies as the main spice. The name 'habang' means 'red' in the Banjar language. The dish usually uses main ingredients such as chicken, fish, eggs, or beef, which are cooked with a mixture of traditional spices such as shallots, garlic, coconut milk, brown sugar, and other spices. Masak Habang's taste tends to be sweet and savory, typical of Banjar cuisine, with an adjustable level of spiciness.
	**Sop Mutiara** is a traditional soup dish from South Kalimantan, renowned for its fresh, clear broth and light yet rich flavor. Typically made with a main ingredient such as chicken or beef, it is cooked with aromatic spices, including garlic, shallots, ginger, and herbs. What distinguishes Sop Mutiara is the addition of "pearls," which are chewy balls made from sticky rice or sago, providing the dish with a unique texture. Sop Mutiara is often served with white rice and is commonly enjoyed at events or as a part of everyday meals.
	**Soto Banjar** is a renowned dish from South Kalimantan, celebrated for its rich soup and distinctive aroma of spices. It features chicken cooked with cinnamon, cloves, cardamom, shallots, garlic, and ginger, resulting in a savory and fragrant flavor. The uniqueness of Soto Banjar also comes from its accompanying side dishes, which include pieces of rice cake, boiled eggs, lime wedges, and vermicelli. The dish is typically served with a special sambal and topped with fried onions and celery leaves, enhancing its rich flavors and varied textures.

(a) (b)

Fig. 2. (a) before removing unnecessary objects (b) after removing unnecessary objects

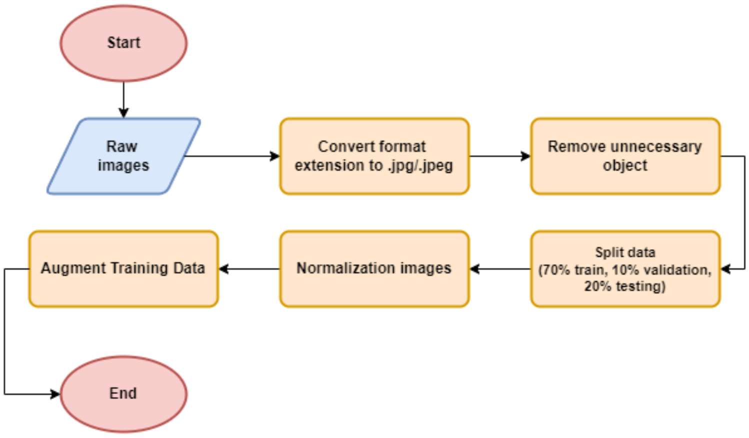


Fig. 3. Pre-Processing Flow Diagram

2.3. Building Model

1) Convolutional Neural Network (CNN)

One of the deep learning algorithms that is often used to complete machine learning tasks is Convolutional Neural Networks (CNN). CNN is modeled after the architecture of the human brain [24]. CNNs represent a specialized architecture of neural networks that are particularly adept at processing data structured in grid-like formats. This includes various forms of data, such as time series, which can be considered as one-dimensional grids, and images, which are typically organized as two-dimensional arrays of pixels [25]. CNN-based models were designed to identify a certain output class after learning the internal representation of 2d pictures. The similar methodology can be applied to time series sequencing data classification and automatic feature learning [26]. The main advantage of CNN over earlier algorithms lies in its capacity to autonomously identify and extract pertinent features from input data without the necessity of human intervention. This inherent ability significantly enhances their applicability and has led to their widespread adoption across various domains [27]. CNN consists of several layers as can be seen in Fig. 4.

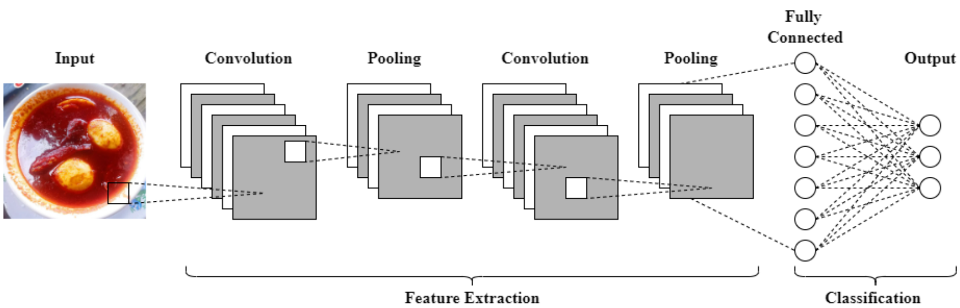


Fig. 4. CNN Architectures

- Convolutional Layer: The convolutional layer serves as a pivotal element within the architecture of Convolutional Neural Networks (CNNs). This layer is characterized by the presence of a collection of convolutional filters, commonly referred to as kernels. These filters are systematically applied to the input image, which is typically represented as N-dimensional matrices, resulting in the generation of output feature maps. Such feature maps are instrumental in emphasizing particular characteristics intrinsic to the data. As the network deepens, the representations derived from these layers transform, evolving into increasingly complex and abstract forms. This progression significantly enhances the network's capacity to learn and represent nonlinear features within the data effectively [28].

- Pooling Layer: The main task of the pooling layer is the sub-sampling of the feature maps. Pooling is a main step in CNNs that decreases the feature maps' dimensionality. Pooling is an essential process in CNN that aims to reduce the dimensionality of feature maps. This technique involves condensing a set of values into a smaller, more manageable set. By retaining only the most crucial information and discarding the less significant data, pooling effectively helps transform the joint feature representation into valuable insights [29].
- The activation function: The activation function is a nonlinear mechanism that assigns an Artificial Neural Network (ANN) to transform input data into a higher-dimensional space, thereby simplifying the creation of hyperplanes for classification. This function is essential as it facilitates the mapping of input to output across the diverse activation functions employed in various neural networks. In Convolutional Neural Networks (CNNs), the activation function introduces nonlinearity, enabling the model to effectively capture intricate relationships between image features and their corresponding class labels [30].
- Fully Connected Layer: The fully connected layer derives its name from its comprehensive connections to the preceding layer. Whether the prior layer is a convolutional layer or a pooling layer, every neuron in the fully connected layer is interconnected with all neurons in the previous layer. Additionally, this layer serves as the final component of a CNN architecture, meaning its output represents the definitive result of the network [31].

2) Pre-Trained Architectures

A pre-trained model refers to a neural network architecture, often a convolutional neural network (CNN), that has already undergone training on a large dataset, such as ImageNet, CIFAR-10, or MNIST. The main objective of utilizing a pre-trained model is to resolve specific classification tasks, including image classification, object detection, and image captioning. By having already learned significant features from the training data, these models conserve both time and computational resources, as they eliminate the need for the initial learning phase, which can be costly and time-consuming [32]. This approach is particularly advantageous, as training a model from scratch can be prohibitively time-consuming and resource-intensive. Pre-trained models have demonstrated remarkable effectiveness across diverse domains, from language tasks to image task [33]. Various pre-trained CNN model architectures can be used to solve machine-learning tasks. In this research, the pre-trained model architectures chosen are EfficientNetB0 and InceptionV3.

