JURNAL RISET INFORMATIKA **P-ISSN: 2656-1743 |E-ISSN: 2656-1735**
Vol. 8, No. 1. December 2025 DOI: https://doi.org/10.34288/jri.v8i1.417
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

TRANSFER LEARNING WITH EFFFICIENTNET-B0 FOR CAT BREED CLASSIFICATION: A COMPARATIVE EVALUATION OF OPTIMIZERS

Aini Azzah[-1], Salamun Rohman Nudin[-2]

Informatics Management
Universitas Negeri Surabaya
aini.21006@mhs.unesa.ac.id[-1], salamunrohman@unesa.ac.id[-2]

Abstract

Cats are widely kept as companion animals and exhibit substantial breed level variation in appearance and behavior that influences their care. This study develops a lightweight, image based classifier for identifying twelve common cat breeds using transfer learning on the EfficientNet-B0 backbone. Experiments contrasted four optimization algorithms (SGD, AdaGrad, RMSProp, and Adam) to identify the training strategy that balances convergence speed and generalization. Model effectiveness was measured with confusion matrix analysis and common classification indicators (accuracy, precision, recall, and F1-score). The best performing setup, EfficientNet-B0 fine tuned with the Adam optimizer attained 92% training accuracy, 89% validation accuracy, and 88% on the held out test partition. Subsequently, we integrated the trained model into a Flask web application, backed by an SQLite database, and conducted black-box testing to assess its functional reliability. All system functions met specifications and runtime predictions corresponded closely to ground truth labels. This platform provides a rapid and accurate tool for cat owners and enthusiasts to identify breeds in real-world scenarios, highlighting the usefulness of transfer learning in a streamlined web based implementation.

Keywords: Cat breed classification; EfficientNet-B0; transfer learning; Flask web applicaton

Abstrak

Kucing banyak dipelihara sebagai hewan pendamping dan menunjukkan variasi antar ras yang signifikan dalam penampilan serta perilaku yang memengaruhi kebutuhan perawatannya. Penelitian ini mengembangkan sebuah klasifikator citra ringan untuk mengidentifikasi dua belas ras kucing umum dengan memanfaatkan pendekatan transfer learning pada backbone EfficientNet-B0. Eksperiment membandingkan empat algoritma optimisasi (SGD, AdaGrad, RMSProp, dan Adam) untuk menentukan strategi pelatihan yang menyeimbangkan kecepatan konvergensi dan kemampuan generalisasi. Kinerja model diukur menggunakan anlisis confusion matrix dan metrik klasifikasi standar (akurasi, presisi, recall, dan F1-score). Konfigurasi terbaik, yakni EfficientNet-B0 yang di fine tuned dengan optimizer Adam, mencapai akurasi pelatihan 92%, akurasi validasi 89%, dan akurasi 88% pada partisi uji terpisah. Model yang terlatih diekspor dan diintegrasikan ke dalam aplikasi web sederhana berbasis Flask dengan penyimpanan SQLite, kemudian diuji menggunakan skema blackbox untuk menilai keandalan fungsional. Seluruh fungsi inti memenuhi spesifikasi dan prediksi runtime menunjukkan kesesuaian yang tinggi dengan label sebenarnya. Sistem ini menawarkan alat cepat dan praktis bagi pemilik serta penggemar kucing untuk mengidentifikasi ras di lingkungan nyata, sekaligus menyoroti manfaat transfer learning pada implementasi web yang ringkas.

Kata kunci: Klasifikasi ras kucing; EfficientNet-B0; transfer learning; aplikasi web Flask

INTRODUCTION

Cats rank among the world's most beloved companion animals, including in Indonesia, where 47% of survey respondents reported keeping cats as pets (Rakuten Insight, 2021). This popularity stems from cats' adaptability to diverse living environments, their relatively low maintenance requirements, and the intense emotional bonds they form with their owners. As domestic cats encompass a variety of breeds, each characterized by distinct coat patterns, body shapes, and behavioral traits, accurate breed identification poses a challenge for many pet owners.

Several studies have underscored the importance of breed-specific knowledge for health

and welfare. For instance, (Salonen, et al., 2019) reported that Persian and Maine Coon cats exhibit elevated risk factors for polycystic kidney disease, whereas (Vapalahti et al., 2016) demonstrated significant behavioral differences, such as aggression levels and activity patterns across breeds. Inaccurate breed recognition may therefore lead to suboptimal nutritional planning and delayed detection of hereditary conditions. Leading pet food manufacturers, such as Royal Canin, have responded by formulating breed-tailored diets, further emphasizing the practical necessity of precise breed determination (Royal Canin, 2024). However, current manual identification, relying on visual inspection by pet owners or non specialist veterinarian sis inherently subjective, time consuming, and prone to error. To overcome these limitations and ensure the accuracy demanded by breed specific care, a more robust and objective approach is imperative.

