JURNAL RISET INFORMATIKA
Vol. 8, No. 1. December 2025
P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i1.450
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

APPLICATION OF TRANSFER LEARNING ON EFFICIENTNET-B0 ARCHITECTURE FOR AUTOMATIC ROOF TILE DAMAG CLASSIFICATION

Rayhan Prasetya Ady [-1], Arif Pramudwiatmoko [-2]

Informatics
Department of Science and Technology
Universitas Teknologi Yogyakarta
rayhan.5220411267@student.uty.ac.id [-1] , arif.pramudwiatmoko@uty.ac.id[-2]

Abstract

Subjectivity in manual quality control for traditional roof tiles poses a significant challenge, as the current process relies on manual, visual inspection and subjective judgment. This research proposes an automatic system to classify tile quality from images using a Convolutional Neural Network (CNN), specifically the EfficientNet-B0 model enhanced with transfer learning. The study utilized a primary dataset comprising 616 local roof tile images collected directly from producers in Berjo Kidul, Godean, Yogyakarta. These images were manually labeled based on producer criteria into three distinct classes: 'Finished' (203 images), 'Underbaked' (213 images), and 'Broken/Cracked' (200 images). The methodology involved resizing all images to 224x224 pixels and applying data augmentation, including random rotation, horizontal flipping, and color jitter, to mitigate overfitting. The EfficientNet-B0 model, pre-trained on ImageNet, was implemented in PyTorch and trained for 10 epochs using an 80/20 train/validation split with the Adam optimizer. The model demonstrated outstanding performance, reaching 99.70% accuracy in validation. Further evaluation confirmed this robustness; the model perfectly identified the 'Underbaked' class and recorded only a single misclassification error on the test set. Qualitative analysis via a Flutter mobile application showed the system is resilient to changes in background and viewing angles, although its accuracy is compromised by poor lighting and strong shadows. This study validates the proposed system as a highly efficient and objective tool for a more reliable quality control process.

Keywords: Image Classification; Deep Learning; Convulational Neural Network; EfficientNet-B0; Transfer Learning; Rooftiles Quality

Abstrak

Subjektivitas pada kontrol kualitas manual untuk genteng tradisional menimbulkan tantangan signifikan, karena proses saat ini bergantung pada inspeksi visual manual dan penilaian subjektif. Penelitian ini mengusulkan sistem otomatis untuk mengklasifikasikan kualitas genteng dari gambar menggunakan Convolutional Neural Network (CNN), khususnya model EfficientNet-B0 yang ditingkatkan dengan transfer learning. Studi ini menggunakan dataset primer yang terdiri dari 616 gambar genteng lokal yang dikumpulkan langsung dari produsen di Berjo Kidul, Godean, Yogyakarta. Gambar-gambar ini dilabeli secara manual berdasarkan kriteria produsen ke dalam tiga kelas berbeda: 'Matang' (203 gambar), 'Kurang Matang' (213 gambar), dan 'Pecah/Retak' (200 gambar). Metodologi yang digunakan melibatkan perubahan ukuran semua gambar menjadi 224x224 piksel dan penerapan augmentasi data, termasuk rotasi acak, horizontal flipping, dan color jitter, untuk mengurangi overfitting. Model EfficientNet-B0, yang telah dilatih sebelumnya pada ImageNet, diimplementasikan menggunakan PyTorch dan dilatih selama 10 epoch menggunakan pembagian 80/20 untuk data latih/validasi dengan optimizer Adam. Model menunjukkan performa luar biasa, mencapai akurasi 99,70% pada data validasi. Evaluasi lebih lanjut mengkonfirmasi ketangguhan ini; model secara sempurna mengidentifikasi kelas 'Kurang Matang' dan hanya mencatat satu kesalahan klasifikasi pada data uji. Analisis kualitatif melalui aplikasi seluler Flutter menunjukkan bahwa sistem ini tangguh terhadap perubahan latar belakang dan sudut pandang, meskipun akurasinya terpengaruh oleh pencahayaan yang buruk dan bayangan yang kuat. Studi ini memvalidasi sistem yang diusulkan sebagai alat yang sangat efisien dan objektif untuk proses kontrol kualitas yang lebih andal.