- EfficientNetB0 serves as the foundational model within the EfficientNet family, which encompasses more advanced variations (B1 – B7). Introduced by Google in 2019, EfficientNet is engineered to deliver high performance while requiring fewer parameters and floating point operations (FLOPs) compared to other CNN architectures [34]. The EfficientNet series begins with its foundational model, EfficientNet B0, which is crafted to strike an optimal balance between model size and performance. The architecture comprises several layers, starting with an input layer that processes images sized at 224×224 pixels, featuring RGB color channels— a standard dimension for many computer vision tasks. Following this are conventional convolutional layers, where the first layer employs 3x3 filters, complemented by batch normalization layers and Swish activation functions (also referred to as SiLU) [35].
- InceptionV3 is a variation of the Inception architecture developed by Szegedy *et al.* at Google. This architecture was originally introduced during the ImageNet Recognition Challenge and presented in the paper titled "Rethinking the Inception Architecture for Computer Vision" in 2015. The input image dimensions for InceptionV3 are specified as 299 × 299 pixels. This architecture employs convolutional kernels of varying sizes, which allows for the establishment of receptive fields at different scales. To optimize the network's design, InceptionV3 uses a modular framework that culminates in a final integration process, enabling effective fusion of features extracted from various scales. Additionally, the architecture incorporates a batch normalization (BN) layer as a regularization technique between the auxiliary classifier and the fully connected (FC) layer, enhancing the model's overall stability and performance [36].

The comparison between EfficientNetB0 and InceptionV3 architectures, is summarized in Table 2.

Table 2. Comparison between between EfficientNetB0 and InceptionV3 architectures

Aspect	EfficientNetB0	InceptionV3
Released Year	2019	2016
Model Size	5.3 million parameters	23.9 million parameters
Standard Input Resolution	224×224 pixels	299×299 pixels
Inference Speed	Relatively faster	Relatively slower
Performance	High performance with greater efficiency on both large and small datasets.	Excellent performance on large datasets such as ImageNet.
Main Architecture	Employs compound scaling to optimize resolution, depth, and model width simultaneously.	Utilizes factorized convolutions for computational efficiency.

3) Fine-Tuning

Fine-tuning is a widely used method in transfer learning. It begins with training a base network, after which the initial n layers are transferred to the corresponding layers of a target network. The remaining layers of the new target network are generally initialized randomly and then trained for a specific task, thereby facilitating the adaptation of the model to novel domains [37]. The fine-tuning concept used in this research is to freeze some pre-trained model layers and retrain the final layers on a new dataset. The rationale underpinning this strategy is influenced by the constraints associated with size-limited datasets and empirical findings indicating that the initial layers (i.e., the lower layers) of a deep neural network tend to encode more abstract and generalized features. These features are often applicable across various tasks, whereas the upper layers are more specialized, containing task-specific representations [37].

The model proposed in the research uses a Pre-Trained CNN architecture with the application of fine-tuning variations. After data pre-processing, the model will be trained separately using two pre-trained CNN architectures, namely EfficientNetB0 and InceptionV3. Training is performed for 100 epochs by applying a reduced learning rate mechanism, where the learning rate will be reduced (current learning rate multiplied by 0.2 with a maximum limit of 0.000001) if validation loss does not show improvement after three consecutive epochs. The early stopping callback function will be applied, which will automatically stop the training after the next two epochs when the learning rate has been reduced, but the validation loss still has not improved. The training process will employ the Adam optimizer, commencing with an initial learning rate set at 0.001. The loss function applied is categorical cross-entropy because the dataset used is multi-class. The model effectively uses a combination of a reduced learning rate and early stopping to mitigate overfitting. By dynamically adjusting the learning rate and stopping the training process when performance improvements plateau, these techniques enhance the model's robustness and ability to generalize.

Training begins by loading the dataset into the input layer, which will resize the dataset to 224 x 224 pixels. Next, the data will be trained using a pre-trained model that has been selected by applying fine-tuning variations to the last few layers of the model. The next training process uses a custom CNN model with a neuron configuration in dense layers of 128 and 256 with the ReLU activation function. Moreover, the model integrates a dropout layer characterized by a neuron randomization rate of 0.45, alongside a second dropout layer that generates ten outputs, in accordance with the number of classes present in the dataset. Dropout is recognized for its ability to reduce overfitting by enhancing the overall robustness of the network and decreasing its reliance on specific neurons. Fig. 5 shows the model architecture used in the training process.

2.4. Evaluation Metrics

The trained model will be evaluated to assess the performance of the model with separate dataset to ensure its generalization capability and robustness [38]. In this research, the evaluation will use several metrics, such as accuracy, precision, recall, f-1 score, and confusion matrix.

Accuracy is quantitatively defined as the ratio of correctly identified crack and non-crack patches to the total number of input patches analyzed. This metric serves as an essential indicator of the proportion of accurate predictions in relation to the entirety of the dataset under consideration [39], [40]. Accuracy is suitable for balanced datasets. Accuracy can be calculated using the formula below.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

True positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) are fundamental metrics utilized in the assessment of classification model performance. TP and TN denote the number of instances accurately classified as positive and negative, respectively. Conversely, FP and FN reflect the number of instances incorrectly identified as positive and negative [41]. These metrics are essential for understanding the efficacy of predictive models in various research and application fields.