To address this need for accurate, efficient, and objective identification, automated image based classification systems have gained traction, leveraging advances in Artificial Intelligence (AI) have enabled automated image classification pipelines based on convolutional architectures to deliver makedly improve results on a broad spectrum of computer vision problems, from medical diagnosis (e.g., detection of diabetic eye disease; (Albelaihi & Ibrahim, 2024) to animal breed recognition for robust object recognition. Surveys of the field highlight a wide range of CNN families (e.g., VGG, ResNet, Inception, MobileNet, EfficientNet) and summarize common challenges, such as data scarcity, class imbalance, and overfitting, as well as their remedies, including transfer learning and data augmentation (Alzubaidi et al., 2021; Zhao et al., 2024). Transfer learning allows CNN models pretrained on large datasets to be adapted to smaller, domain specific image collections, improving accuracy while lowering computational demans (Janiesch, Zschech, & Heinrich, 2021). EfficientNet has emerged as a leading transfer learning backbone because its compound scaling method jointly balances network, depth, width, and resolution for improved accuracy per compute cost (Tan & Le, 2019). Empirical comparisons indicate that EfficientNet variants frequently achieve competitive accuracy while maintaining favorable compute efficiency relaive to contemporary architectures such as ViT and gMLP, supporting the selection of EfficientNet-B0 for resource sensitive image classification tasks (Al-Rahhal et al., 2022). Also the base variant, EfficientNet-B0, has demonstrated high accuracy in animal image classification challenges (Reddy Aleti & Kurakula, 2024).

Building on the Oxford-IIIT Pet Dataset, which includes twelve common cat breeds (Parkhi et al., 2012), this study develops and evaluates a web-based cat breed classifier using EfficientNet-B0. We compare the performance of four optimizers, SGD, RMSprop, AdaGrad, and Adam by measuring accuracy, precision, recall, and F1-score. The trained model is intregated into a Flask web application with an SQLite backend and subjected to blackbox testing to verify functional reliability. By enabling rapid and accurate breed identification, our system aims to empower cat owners and enthusiasts to deliver breed appropriate care and to demonstrate the viability of lightwight, transfer learning based deployments in real world settings.

RESEARCH METHODS

This study outlines a systematic research workflow for developing a cat breed identification model using transfer learning on the EffcientNet architecture, with a comparative analysis of four optimizers. The process begins with problem identification and dataset collection, then progresses through data preparation, model development, including dedicated optimizer experiments followed by performance evaluation. The optimal model is then deployed and subjected to system testing. Figure 1 illustrates the complete research workflow.

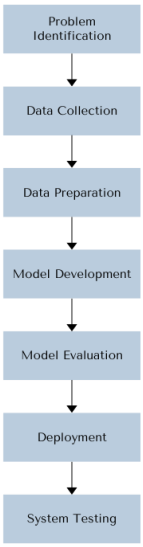


Figure 1. Research Workflow

Problem Identification

Problem identification in this study involves analyzing the challenge of accurately detecting cat breeds using transfer learning on the

EfficientNet architecture. The goal is to implement and evaluate this approach within a web based system to streamline breed recognition for end users.

Data Collection

This study utilizes the Oxford-IIIT Pet Dataset (Parkhi et al., 2012) as its primary image source, owing to its extensive use in pet classification research and its balanced distribution across categories. From the full dataset of 37 animal classes, the twelve most common cat breeds were selected, yielding 2400 colour images (200 per breed). All images were downloaded in their original JPG format and subjected to an integrity check (file size and format verification) before being organized into directories by breed label. To confirm label accuracy, breed annotations were cross referenced with a reputable Kaggle mirror of the same dataset.