***Kata kunci**: Klasifikasi Citra; Deep Learning; Convolutional Neural Network; EfficientNet-B0; Transfer Learning; Kualitas Genteng*

INTRODUCTION

Roof tiles, defined as fired clay materials used for roofing (Achiruddin et al., 2019), are integral to Indonesia's growing housing industry. Clay roof tiles are the most widely used roofing material (Saputra et al., 2023). This growth, driven by population increases and urbanization (Perdamaian & Zhai, 2024) creates a high demand for durable, high-quality tiles. Local producers play a vital role in meeting this demand, but they face a significant obstacle in the quality control (QC) process. Currently, QC is a manual, visual inspection that relies heavily on subjective judgment and experience, a substantial challenge when 500 to 1,000 tiles must be checked per batch. This method can lead to inconsistent assessments and errors in identifying critical quality indicators such as cracks, precise shape, and color (which indicates firing maturity).

Previous computational research on similar industrial items, like ceramic tiles, has explored various methods. Traditional machine learning approaches include. Alamsyah et al. (2019) used k-nearest neighbors (KNN) to detect tile defects with 98.0% accuracy, more recently, Aryo Hardirega et al. (2024) applied a Convolutional Neural Network (CNN), EfficientNet-B1, to classify batik motifs with 0.98 testing accuracy. Zakaria et al. (2024) also used a basic CNN for ceramic tile classification, achieving 83.33% accuracy on a small dataset. However, traditional machine learning and image processing often require complex, manual feature engineering, which may not capture complex visual variations optimally. Xu et al., (2022) employed a deep learning model (Mask R-CNN) to automatically analyze aerial photographs and classify roof damage following Typhoon Faxai in Chiba, Japan. The model was trained to specifically detect three key objects: roof outlines, blue tarps, and destroyed roofs. This detection process enabled an automated five-level damage classification (Level 0-4), which was based on the percentage of the roof covered by blue tarps or if it was destroyed. The study's method proved significantly faster than traditional field surveys and yielded an average F-value of 0.818 across the five damage categories. Bahtiar et al.,(2024) developing a Convolutional Neural Network (CNN) model to categorize ceramic roof tiles into several quality classes. The researchers implemented basic preprocessing, including resizing and normalization, and constructed a simple CNN architecture. After training the model on an internal dataset, they reported a peak performance of approximately 83% validation accuracy. The authors also acknowledged the limitations posed by a small dataset and recommended using data augmentation in future work.

Deep Learning, especially CNNs, offers a distinct advantage. Defined as a branch of machine learning using deep neural networks (Mardiyah, 2020), CNNs are powerful tools for image classification and object detection (Purwono et al., 2022). They revolutionize inspection by performing automatic, hierarchical feature extraction, learning to identify relevant patterns from simple edges to complex textures directly from image data (Khan et al., 2019). This study specifically adopts EfficientNet-B0. A modern architecture known for balancing network depth, width, and resolution to achieve high accuracy with low computational cost (Tan & Le, 2019). EfficientNet-B0 is highly effective for specific tasks with limited datasets (Putra & Akbar, 2024). EfficientNet demonstrates strong capabilities in image analysis and can accurately classify different categories (Bai et al., 2025). EfficientNet-B0 achieves a balance between performance and efficiency by simultaneously scaling depth, width, and resolutionddd(Zhang et al., 2024). To enhance performance on a limited dataset, we apply transfer learning, a technique that repurposes a model pre-trained on a large dataset (ImageNet) as a starting point for a new, related task (Hosna et al., 2022). Transfer learning significantly improves accuracy in classification compared to CNN models without transfer learning (Rahman et al., 2025). The use of fine tuning in model training is considered more effective and efficient than manual model training (Rozi et al., 2023). Deep Learning has been proven to provide accuracy improvements comparable to the addition of data. (Suyanto, 2022). PyTorch is popular for its ease of programming and debugging, using both static and dynamic computational graphs (Xia & Zhou, 2022).

This research differs from previous work. Unlike Zakaria et al., it uses the advanced EfficientNet-B0 architecture with transfer learning for traditional roof tiles, classifying them as 'Finished', 'Underbaked', or 'Broken/Cracked'. Unlike, Alamsyah et al., and Bahtiar et al., It employs a deep learning approach for automatic feature extraction, avoiding the manual feature engineering of traditional ML. Finally, while Hardirega et al. used a similar architecture, this study applies it to a different domain (industrial QC) and uses the lighter 'B0' variant to test its effectiveness. Xu et al., use all part of the roof for the dataset, this research only use single part of the tiles to clasify the quality.