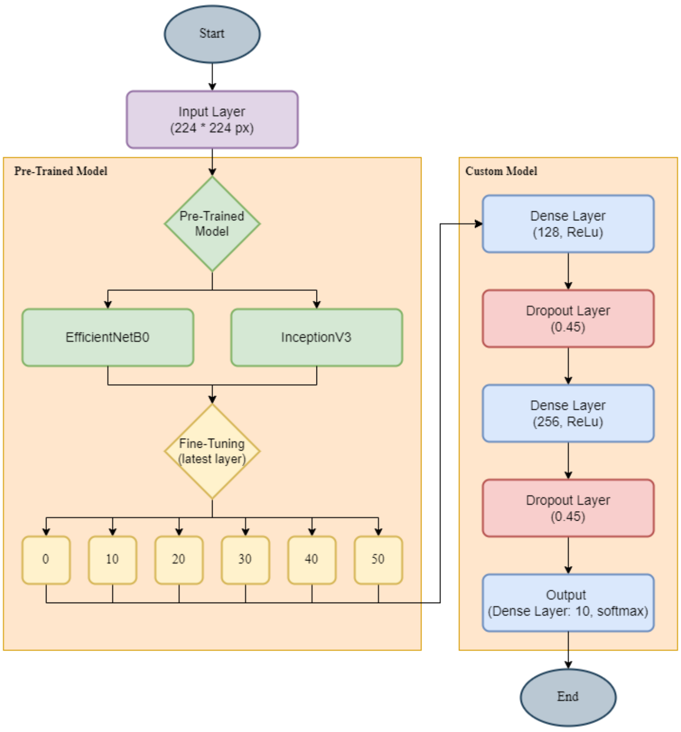


Fig. 5. Model Training Architectures

Precision is defined as the ratio of true positive predictions (TP) to the total number of predicted positive instances, encompassing both true positives and false positives (TP + FP). Optimal precision is represented by a value of 1.0, indicating perfect predictive accuracy, while a value of 0.0 signifying no correct positive predictions [42], [43]. This metric reflects the model's ability to minimize false positives, which is crucial when incorrect classifications can lead to significant consequences. In the context of this study, precision ensures that predictions for a specific traditional food class are accurate and not falsely attributed to unrelated images.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Recall, often known as sensitivity, quantifies a model's capability to identify all relevant instances within a dataset. It is calculated as the ratio of true positives to the total of true positives and false negatives. This metric reflects how many actual positive instances the model successfully identified. Recall values range from 0 to 1, where a value of 1 indicates flawless prediction of the positive class, while a value of 0 signifies that none of the positive samples were accurately predicted [44], [45]. In this study, recall is particularly important to ensure that all instances of a specific traditional food class are detected, minimizing the likelihood of false negatives where actual instances are overlooked.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

The F1 score is a critical metric in evaluating classification models, representing the harmonic mean of precision and recall. This score is constrained within the interval [0, 1]. A score of 0 indicates a scenario with no true positive instances (TP = 0), thereby suggesting that all positive samples have been misclassified. In contrast, a maximal score of 1 is attained when false negatives (FN) and false positives (FP) are equal to zero, which signifies flawless classification performance [46], [47]. F1-score is essential for assessing the balance between precision and recall across all traditional food classes. It ensures that the evaluation is not biased toward either false positives or false negatives, particularly in cases where some classes might be harder to classify due to subtle feature differences.

$$F1\ Score = 2 \cdot \frac{precision \cdot recall}{precision + recall} = 2 \cdot \frac{TP}{TP + FP + FN} \tag{4}$$

A confusion matrix serves as a fundamental instrument for evaluating the performance of classification algorithms, providing a comprehensive framework for quantifying predictive accuracy and identifying specific areas of misclassification [48]. A confusion matrix is a table that summarizes the instances classified by two raters: the actual classifications and the predicted classifications. The classes are consistently arranged along the rows and columns. As a result, the correctly classified items appear on the main diagonal, extending from the top left to the bottom right of the matrix, which indicates the number of times the two raters have agreed [49]. Fig. 6 presents an illustration of the confusion matrix.

		Predicted Class		
	Classes	**A**	**B**	**C**
Actual Class	**A**	19	0	1
	B	0	12	8
	C	1	2	17

Fig. 6. Example of Confusion Matrix

3. RESULTS AND DISCUSSION

3.1. Overall Result

The models generated from both architectures were evaluated using new test data that was not used during training. This test data aims to assess the model's ability to generalize to previously unrecognized data and ensure that the model has reliable and accurate performance in classifying traditional food images of South Kalimantan in more real conditions. The evaluation metrics used in this study include accuracy, precision, recall, and the F1-score. Confusion matrix will be included to provide a more detailed picture of the model's performance in classifying each class precisely so that the strengths and weaknesses of the model in each category can be analyzed in depth. Table 3 are the evaluation results of all models in this study.

Table 3. Comparison of Evaluation Result for Each Model

Model/Architecture	Fine-Tuning (latest layer)	Accuracy	Precision	Recall	F1-Score
EfficientNEtB0	0	85.00%	0.8671	0.8500	0.8505
	10	89.50%	0.9000	0.8950	0.8942
	20	90.50%	0.9198	0.9050	0.9046
	30	**94.50%**	**0.9493**	**0.9450**	**0.9448**
	40	91.00%	0.9175	0.9100	0.9100
	50	91.50%	0.9197	0.9150	0.9149
InceptionV3	0	30.00%	0.2230	0.3000	0.2285
	10	64.50%	0.6691	0.6450	0.6417
	20	88.50%	0.8900	0.8850	0.8805
	30	91.50%	0.9050	0.9150	0.9007
	40	**92.00%**	**0.9250**	**0.9200**	**0.9203**
	50	90.00%	0.9102	0.9000	0.9086

In Table 3, the evaluation results on both models without *fine-tuning* (0 trainable layers) show a significant performance difference between EfficientNetB0 and InceptionV3 architectures. EfficientNetB0 without *fine-tuning can* achieve 85.00% accuracy with precision, *recall*, and *F1-score* metrics of 0.8671, 0.8500, and 0.8505, respectively. These results show that the model is already good enough to classify traditional South Kalimantan food with consistent results across all evaluation metrics. In contrast, InceptionV3 with 0 *fine-tuning layers* showed deficient performance, with an accuracy of only 30.00% and precision, *recall*, and *F1-score of* 0.2230, 0.3000, and 0.2285, respectively. These results indicate that the InceptionV3 model without *fine-tuning* cannot recognize enough patterns from the training data to make accurate predictions on the test data. From the results of the two models, it can be said that EfficientNetB0 has a more stable and ready-to-use initial performance than InceptionV3 when there are no trainable layers.

The evaluation results of the EfficientNetB0 model in Table 3 show a consistent improvement in classification performance with the number of fine-tuning layers added. In the no fine-tuning condition (0th layer), the model achieved 85.00% accuracy, with relatively good precision, recall, and F1-score, but not optimal performance. By fine-tuning the last 10 to 20 layers, there was a significant improvement in accuracy and other metrics, with accuracy increasing to 90.50% by the 20th layer. The best performance improvement occurred with fine-tuning the last 30 layers, with the highest accuracy of 94.50% and improved precision, recall, and F1-score values. These results show that the model can better understand the data's unique features with deeper fine-tuning, resulting in more accurate classification. However, the performance slightly decreased when fine-tuning was applied to the last 40 and 50 layers, with 91.00% and 91.50% accuracy. This decrease suggests that too deep fine-tuning may result in overfitting the training data, reducing the model's generalization ability.

The InceptionV3 model shows significant performance variations depending on the number of fine-tuning layers applied. When fine-tuning is performed at 0 layers (no fine-tuning), the model only achieves 30% accuracy, with very low precision, recall, and F1-score metrics, indicating that the model cannot perform classification well. However, as fine-tuning layers increase, the model's overall performance improves. After ten fine-tuning layers, the accuracy increased to 64.5%, with better precision and recall. At the 20-layer fine-tuning variation, the model recorded an accuracy of 88.5%, with precision, recall, and F1-score values close to 0.89, showing significant progress in recognizing the data. The best performance was recorded at fine-tuning 40 layers, with accuracy reaching 92% and excellent evaluation metrics of precision 0.925, recall 0.92, and F1-score 0.9203, indicating the model's ability to perform classification well. However, after 50 layers of fine-tuning, the model's performance slightly decreased (the accuracy dropped to 90%), but the other metrics remained relatively high.