Data Preparation

Prior to training, the image corpus was prepared via a four stage preprocessing pipeline. A deliberate 80:10:10 split (training: validation:test) was employed to ensure that all twelve feline categories were adequately and evenly represented in each partition. Second, every image was resized to 224 x 224 pixels, ensuring uniform input dimensions for the EfficientNet-B0 architecture. Third, pixel intensities were normalized by rescaling values to the [0, 1] range, which promotes faster convergence and stable gradient updates.

	class	train	val	test
0	Abyssinian	120	40	40
1	Bengal	120	40	40
2	Birman	120	40	40
3	Bombay	120	40	40
4	BritishShorthair	120	40	40
5	EgyptianMau	120	40	40
6	MaineCoon	120	40	40
7	Persian	120	40	40
8	Ragdoll	120	40	40
9	RussianBlue	120	40	40
10	Siamese	120	40	40
11	Sphynx	120	40	40

Figure 2. Dataset composition

Finally, on the fly data augmentation was applied during training to broaden sample variability and help prevent overfitting , transformations included random horizontal and vertical flips, rotations, zooms, crops, and shear. Such augmentation techniques are widely recommended to improve model robustness for image based tasks. To demonstrate that the applied transformations maintain the semantic characteristics of the cat breeds and are relevant to the classification objective, a visual example of the augmentation pipeline is presented in Figure 3. These operations and their taxonomy are widely discussed in the literatur eon image augmentation (Xu et al., 2023; Yang et al., 2023). In particular, recent surveys emphasize that characteristic and the downstream classification objective, and that generative augmentation methods can be considered when simple transforms do not provide sufficient diversity (Teerath Kumar et al., 2024).

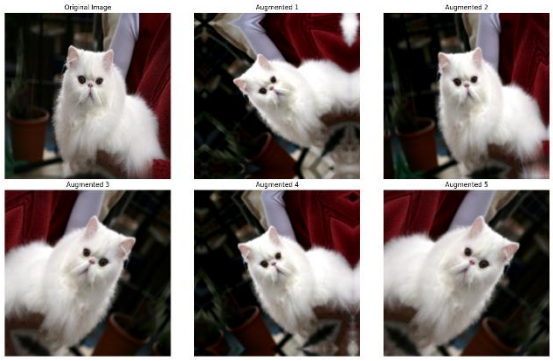


Figure 3. Data Augmentation Visualization

Model development

Model development was carried out by fine tuning the EfficientNet-B0 convolutional neural network using a transfer learning approach, whereby pretrained ImageNet weights provide initial feature representations that are adapted to the cat breed classification task. The choice of EfficientNet-B0 is informed by comparative analyses of CNN families that emphasize the importance of balancing accuracy and computational efficiency for transfer learning tasks (Bhatt et al., 2021; Tan & Le, 2019). By reusing general purpose visual representation acquired from large scale sources, transfer learning reduces the number of epochs required for convergence and typically improves performance on limited target dataset (Gupta et al., 2022).

For implementation, the pre trained EfficientNet-B0 backbone was intially frozen to utilize its feature extraction capabilities. As depicted in Figure 4, a custom classification head was then appended to the base model. This head consisted of a Global Average Pooling 2D layer, a Dropout layer (set to 0.5) for regularization, and a Dense layer with 12 output notes (corresponding to the target classes) using the *softmax* activation funciton. This configuration allowed the model to rapidly learn the classification task based on the high level features extracted by the frozen backbone. The overal structure is defined by the *build_model* function in the implementation.

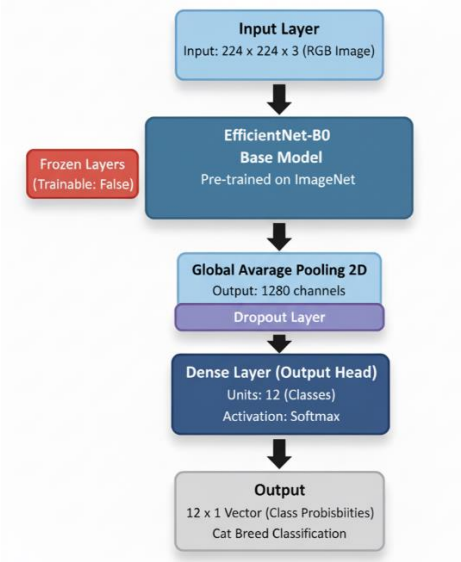


Figure 4. Proposed EfficientNet-B0 Model Architecture

Optimizer selection and configuration played a critical role in training dynamics. Empirical comparisons in computer vision context further demonstrate measureable performance differences among commonly used methods, motivating an experimental evaluation of candidate optimazers for this task (Bashetty et al., 2022; Hassan et al., 2023). Four gradient based optimizers, Stochastic Gradient Descent (SGD), AdaGrad, RMSProp, and Adam were therefore evaluated under consistent hyperparameter regimes to isolate their effects on learning dynamics and final accuracy.