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i1.450
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

Consequently, this research aims to develop a fast, consistent, and accurate roof tile quality classification system using the EfficientNet-B0 architecture with transfer learning. This system is intended to enhance the efficiency and objectivity of quality assessment and serve as a reference for similar industrial applications

RESEARCH METHODS

This study employs a quantitative approach structured by the Cross-Industry Standard Process for Data Mining (CRISP-DM) to ensure a systematic workflow, from initial problem understanding through to model implementation. CRISP-DM is a comprehensive data mining methodology and process model that offers a complete guide for executing a data mining project (Shearer, 2000). It divides the project lifecycle into six phases: business understanding, data understanding, data preparation, modeling, evaluation, and deployment (Asyraf & Prasetya, 2024). The CRISP-DM application in this research can be see in Figure 1.

May 22, this date considered less rain occured in Berjo Kidul Godean Sleman Yogyakarta.

Research Target / Subject

Research targets the rooftiles that Berjo Kidul rooftiles maker produce after the firing session of rooftiles held.

Procedure

This research implemented a non-experimental quantititative study with take the data straight with the smartphone camera. The specific CRISP-DM workflow for this research is illustrated in Figure 1.

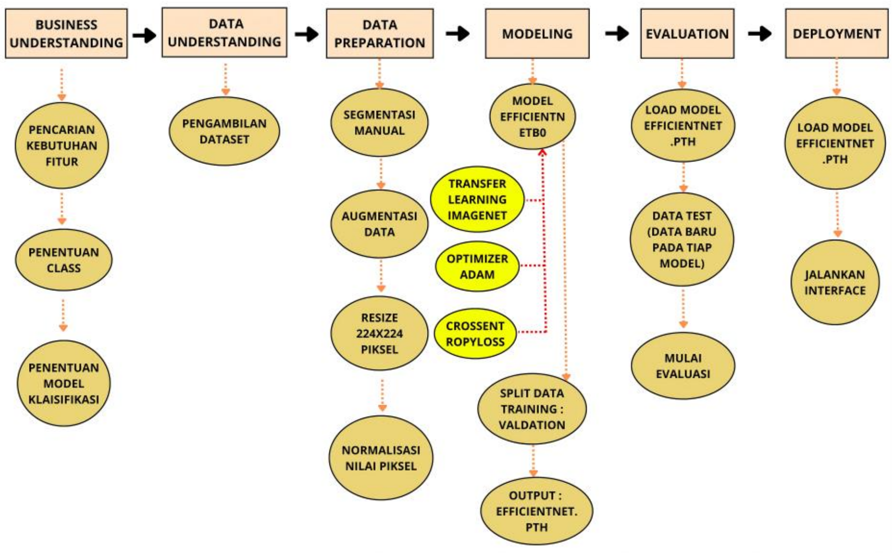

Figure 1. CRISP-DM Processes

Types of research

This research uses a quantitative approach CRISP-DM to ensure systematic workflow, from initial problem understanding through to model implementation.

Time and Place of Research

Research starts from data collection process that was conducted between April 24 to

Data, Instruments, and Data Collection Techniques

The dataset employed in this research comprises primary data collected directly from traditional roof tile producers in Berjo Kidul, Godean, Yogyakarta. The data collection process was conducted between April 24 and May 22, 2025, using a smartphone camera (POCO M5) under natural daylight conditions. A total of 616 roof tile

Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

images were successfully gathered. The image that has been taken is then cropped to make it uniform. All images were manually labeled based on the producers' criteria into three quality classes, some of which are shown in Figures 2, 3, and 4. The dataset includes: 'Finished' (203 images), 'Underbaked' (213 images), and 'Broken/Cracked' (200 images).

Figure 2. Finished

Figure 3. Underbaked Rooftiles

Figure 4. Broken/Cracked Rooftiles

Preprocessing Dataset

How To enhance the training data and mitigate overfitting, a process of preprocessing and augmentation was applied. The application of augmentation to the algorithm helps the model work better (Rhamadiyanti & Kurini, 2024). The preprocessing stage involved resizing all images to 224x224 pixels, the standard input for EfficientNet-B0 and normalizing their pixel values. Following this, the training data was augmented using random rotation, horizontal flipping, and color jitter to introduce more variability. The effect of this augmentation is illustrated in Figure 5.