To give a clearer picture of each model's classification ability, the following shows the confusion matrix of the best model of each architecture, namely EfficientNetB0 with fine-tuning 30 layers and InceptionV3 with fine-tuning 40 layers. The Confusion matrix of the EfficientNetB0 model with 30-layer fine-tuning can be seen in Fig. 7.

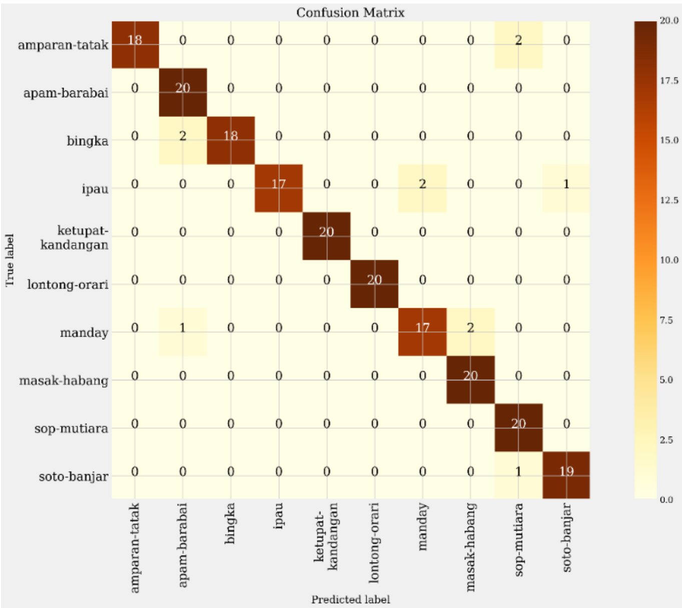


Fig. 7. Confusion Matrix EfficientNetB0 with 30 fine-tuning layers

Fig. 7 shows that the Confusion matrix of the EfficientNetB0 model with 30-layer fine-tuning performs very well in the classification of this dataset. The high diagonal values indicate that most images in each class are correctly classified. The model can recognize most classes consistently, such as **Apam Barabai**, **Ketupat Kandangan**, **Lontong Orari**, **Masak Habang**, and **Sop Mutiara**. However, there is a slight misclassification in some classes. **Amparan Tatak** had two images that were misclassified as **Sop Mutiara**. Similarly, **Bingka** had two images classified as **Apam Barabai. Meanwhile, Ipau** showed misclassification with some images categorized as **Manday** and **Soto Banjar**.

Meanwhile, the Confusion matrix of the InceptionV3 model with fine-tuning at 40 layers in Fig. 8 shows fairly good classification results with most of the class samples in **Apam Barabai**, **Bingka**, **Lontong Orari**, **Manday**, and **Masak Habang** being almost entirely correctly classified. However, some classes showed misclassification. **Amparan Tatak** was misclassified as **Ketupat Kandangan** for one image. **Apam Barabai** and **Bingka** also experienced some confusion, each with one misclassified image. In addition, **Ketupat Kandangan** was also misclassified twice as **Soto Banjar**.

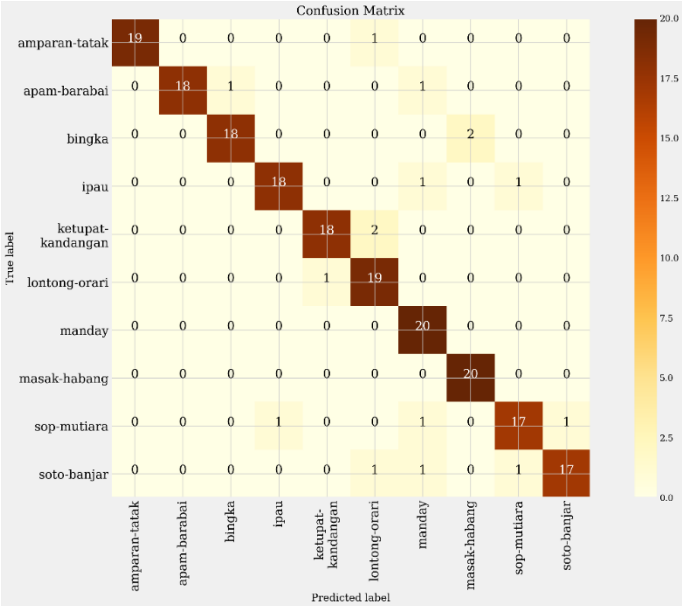


Fig. 8. Confusion Matrix InceptionV3 with 40 fine-tuning layers.

Overall, the performance of both models shows satisfactory results in the classification of South Kalimantan food images. Both models successfully classified most classes with high accuracy, as seen from the high values on the diagonal of their respective confusion matrices. However, there were some misclassifications in both models, which generally occurred between classes that may have visual similarities, such as **Amparan Tatak** sometimes being misrecognized as **Sop Mutiara** and **Ketupat Kandangan** being misrecognized as **Soto Banjar**. This suggests that the visual features of certain foods may need more emphasis or additional data to reduce ambiguity in classification. A comparison between the two models shows that **EfficientNetB0** is slightly more stable in classifying certain classes.

3.2. Classes Accuracy Performance

The following Table 4 compares of the accuracy of each class from each of the best models—EfficientNetB0 with 30 finetuning layers and InceptionV3 with 40 finetuning layers.

Table 4. Comparison of Accuracy Result for Each Classess

Class(es)	Architecture(s)	
	EfficientNetB0	**InceptionV3**
Amparan Tatak	90.00%	95.00%
Apam Barabai	100%	90.00%
Bingka	90.00%	90.00%
Ipau	85.00%	90.00%
Ketupat Kandangan	100%	90.00%
Lontong Orari	100%	95.00%
Manday	85.00%	100%
Masak Habang	100%	100%
Sop Mutiara	100%	85.00%
Soto Banjar	95.00%	85.00%
Average	**94.50%**	**92.00%**

Based on the given table, the EfficientNetB0 architecture shows a very good performance with very high accuracy, reaching 100% in the **Apam Barabai**, **Ketupat Kandangan**, **Lontong Orari**, and **Sop Mutiara** classes. This result indicates that the features possessed by these foods can be recognized very well by EfficientNetB0, so the model is able to avoid misclassification in these classes. Meanwhile, for the soto-banjar class, EfficientNetB0 achieved 95% accuracy, which is slightly lower than the other classes, but still shows good performance in recognizing the class.

In Table 4, there are also some classes where InceptionV3 shows better performance than EfficientNetB0. For example, in the Amparan Tatak and Ipau classes, InceptionV3 achieved 95% and 90% accuracy, while EfficientNetB0 achieved 90% and 85%. In addition, for the **Manday** class, InceptionV3 successfully classified all images correctly (100%), while EfficientNetB0 achieved 85% accuracy. Table 2 also shows that there is a balance of performance between the EfficientNetB0 and InceptionV3 models in some classes, such as Bingka and Masak Habang, where both models show the same accuracy of 90% and 100%, respectively.

Overall, the EfficientNetB0 model showed superior performance compared to InceptionV3 with an average accuracy of 94.50% compared to 92.00%. So, it can be said that EfficientNetB0 is more reliable in handling visual variation and complexity in the South Kalimantan specialty food dataset, especially the dataset in this study, where it provides a more consistent accuracy level in almost all classes.

3.3. Misclassification Analysis

Misclassification analysis will focus on the two pre-trained models with the best performance—EfficientNetB0 with 30 fine-tuning layers and InceptionV3 with 40 fine-tuning layers. Fig. 9 show missclassification data on model EfficientnetB0 with 30 fine-tuning layers.

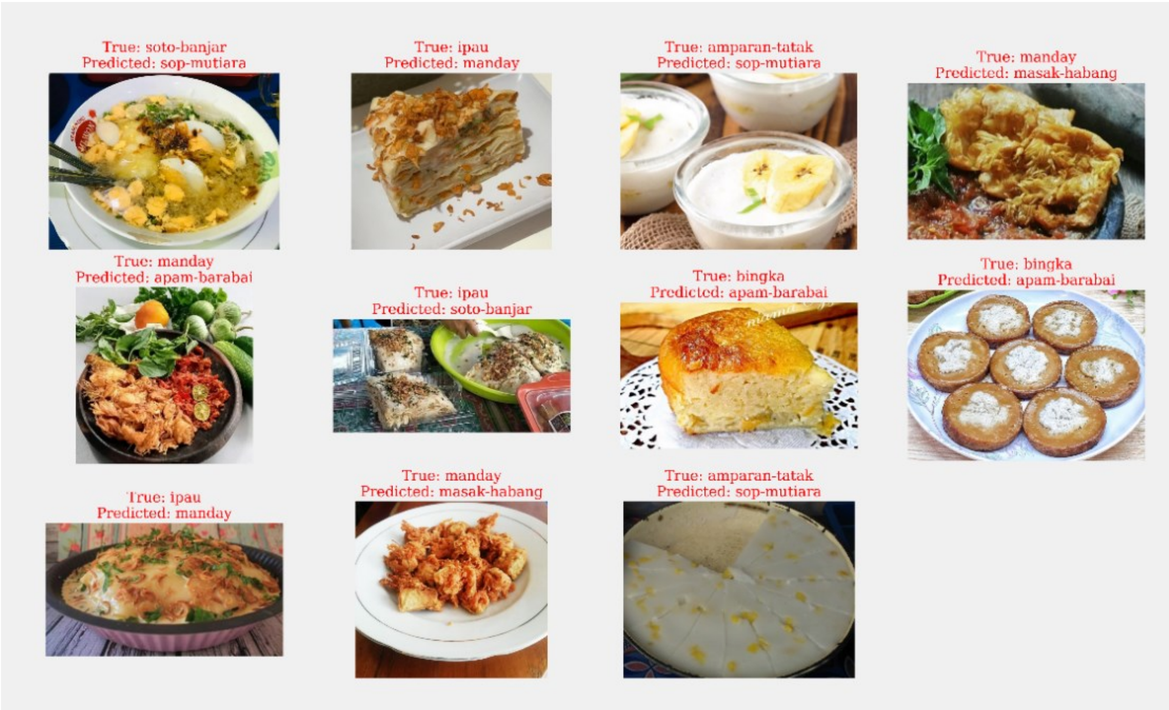


Fig. 9. Misclassified Samples of the EfficientNetB0 Model with 30 Fine-Tuning Layers

Model EfficientNetB0 with 30 fine-tuning layers highlights several challenges in distinguishing visually similar traditional food classes. Notable errors include **Amparan Tatak** being misclassified as **Sop Mutiara** and **Manday** being confused with **Masak Habang**. These errors suggest that the model may rely heavily on surface-level features such as shape, color, or general appearance, which can be similar across some classes.

For instance, **Amparan Tatak** and **Sop Mutiara** share a creamy texture and light-colored appearance, making them visually similar. The same goes for **Soto Banjar**, which was misclassified as **Sop Mutiara** because the similar visual style, such as light-colored soup with similar toppings, made it difficult for the model to distinguish the unique characteristics of each food. Similarly, **Masak Habang** and **Manday** have overlapping colour palettes and presentation styles, potentially leading to confusion. Foods like **Bingka** and **Apam Barabai** also show misclassification due to similarities in shape and texture.

The InceptionV3 model with 40 fine-tuning layers also showed challenges distinguishing visually similar traditional food classes. Fig. 10 presents the data related to the misclassified classes.

\

Fig. 10. Misclassified Samples of the InceptionV3 Model with 40 Fine-Tuning Layers

The model showed difficulty distinguishing between images of dishes with similar soft textures or toppings as in the Sop Mutiara class, which was misclassified with Ipau and Manday. Also, Ketupat Kandangan was misclassified as Lontong Orari in some cases due to overlapping visual features such as similar texture elements (color, shape, and others). Soto Banjar, often misclassified as Manday or Lontong Orari, highlights the difficulty in identifying unique presentation details that distinguish it from other dishes. Ipau and Sop Mutiara often intersect in misclassification, demonstrating the challenge of distinguishing dishes with a soft, multi-layered texture and delicate appearance. In addition, Amparan Tatak being mistaken for Lontong Orari further underscores the difficulty of modeling foods with the same light color and subtle differences in structure or presentation style.

Both models show a reliance on common visual patterns, making classification difficult when foods have similar aesthetic characteristics. Although both architectures achieve high performance, the errors indicate that these classification approaches can still be improved using additional techniques. Strategies like multi-modal learning or increasing the dataset with more diverse and discriminative samples could improve classification accuracy.

3.4. Comparison with Others Research

A comparison of accuracy results with previous studies is presented in Table 5. From Table 5, it can be seen that the model used in this study achieved the highest accuracy of 94.50%. This result shows the model's advantage to handle the visual complexity of food images, especially for the dataset used in this study. Compared to research by [53], who used the K-Nearest Neighbors (KNN) and Support Vector Machine (SVM) methods with an accuracy of 62.79%, which shows the limitations of traditional methods such as KNN and SVM in handling complex food image classification, especially when dealing with high visual and texture variations in food images. The advantage of the model in this study can also be seen from the comparison with other studies that also use deep learning approaches with various pre-trained architectures, such as InceptionV3, MobileNet, and VGG, which achieve the highest accuracy between 81.91% to 91.49%.