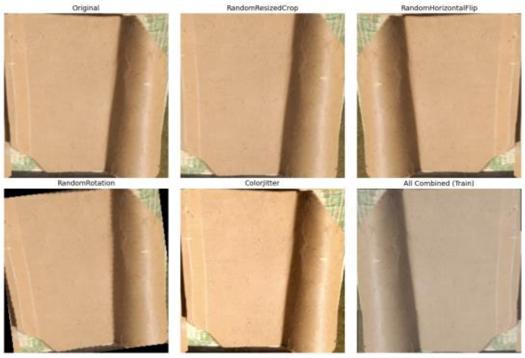

Figure 5. Augmentation Comparation

This research utilized the PyTorch framework to implement an EfficientNet-B0 model, leveraging transfer learning from a model pre-trained on ImageNet. The pre-trained backbone was used for feature extraction, and its classifier head was replaced with a new structure. This new head includes a Fully Connected Layer and a Softmax activation for the three-class classification, preceded by a Dropout layer to mitigate overfitting. The dataset was divided into an 80/20 train/validation split. The model was trained for 10 epochs using a batch size of 32, the Adam optimizer (learning rate 0.001), and Cross-Entropy Loss. The best-performing model on the validation set was saved for evaluation. This complete architecture and its parameters are presented in Table 1.

Table 1. EfficientNet-B0 Architecture

Blok	Main layer Type	Output Shape	Total Params
1. Stem	Conv2d, BatchNorm2d, SiLU	[-1, 32, 112, 112]	928
2. Blok MBConv (1x)	MBConv, SqueezeExcitation	[-1, 16, 112, 112]	1.448
3. Blok MBConv (2x)	MBConv, SqueezeExcitation	[-1, 24, 56, 56]	8.96
4. Blok MBConv (2x)	MBConv, SqueezeExcitation	[-1, 40, 28, 28]	27.08

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i1.450
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

5. Blok MBConv (3x)	MBConv, SqueezeExcitation	[-1, 80, 14, 14]	155.04
6. Blok MBConv (3x)	MBConv, SqueezeExcitation	[-1, 112, 14, 14]	316.928
7. Blok MBConv (4x)	MBConv, SqueezeExcitation	[-1, 192, 7, 7]	971.968
8. Blok MBConv (1x)	MBConv, SqueezeExcitation	[-1, 320, 7, 7]	370.12
9. Classification Head	Conv2d, AvgPool, Dropout, Linear	[-1, 3]	416.003

The model's performance was quantitatively evaluated using an unseen test set. We used a Confusion Matrix to analyze the model's effectiveness, as it compares the actual classifications with the model's predicted classifications. This comparison yields four primary values: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) In the context of this study, a TP is a correct 'Finished' classification, while an FP is when a 'Broken' tile is incorrectly labeled as 'Finished'. Similarly, a TN is a correct rejection of a class, and an FN is when a 'Finished' tile is missed. These four outputs are the foundation for computing key metrics such as accuracy, precision, recall, and F1-score.

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

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

$$\text{F1-Score} = \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

To test and deploy the trained classification model, a cross-platform mobile application was developed using the Flutter framework and the Dart programming language. This platform enables developers to create multiplatform applications from a single codebase, allowing the app to be used across various platforms, including Android, iOS, web, and desktop (Fahriza et al., 2024). Flutter offers greater precision in development; application creation time is reduced as it eliminates the need to intensively synchronize two separate files (like .java and .xml), which often requires developers to switch back and forth (Japara & Arifin, 2023).

The application is designed with a client-server architecture. The app functions as the client, responsible for capturing images and sending them to the server for processing. The deep learning model is hosted on a server and accessed via an API. The application sends the user-selected roof tile image through an HTTP POST request to the server endpoint. The server then processes the image using the EfficientNet-B0 model and returns the prediction result in JSON format, which includes the classification label ('Finished', 'Underbaked', or 'Broken/Cracked'). The results also calculates a probability for each of the three classes. The class with the highest probability is chosen as the final prediction, and that probability is its "confidence score". Key implemented features include live image capture via the camera, image selection from the device gallery, real-time display of classification results, and a classification history for review.

RESULTS AND DISCUSSION

Results

The experiment was run on Google Colaboratory, utilizing Python 3.10.12, PyTorch 2.3.0, and CUDA 12.1. We split the 616-image dataset into an 80% training set (492 images) and a 20% validation set (124 images). During the 10-epoch training process, the EfficientNet-B0 model demonstrated excellent performance. Both training and validation accuracy rose consistently, converging at 99.7%. Similarly, both loss curves steadily decreased to very low values. These results suggest the model learned the tile features effectively without significant overfitting.