The models used in this study demonstrate the effectiveness of applying fine-tuning to pre-trained CNN architectures, especially EfficientNetB0 and InceptionV3. This method optimizes the model's ability to extract visual features from images of typical South Kalimantan food so that the model can adapt to the specific characteristics of the dataset used. High-accuracy results show that this model can handle food image classification effectively even with limited dataset availability.

Table 5. Comparison With Previous Research

No	Researcher(s)	Method(s)	Case	Accuracy
1	Chun, *et al.* [50]	**InceptionResnetV2**, NasNetLarge, MobileNetV2, ResNet101-V2,	Korean food image classification	81.91%
2	Wulandari [51]	ResNet152-V2, ResNet50-V2, VGG16, InceptionV3, MobileNetV3, **DenseNet-201**	Indonesian Speciality Cuisine Classification	90.45%
3	Ittisoponpisan, *et al.* [52]	MobileNetV1, MobileNetV2, MobileNetV3, **EfficientNetV1**, EfficientNetV2, ResNet50V1, ResNet50V2	Thai food image classification using THFOOD-50 dataset	91.49%
4	Rao, *et al.* [53]	KKN, SVM	Indian food image classification	62.79%
5	Tasci [54]	ResNet, GoogleNet, VGGNet, and InceptionV3 with fine-tuning	Image classification on Food-101, UEC-FOOD100, and UEC-FOOD256 datasets.	84.52%
6	**This Method**	**EfficientNetB0 & InceptionV3 with Fine-Tuning Implement**	**South Kalimantan traditional food images**	**94.50%**

4. CONCLUSION

This study demonstrates the successful application of transfer learning for South Kalimantan traditional food classification using EfficientNetB0 and InceptionV3 architectures. Through systematic evaluation of fine-tuning strategies, we established that EfficientNetB0 achieves superior performance with 94.50% accuracy, compared to InceptionV3's 92.00%. These findings validate the effectiveness of transfer learning for specialized image classification tasks with limited domain-specific data, suggesting that smaller, efficiently scaled architectures like EfficientNetB0 may be particularly well-suited for regional food classification tasks.

While our balanced dataset of 10 food classes provided valuable insights, several opportunities for future research emerge. These include expanding the dataset to cover more South Kalimantan dishes, investigating model performance under unbalanced dataset conditions, and exploring alternative fine-tuning strategies. Furthermore, future studies could examine the integration of data augmentation techniques specific to food imagery to enhance model robustness and generalization ability.

In conclusion, this research contributes to both the technical understanding of transfer learning in specialized domains and the practical application of deep learning for cultural heritage preservation, providing a foundation for developing robust food classification systems that can enhance culinary tourism experiences.

REFERENCES

[1] N. Suryani, A. N. Eliandha, and O. Libri, “Analisa Kandungan Zat Gizi Makro, Serat, Natrium, dan Kolesterol pada Kue Khas Kalimantan Selatan,” *Jurnal Kesehatan Indonesia (The Indonesian Journal of Health)*, vol. 7, no. 3, pp. 125–29, Jul. 2021, [Online]. Available: https://journal.stikeshb.ac.id/index.php/jurkessia/article/download/448/194

[2] J. Jiang, “The eye of artificial intelligence - Convolutional Neural Networks,” *Applied and Computational Engineering*, vol. 76, no. 1, pp. 273–279, Jul. 2024, https://doi.org/10.54254/2755-2721/76/20240613.

[3] L. Balderas, M. Lastra, and J. M. Benítez, “Optimizing Convolutional Neural Network Architecture,” *Mathematics*, vol. 12, no. 3032, pp. 1–19, 2024, https://doi.org/10.3390/math12193032.

[4] H. Hashmi, R. K. Dwivedi, and A. Kumar, “Comparative Analysis of CNN-Based Smart Pre-Trained Models for Object Detection on Dota,” *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 18, no. 2, pp. 31–45, Jun. 2024, https://doi.org/10.14313/JAMRIS/2-2024/11.

[5] G. Marques, D. Agarwal, and I. de la Torre Díez, “Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network,” *Applied Soft Computing Journal*, vol. 96, Nov. 2020, https://doi.org/10.1016/j.asoc.2020.106691.

[6] M. Harahap *et al.*, “Skin cancer classification using EfficientNet architecture,” *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 4, pp. 2716–2728, Aug. 2024, https://doi.org/10.11591/eei.v13i4.7159.

[7] B. Alotaibi and M. Alotaibi, “A Hybrid Deep ResNet and Inception Model for Hyperspectral Image Classification,” *PFG - Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, vol. 88, no. 6, pp. 463–476, Dec. 2020, https://doi.org/10.1007/s41064-020-00124-x.

[8] A. E. Minarno, L. Aripa, Y. Azhar, and Y. Munarko, “Classification of Malaria Cell Image Using Inception-V3 Architecture,” *International Journal On Informatics Visualization*, vol. 7, no. 2, pp. 273–278, 2023, https://dx.doi.org/10.30630/joiv.7.2.1301.

[9] F. Raza, “Transfer Learning in Deep Neural Networks,” *Cosmic Bulletin Of Business Management*, vol. 2, no. 1, pp. 189–205, Sep. 2023, http://dx.doi.org/10.13140/RG.2.2.25635.63529.

[10] B. Arslan, S. Memis, E. B. Sonmez, and O. Z. Batur, “Fine-Grained Food Classification Methods on the UEC FOOD-100 Database,” *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 2, pp. 238–243, Apr. 2022, https://doi.org/10.1109/TAI.2021.3108126.

[11] D.-R.-N. Wang and Y. Dong, "Research on Food Fine Grain Image Classification Based on Improved Residual Network," *Res Sq*, 2023, https://doi.org/10.21203/rs.3.rs-3377448/v1.
[12] S. Yadav, Alpana, and S. Chand, "Automated Food image Classification using Deep Learning approach," in *2021 7th International Conference on Advanced Computing and Communication Systems, ICACCS 2021*, Institute of Electrical and Electronics Engineers Inc., Mar. 2021, pp. 542–545. https://doi.org/10.1109/ICACCS51430.2021.9441889.
[13] E. S. Sreetha, G. N. Sundar, and D. Narmadha, "Enhancing Food Image Classification with Particle Swarm Optimization on NutriFoodNet and Data Augmentation Parameters," *International Journal of Computational and Experimental Science and Engineering*, vol. 10, no. 4, pp. 718–730, Oct. 2024, https://doi.org/10.22399/ijcesen.493.
[14] S. Feng, Y. Wang, J. Gong, X. Li, and S. Li, "A fine-grained recognition technique for identifying Chinese food images," *Heliyon*, vol. 9, no. 11, Nov. 2023, http://dx.doi.org/10.1016/j.heliyon.2023.e21565.
[15] K. Yu, "DimSum50: A benchmark dataset of Chinese food," *Applied and Computational Engineering*, vol. 73, no. 1, pp. 271–279, Jul. 2024, https://doi.org/10.54254/2755-2721/73/20240411.
[16] S. Mainkar, "Efficient Deep Learning Approach for Food Image Classification," *The International journal of analytical and experimental modal analysis*, vol. 15, no. 5, pp. 132–135, 2023, [Online]. Available: https://www.researchgate.net/publication/380597055
[17] A. M. Uddin, A. Al Miraj, M. Sen Sarma, A. Das, and M. M. Gani, "Traditional Bengali Food Classification Using Convolutional Neural Network," in *TENSYMP 2021 - 2021 IEEE Region 10 Symposium*, Institute of Electrical and Electronics Engineers Inc., Aug. 2021. https://doi.org/10.1109/TENSYMP52854.2021.9550874.
[18] M. A. Rasyidi, Y. S. Mardhiyyah, Z. Nasution, and C. H. Wijaya, "Performance comparison of state-of-the-art deep learning model architectures in Indonesian food image classification," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 5, pp. 3355–3368, Oct. 2024, https://doi.org/10.11591/eei.v13i5.7996.
[19] M. Salvi, U. R. Acharya, F. Molinari, and K. M. Meiburger, "The impact of pre- and post-image processing techniques on deep learning frameworks: A comprehensive review for digital pathology image analysis," *Elsevier Ltd.*, 2021, https://doi.org/10.1016/j.compbiomed.2020.104129.
[20] M. M. Ahsan, T. E. Alam, T. Trafalis, and P. Huebner, "Deep MLP-CNN model using mixed-data to distinguish between COVID-19 and Non-COVID-19 patients," *Symmetry (Basel)*, vol. 12, no. 9, pp. 1–14, Sep. 2020, https://doi.org/10.3390/sym12091526.
[21] A. Lalande *et al.*, "Deep Learning methods for automatic evaluation of delayed enhancement-MRI. The results of the EMIDEC challenge," *Med Image Anal*, vol. 79, no. 102428, pp. 1–13, Jul. 2022, https://doi.org/10.1016/j.media.2022.102428.
[22] K. Tong, Y. Wu, and F. Zhou, "Recent advances in small object detection based on deep learning: A review," *Image Vis Comput*, vol. 97, no. 103910, pp. 1–14, May 2020, https://doi.org/10.1016/j.imavis.2020.103910.
[23] O. Nassar, *Data Leakage in Machine Learning*. 2020. http://dx.doi.org/10.13140/RG.2.2.27468.59528.
[24] H. Abdel-Jaber, D. Devassy, A. Al Salam, L. Hidaytallah, and M. El-Amir, "A Review of Deep Learning Algorithms and Their Applications in Healthcare," *Algorithms*, vol. 15, no. 2, 2022, https://doi.org/10.3390/a15020071.
[25] M. Abbasi, A. Shahraki, and A. Taherkordi, "Deep Learning for Network Traffic Monitoring and Analysis (NTMA): A Survey," *Comput Commun*, vol. 170, pp. 19–41, Mar. 2021, https://doi.org/10.1016/j.comcom.2021.01.021.
[26] A. Wahid, K. Ullah, S. I. Ullah, M. Amin, S. Almutairi, and M. Abohashrh, "sEMG-based Upper Limb Elbow Force Estimation Using CNN, CNN-LSTM, and CNN-GRU Models," *IEEE Access*, vol. 12, 2024, https://doi.org/10.1109/ACCESS.2024.3451209.
[27] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J Big Data*, vol. 8, no. 53, pp. 1–74, Dec. 2021, https://doi.org/10.1186/s40537-021-00444-8.
[28] L. Jiang *et al.*, "Numerical investigation of the effective receptive field and its relationship with convolutional kernels and layers in convolutional neural network," *Front Mar Sci*, vol. 11, Oct. 2024, https://doi.org/10.3389/fmars.2024.1492572.
[29] E. A. Mohamed, T. Gaber, O. Karam, and E. A. Rashed, "A Novel CNN pooling layer for breast cancer segmentation and classification from thermograms," *PLoS One*, vol. 17, pp. 1–18, Oct. 2022, https://doi.org/10.1371/journal.pone.0276523.
[30] M Mesran, Sitti Rachmawati Yahya, Fifto Nugroho, and Agus Perdana Windarto, "Investigating the Impact of ReLU and Sigmoid Activation Functions on Animal Classification Using CNN Models," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 8, no. 1, pp. 111–118, Feb. 2024, https://doi.org/10.29207/resti.v8i1.5367.
[31] Y. Lee, "The CNN: The Architecture Behind Artificial Intelligence Development," *Journal of Student Research*, vol. 12, no. 4, pp. 1–12, 2023, https://doi.org/10.47611/jsrhs.v12i4.5579.
[32] J. Pardede and A. S. Purohita, "The Advantage of Transfer Learning with Pre-Trained Model in CNN Towards Ct-Scan Classification," *Khazanah Informatika (Jurnal Ilmu Komputer dan Informatika)*, vol. 9, no. 2, pp. 155–161, Oct. 2023, https://doi.org/10.23917/khif.v9i2.19872.
[33] Y. Gao, P.-C. Luan, and A. Alahi, "Multi-Transmotion: Pre-trained Model for Human Motion Prediction," *ArXiv*, 2024, https://doi.org/10.48550/arXiv.2411.02673.
[34] A. Philip, "Optimizing Food101 Classification with Transfer Learning: A Fine-Tuning Approach Using EfficientNetB0," *International Journal of Intelligent Information Systems*, vol. 13, no. 4, pp. 59–77, Aug. 2024, https://doi.org/10.11648/j.ijiis.20241304.11.
[35] K. Kansal, T. B. Chandra, and A. Singh, "ResNet-50 vs. EfficientNet-B0: Multi-Centric Classification of Various Lung Abnormalities Using Deep Learning," *Procedia Comput Sci*, vol. 235, pp. 70–80, 2024, https://doi.org/10.1016/j.procs.2024.04.007.