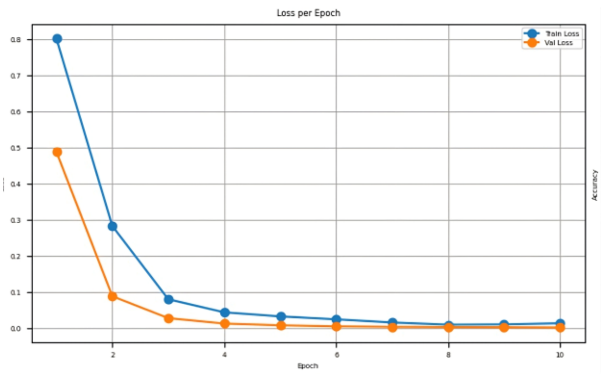

Figure 6. Loss for Train and Validaton Graph

The model evaluation was conducted on a total of 328 test images that the model had not previously seen. The results of the model's evaluation report on this test data are presented in Tables 2 and 3. The 'support' column indicates the number of evaluation samples

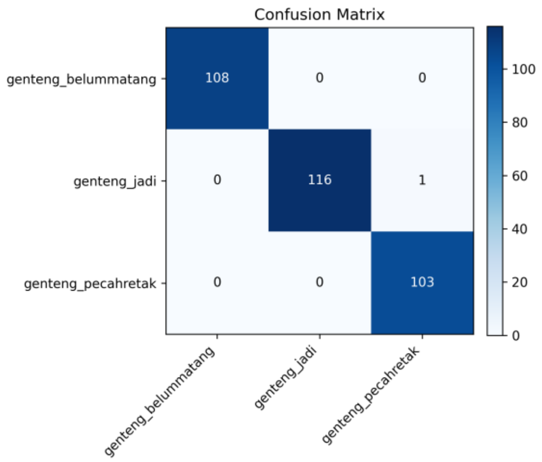

Figure 7. Confusion Matrix

Table 2. Evaluation of each Class

Class	Precision	Recall	f1-score	Support
genteng_belummatang	1.0000	1.0000	1.0000	108
genteng_jadi	1.0000	0.9915	0.9957	117
genteng_pecahretak	0.9904	1.0000	0.9952	103

Table 3. Evaluation All Class

Matrix		Value		Support
Accuracy		0.9970		328
Macro avg	0.9968	0.9972	0.9970	328
Weighted avg	0.9970	0.9970	0.9970	328

Discussion

Based on the analysis in Figure 6, the training process for the roof tile quality classification model proved to be highly successful and effective. The training graph shows a consistently decreasing loss curve without any indication of overfitting, which signals that the model has excellent generalization capabilities for new data and has reached its optimal performance.

This is reinforced by the test results in the confusion matrix in Figure 7; the results on the main diagonal (108, 116, 103) show the model successfully predicted nearly all the data correctly. The only recorded error was a single 'Finished' tile image that was mistakenly classified as 'Broken/Cracked', which caused the recall and precision values for those two classes to be imperfect. Aside from this minor anomaly, the model demonstrated very high accuracy and experienced no confusion in distinguishing between the quality classes.

According to the evaluation results in Tables 2 and 3, the tile classification model exhibited very high performance, achieving an overall accuracy of 99.70%. A per-class analysis revealed that the model perfectly identified the 'Underbaked' class with precision, recall, and f1-score values of 1.00. For the 'Finished' class, the model had perfect precision, although one sample was missed (recall 9.915). Conversely, in the 'Broken/Cracked' class, the model was able to find all relevant samples (recall 1.00), but there was one prediction error (precision 9.904) for a broken/cracked tile.

To understand the influence of the training data volume on model performance, an experiment was conducted with three different data split scenarios: 80:20, 70:30, and 60:40 for training and validation data, respectively. The model was trained in each scenario using the same hyperparameters and was subsequently evaluated using an identical test set.