[36] J. Cao, M. Yan, Y. Jia, X. Tian, and Z. Zhang, "Application of a modified Inception-v3 model in the dynasty-based classification of ancient murals," *EURASIP J Adv Signal Process*, vol. 49, 2021, https://doi.org/10.1186/s13634-021-00740-8.
[37] G. Vrbančič and V. Podgorelec, "Transfer learning with adaptive fine-tuning," *IEEE Access*, vol. 8, pp. 196197–196211, Oct. 2020, https://doi.org/10.1109/ACCESS.2020.3034343.
[38] S. Ben Brahim, S. Dardouri, and R. Bouallegue, "Brain Tumor Detection Using a Deep CNN Model," *Applied Computational Intelligence and Soft Computing*, vol. 2024, no. 1, Jan. 2024, https://doi.org/10.1155/2024/7634426.
[39] L. Ali, F. Alnajjar, H. Al Jassmi, M. Gochoo, W. Khan, and M. A. Serhani, "Performance evaluation of deep CNN-based crack detection and localization techniques for concrete structures," *Sensors*, vol. 21, no. 5, pp. 1–22, Mar. 2021, https://doi.org/10.3390/s21051688.
[40] M. Owusu-Adjei, J. Ben Hayfron-Acquah, T. Frimpong, and G. Abdul-Salaam, "Imbalanced class distribution and performance evaluation metrics: A systematic review of prediction accuracy for determining model performance in healthcare systems," *PLOS Digital Health*, vol. 2, no. 11 November, Nov. 2023, https://doi.org/10.1371/journal.pdig.0000290.
[41] Y. S. Taspinar, I. Cinar, and M. Koklu, "Classification by a stacking model using CNN features for COVID-19 infection diagnosis," *J Xray Sci Technol*, vol. 30, no. 1, pp. 73–88, 2022, https://doi.org/10.3233/XST-211031.
[42] D. F. Anas, I. Jaya, and Y. Herdiyeni, "Measurement and Analysis of Detecting Fish Freshness Levels Using Deep Learning Method," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 18, no. 4, pp. 1–12, 2024, https://doi.org/10.22146/ijccs.95054.
[43] Ž. Vujović, "Classification Model Evaluation Metrics," *(IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, pp. 599–606, 2021, https://doi.org/10.14569/IJACSA.2021.0120670.
[44] S. Suyahman, S. Sunardi, M. Murinto, and A. N. Khusna, "Siamese Neural Network Optimization Using Distance Metrics for Trademark Image Similarity Detection," *International Journal of Computing and Digital Systems*, vol. 17, no. 1, 2024, http://dx.doi.org/10.13140/RG.2.2.11236.59522.
[45] S. A. Hicks *et al.*, "On evaluation metrics for medical applications of artificial intelligence," *Sci Rep*, vol. 12, no. 5979, pp. 1–9, 2022, https://doi.org/10.1038/s41598-022-09954-8.
[46] T. Thanchomnang, N. Chaibutr, W. Maleewong, and P. Janwan, "Automatic detection of Opisthorchis viverrini egg in stool examination using convolutional-based neural networks," *PeerJ*, vol. 12, 2024, http://dx.doi.org/10.7717/peerj.16773.
[47] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 6, pp. 1–13, Jan. 2020, https://doi.org/10.1186/s12864-019-6413-7.
[48] S. Farhadpour, T. A. Warner, and A. E. Maxwell, "Selecting and Interpreting Multiclass Loss and Accuracy Assessment Metrics for Classifications with Class Imbalance: Guidance and Best Practices," *Remote Sens (Basel)*, vol. 16, no. 3, pp. 1–22, Feb. 2024, https://doi.org/10.3390/rs16030533.
[49] M. Grandini, E. Bagli, and G. Visani, "Metrics for Multi-Class Classification: an Overview," *ArXiv*, pp. 1–17, 2020, https://doi.org/10.48550/arXiv.2008.05756.
[50] M. Chun, H. Jeong, H. Lee, T. Yoo, and H. Jung, "Development of Korean Food Image Classification Model Using Public Food Image Dataset and Deep Learning Methods," *IEEE Access*, vol. 10, pp. 128732–128741, 2022, https://doi.org/10.1109/ACCESS.2022.3227796.
[51] A. Wulandari, "Comparing CNN Architecture for Indonesian Speciality Cuisine Classification," *Engineering, MAthematics and Computer Science Journal (EMACS)*, vol. 6, no. 1, pp. 55–60, Jan. 2024, https://doi.org/10.21512/emacsjournal.v6i1.11076.
[52] S. Ittisoponpisan, C. Kaipan, S. Ruang-On, R. Thaiphan, and K. Songsri-In, "Pushing the Accuracy of Thai Food Image Classification with Transfer Learning," *Engineering Journal*, vol. 26, no. 10, pp. 57–71, Oct. 2022, http://dx.doi.org/10.4186/ej.2022.26.10.57.
[53] D. D. Rao and N. P. Ranga, "Indian Food Image Classification Using K-Nearest-Neighbour and Support-Vector-Machines," *International Journal For Advanced Research In Science & Technology (IJARST)*, vol. 9, no. 6, pp. 201–208, Jun. 2021, Accessed: Oct. 22, 2024. [Online]. Available: https://www.ijarst.in/public/uploads/paper/224541651315848.pdf
[54] E. Tasci, "Voting combinations-based ensemble of fine-tuned convolutional neural networks for food image recognition," *Multimed Tools Appl*, vol. 79, no. 79, pp. 30397–30418, 2020, https://doi.org/10.1007/s11042-020-09486-1.

BIOGRAPHY OF AUTHORS

Muhammad Ridha Maulidi is an undergraduate student in the Department of Computer Science at Lambung Mangkurat University, class of 2021. His research interest is focused on Data Science. He can be contacted at: mridha230502@gmail.com. Orcid ID: 0009-0001-9924-1124.

Fatma Indriani is a lecturer in the Department of Computer Science at Lambung MangkuratUniversity. Her research interest is focused on Data Science. Before becoming a lecturer, she completed her undergraduate program in the Informatics Department at Bandung Institute of Technology. In 2008, she started working as a lecturer in the Computer Science department at Lambung Mangkurat University. She then completed her master's degree at Monash University, Australia,in 2012. Her latest education is a doctorate in Bioinformatics at Kanazawa University, Japan, which was completed in 2022. The research fields she focuses on are Data Science and Bioinformatics. She can be contacted at: f.indriani@ulm.ac.id. Orcid ID: 0009-0006-7180-6708.

Andi Farmadi is a senior lecturer in the Departement of Computer Science at Lambung Mangkurat University. His research area focused on Data Science. He completed his bachelor’s degree in Insrumental Physics at Hasanuddin University in 1999. In 2007, he pursued a master’s degree in Informatics at Bandung Institute of Technology. He can be contacted at: andifarmadi@ulm.ac.id. Orcid ID: 0009-0009-0926-8082.

Irwan Budiman successfully finished his bachelor's degree in the informatics department at the Islamic University of Indonesia. Subsequently, he assumed the role of a lecturer in Computer Science at Universitas Lambung Mangkurat starting in 2008. Additionally, in 2010, he pursued a master’s degree in information systems at Diponegoro University. His area of research expertise lies in Data Science. He can be contacted at: irwan.budiman@ulm.ac.id. Orcid ID: 0000-0002-0514-7429.

Dwi Kartini received her Bachelor's and Master's degrees in computer science from the Faculty of Computer Science, Putra Indonesia University "YPTK" Padang, Indonesia. Her research interests include the applications of artificial intelligence and data mining. She is an Assistant Professor of the Department of Computer Science, Faculty of Mathematics and Natural Sciences, Lambung Mangkurat University in Banjarbaru, Indonesia. She can be contacted at: dwikartini@ulm.ac.id. Orcid ID: 0000-0002-7382-5084.