Table 4. Accuracy Comparasion each Split

Split (Train:Val)	Validation Accuracy (%)	Test Accuracy (%)
80% : 20%	98.81	99.70
70% : 30%	98.48	98.48
60% : 40%	97.70	97.56

Table 5. Accuracy Comparasion each Split without Augmentation

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i1.450
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

Split (Train:Val)	Validation Accuracy (%)	Test Accuracy (%)
80% : 20%	82.79	76.60
70% : 30%	79.87	73.34
60% : 40%	71.62	60.54

From Tables 4 and 5, a clear trend is evident: the larger the proportion of data used for training, the higher the accuracy achieved on both the validation and test data. The scenario with an 80:20 split, which utilized data augmentation, provided the best results, reaching a test accuracy of 99.70%. This confirms the importance of data volume in training deep learning models to generalize more effectively. The performance decrease in the 60:40 and 70:30 scenarios, as shown in both tables, indicates that with less training data, the model had a reduced opportunity to learn all the feature variations present in the dataset. Therefore, the 80:20 scenario with augmentation was selected as the final configuration for this research model.

To further test the model's robustness and generalization, a series of qualitative trials were conducted using the Flutter mobile application. This testing utilized new, randomly captured tile images under intentionally varied and non-ideal conditions. Testing via the application also served to directly identify challenges for an effective and practical automatic classification deployment. The non-ideal conditions tested included variations in the image background (different from training data), different shooting angles, break/crack types not seen in the training data, shadows obscuring the image, capturing more than one tile, and changes in position (rotation). 21 samples of the condition

Table 6. Classification Testing with non-Ideal Conditions

Description	Classification	Confidence (%)
A cracked tile with a background different from the training images, shot at a 0° angle from normal.	genteng_pecahretak	99.03
A broken tile in landscape orientation with a background different from the training images, shot at a 30° angle from normal.	genteng_pecahretak	73.87
A broken tile with the break in a different location and a background different from the training images, shot at 0° from normal.	genteng_pecahretak	99.68
A broken tile in landscape orientation with a background different from the training images, shot at a 50° angle from normal.	genteng_pecahretak	86.69
A finished tile photographed from a position higher than the ideal shooting position.	genteng_jadi	97.73
A finished tile rotated 180° from its normal position, shot at a 0° angle from normal.	genteng_jadi	82.00
A finished tile rotated 90° to the right, shot at	genteng_jadi	73.84

P-ISSN: 2656-1743 | E-ISSN: 2656-1735 JURNAL RISET INFORMATIKA
DOI: https://doi.org/10.34288/jri.v8i1.450 Vol. 8, No. 1. December 2025
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

a 60° angle from normal.		
A finished tile rotated 90° to the left, shot at a 60° angle from normal.	genteng_jadi	81.63
A broken tile in landscape orientation with an intense shadow covering 40% of the tile.	genteng_jadi	43.20
Nine unfinished tiles photographed simultaneously in one frame, shot at a 0° angle from normal.	genteng_belummatang	76.41
Three finished tiles photographed simultaneously in one frame, shot at a 30° angle from normal.	genteng_jadi	83.68
A cracked tile where the camera position was lowered from ideal, showing only the crack.	genteng_pecahretak	49.21
A cracked tile photographed from the bottom edge, but still showing the full image of the tile.	genteng_pecahretak	49.44
A broken tile in landscape orientation, shot from a higher position than ideal and at a 60° angle from normal.	genteng_jadi	75.00
A broken tile shot from the right edge and from a higher position than ideal, at a 60° angle from normal.	genteng_jadi	49.14
A cracked tile with a faint shadow covering 70% of the tile, shot at a 0° angle from normal.	genteng_jadi	55.04
An unfinished tile photographed in dimmer lighting than ideal, shot at a 0° angle from normal.	genteng_belummatang	98.36
An unfinished tile with a vertical shadow covering 20% of it, shot at a 0° angle from normal.	genteng_belummatang	95.84
Five unfinished tiles, rotated 180°, photographed simultaneously in one frame at a 0° angle from normal.	genteng_belummatang	92.57

Three unfinished tiles in normal position and two unfinished tiles rotated 180°, photographed simultaneously in one frame at a 0° angle from normal.	genteng_belummatang	75.50
A cracked tile rotated 60° to the left, shot at a 0° angle from normal.	genteng_pecahretak	99.14

Based on the trial results in table 6, the model showed good robustness against variations in background, object position, and partial shadows, where it was still able to provide accurate predictions. Nevertheless, several critical limitations were also identified. The most significant failure was the misclassification of damaged tiles as finished tiles, especially under difficult conditions such as images taken from a distance, at an angle, or when cracks were covered by shadows. The model also showed difficulty with complex scenes containing many objects.

However, a positive finding was that in almost all failure cases, the model returned a low confidence score. In this system, the class with the highest probability is chosen as the final prediction, and that probability is its "confidence score." The trials revealed that even when the model's final prediction was incorrect, this "highest" probability was still notably low, indicating the model was "unsure" of its decision. This opens the opportunity to implement a confidence threshold to filter dubious results for manual re-examination. In short, these trials prove that although highly accurate on ideal data, the model's performance can degrade under challenging lighting and image composition conditions, providing valuable insights for future improvement.

CONCLUSIONS AND SUGGESTIONS

Conclusion

Based on the research conducted, the application of the EfficientNet-B0 architecture with transfer learning was successfully implemented for the automatic classification of roof tile quality. The developed model demonstrated highly reliable performance, achieving an accuracy of 99.70% on the test data. The model was able to perfectly classify 'Underbaked' tiles and made only a single error in the other classes. Overall, this research proves that a deep learning-based automatic classification system is an effective solution for overcoming the subjectivity and inconsistency inherent in the traditional roof tile quality control process.

Suggestion

For future research, it is recommended to enhance the dataset by increasing its volume and variety, particularly by including more images from diverse producers and under challenging conditions where the model showed weakness, such as low light and strong shadows. From a technical perspective, the application's practicality could be improved by converting the model for on-device deployment (e.g., TensorFlow Lite) to enable offline, real-time classification. Implementing a confidence threshold could also create a 'human-in-the-loop' system, flagging low-confidence predictions for manual review. Finally, the research scope could be expanded from classification to more granular object detection or segmentation, allowing the system to not only identify a tile as defective but also to pinpoint the specific location and type of the damage.

REFERENCES

Achiruddin, Maghfirah, A., Sembiring, A. D., & Sofyan, H. D. (2019). The Fabrication of Roof Tiles Utilizing Palm Oil Boiler Ash and Used Rubber Thread Fibers Waste. *Journal of Technomaterials Physics *Corresponding author at: Jl. Bioteknologi No.1 Kampus USU*, *1*(2), 102–109.

Alamsyah, R., Wiranata, A. D., & Rafie, R. (2019). Deteksi Cacat Ubin Keramik Dengan Metode K-Nearest Neighbor. *Techno.Com*, *18*(3), 245–250. https://doi.org/10.33633/tc.v18i3.2459

Asyraf, H., & Prasetya, M. (2024). Implementasi Metode CRISP DM dan Algoritma Decision Tree Untuk Strategi Produksi Kerajinan Tangan pada UMKM A. *Jurnal Media*

Informatika Budidarma, *8*, 94. https://doi.org/10.30865/mib.v8i1.7050

Bahtiar, Y., Maulindar, J., & Yuliana, M. E. (2024). Prediksi Kualitas Genteng Mantili Berdasarkan Komposisi Bahan Baku Menggunakan Algoritma K-Nearest Neighbour. *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, *9*(4), 2030–2041. https://doi.org/10.29100/jipi.v9i4.5589

Bai, K., Zhang, Z., Jin, S., & Dai, S. (2025). Rock image classification based on improved EfficientNet. *Scientific Reports*, *15*(1), 18683–18697. https://doi.org/10.1038/s41598-025-03706-0

Fahriza, R., Setyadi, H. J., & Widagdo, P. P. (2024). Pengembangan Aplikasi Mobile E-Tourism Berbasis Flutter Untuk Dinas Pariwisata Kabupaten Kutai Kartanegara Menggunakan Metode Rapid Application Development. *Kreatif Teknologi dan Sistem Informasi (KRETISI)*, *2*(2), 23–28. https://doi.org/10.30872/kretisi.v2i2.1850

Rozi, M. I. F. R., Adiwijaya, N. O., & Swasono, D. I. (2023). Identifikasi Kinerja Arsitektur Transfer Learning Vgg16, Resnet-50, Dan Inception-V3 Dalam Pengklasifikasian Citra Penyakit Daun Tomat. *Jurnal Riset Rekayasa Elektro*, *5*(2), 145. https://doi.org/10.30595/jrre.v5i2.18050

Hardirega, A., Jaelani, I., & Minarto. (2024). Implementasi Convulational Neural Network (CNN) Klasifikasi Motif Batik menggunakan Efficientnet-b1. *Jurnal Mahasiswa Teknik Informatika*, *8*(5), 10024–11028. https://doi.org/https://doi.org/10.36040/jati.v8i5.10901

Hosna, A., Merry, E., Gyalmo, J., Alom, Z., Aung, Z., & Azim, M. A. (2022). Transfer Learning: a Friendly Introduction. *Journal of Big Data*, *9*(1), 102. https://doi.org/10.1186/s40537-022-00652-w

Japara, E. M., & Arifin, S. (2023). Android application development using flutter framework: Creation of geolocation system module to validate user location coordinates. *AIP Conference Proceedings*, *2734*(1), 1–9. https://doi.org/10.1063/5.0155337

Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2019). A Survey of the Recent Architectures of Deep Convolutional Neural Networks. *Artificial Intelligence Review*, *53*(8), 5455–5516. https://doi.org/10.1007/s10462-020-09825-6

Mardiyah, M. I. (2020). *Implementasi Deep Learning untuk Image Classification menggunakan Algoritma Convulational Neural Network (CNN) pada Citra Kebun dan Sawah* [Skripsi, Universitas Islam Indonesia]. https://dspace.uii.ac.id/123456789/28083

Perdamaian, L. G., & Zhai, Z. (2024). Status of Livability in Indonesian Affordable Housing. *Architecture*, *4*(2), 281–302. https://doi.org/10.3390/architecture4020017

Purwono, Ma'arif, A., Rahmaniar, W., Fathurrahman, H. I. K., Frisky, A. Z. K., & Haq, Q. M. U. (2022). Understanding of Convolutional Neural Network (CNN): A Review. *International Journal of Robotics and Control Systems*, *2*(4), 739–748. https://doi.org/10.31763/ijrcs.v2i4.888

Putra, I. F. A., & Akbar, H. (2024). Pengembangan Aplikasi Mobile Klasifikasi Penyakit Kulit Berbasis EfficientNet-B0, Arsitektur MVVM dan CI/CD Pipeline. *Jurnal Ilmiah Komputasi*, *23*(4), 579–586. https://doi.org/10.32409/jikstik.23.4.3676

Rahman, A., Salim, M., & Riadi, I. (2025). Klasifikasi Citra Spesies Bunga Di Indonesia Berbasis Convolutional Neural Network Menggunakan Teknik Transfer Learning. *Jurnal Software Engineering and Computational Intelligence*, *2*(02), 92–100. https://doi.org/10.36982/jseci.v2i02.4942

Rhamadiyanti, D. T., & Kurini. (2024). Analisa Performa Convolutional Neural Network dalam Klasifikasi Citra Apel dengan Data Augmentasi. *Kajian Ilmiah Informatika dan Komputer*, *5*(1), 154–162. https://doi.org/DOI 10.30865/klik.v5i1.2023

Saputra, H. A., Sunarsih, E. S., & Siswanto, B. (2023). Analisis Karakteristik Genteng Keramik Hasil Campuran Limbah Abu Ampas Tebu dan Abu Terbang Batubara sebagai Pengganti Sebagian Tanah Lempung. *Indonesian Journal Of Civil Engineering Education*, *8*(2), 27. https://doi.org/10.20961/ijcee.v8i2.70876

Shearer, C. (2000). The CRISP-DM Model: The New Blueprint for Data Mining. *Journal of Data Warehousing*, *5*(4), 13–22. https://www.dw-institute.com

Suyanto. (2022). *Machine Learning Tingkat Dasar dan Lanjut* (2nd ed.). Penerbit Informatika Bandung.

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *Proceedings of the 36th International Conference on Machine Learning*, 2–8. https://doi.org/10.48550

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i1.450
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

Xia, X., & Zhou, S. (2022). Performance Comparison between Pytorch and Mindspore. *International Journal of Database Management Systems*, *14*(02), 14–2. https://doi.org/10.5121/ijdms.2022.14201

Xu, J., Zeng, F., Liu, W., & Takahashi, T. (2022). Damage Detection and Level Classification of Roof Damage after Typhoon Faxai Based on Aerial Photos and Deep Learning. *Applied Sciences*, *12*(10), 4912. https://doi.org/10.3390/app12104912

Zakaria, A. D., Eviyanti, A., Maulina, M. I., & Azinar, W. A. (2024). *Classification of Ceramic Roof Tiles Using the CNN Method* [Skripsi, UMSIDA]. https://doi.org/10.21070/ups.5804

Zhang, Y., Kong, L., Antwi-Afari, M. F., & Zhang, Q. (2024). An Integrated Method Using a Convolutional Autoencoder, Thresholding Techniques, and a Residual Network for Anomaly Detection on Heritage Roof Surfaces. *Buildings*, *14*(9), 2828–2849. https://doi.org/10.3390/buildings14092828