All models were trained for up to 100 epochs using the *categorical crossentropy* loss function. To ensure robustness and prevent overfitting, the following practice measures were strictly applied. First, fixed *random seeds* used to promote full reproducibility. *Model checkpointing* applied to save the model weights that achieved the best validation loss throughout the training process, and *Early Stopping* to the validation loss with patience of 5 epochs. This mechanism automatically terminates training if no improvement in validation los sis observed for five consecutive epochs, and the best weights are restored. The relative performance of these optimizers, as measured by training and validation accuracy curves and time to convergence, will be illustrated alongside a comparison of other popular CNN transfer learning architectures.

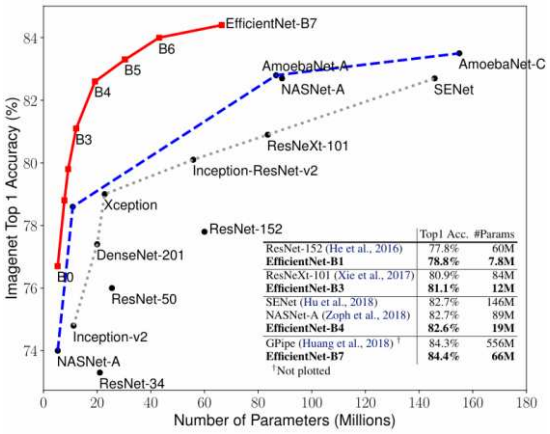


Figure 5. Comparison of CNN Transfer Learning Architectures

Model Evaluation

Model evaluation was performed using multiclass confusion matrix analysis to quantify the classifier's ability to distinguish among the twelve cat breeds. The confusion matrix evaluates classifier performance by comparing predicted versus actual labels and organizing results into four outcome types, true positive (TP), true negative (TN), false positive (FP), and false negative (FN), which serve as the basis for further metrics (Kulkarni et al., 2020). From these counts, four primary metrics were computed to assess overall and per-class performance.

Overall accuracy is calculated as the rasio of correctly predicted examples to the total examples across all classes, formally given by

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}.$$

However, accuracy is prone to misinterpretation when class are imbalanced, consequently, precision is provided as a complementary metric, defined as the ratio of true positive predictions to all positive predictions made by the model,

$$Precision = \frac{TP}{TP+FP},$$

and is critical when false positives carry significant cost. Recall (sensitivity) quantifies the rasio of true positives to the total number of actual positive examples, reflecting the model's capacity to capture real positive cases,

$$Recall = \frac{TP}{TP+FN},$$

and is essential when a positive instance is missing, as it can be costly. Defined as the harmonic mean of

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i1.417
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

precision and recall, the F1-score furnishes a unified performance indikator that accounts for both error types.

$$F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall}.$$

These metrics were generated for each breed and averaged (macro and weighted averages) to produce a comprehensive classification report.

The model evaluation stage will include a full classification report detailing precision, recall, and F1-score for each class. Additionally, the complete confusion matrix for the test set will also be visualized in this stage of research, to illustrate specific patterns of misclassification and guide future model refinements.

Deployment

Following model evaluation, the best performing EfficientNet-B0 classifier was exported as a serialized weight file and integrated into the application environment. The deployment stage involved loading the trained model into a lightweight interface pipeline, implemented via a Flask REST API, which accepts image inputs and returns breed predictions. To align with current MLOps guadiance, the model and its artifacts were loaded once at application startup, and lightweight practices for artifact/version management and runtime monitoring were adopted to minimise latency and increase operational reliability (Bayram & Ahmed, 2025; Mboweni, Masombuka, & Dongmo, 2022). This minimal deployment setup ensures rapid inference while maintaining compatibility with the upstream web interface.

System Testing

To assess real world performance beyond the held out test set, system testing was conducted using previously unseen images collected from end users and domain experts. These out of sample examples, separate from the training, validation, and test splits, were sent to the deployed API under controlled conditions for evaluation . Functional correctness, response time, and prediction accuracy on these real life data were measured to validate system robustness and generalizability.

RESULTS AND DISCUSSION

In this section, the training dynamics and validation behavior of EfficientNet-B0 models, fine-tuned with four different optimizers, are presented and analyzed. First, the evolution of accuracy and loss over epochs is visualized to illustrate convergence characteristics. Next, a quantitative comparison of peak training and validation metrics is provided. Together, these results demonstrate how the choice of optimizer impacts both learning speed and the ultimate performance of the model.

Training and Validation Performance

The accuracy and loss curves for each optimizer are shown in Figure 4. These plots reveal that AdaGrad achieves the fastest initial rise in training accuracy but begins to plateau earlier. In contrast. In contrast, SGD converges more slowly but attains a higher peak validation accuracy before overfitting. RMSProp and Adam exhibit intermediate behaviors, with Adam stricking the best balance between convergence speed and stability.

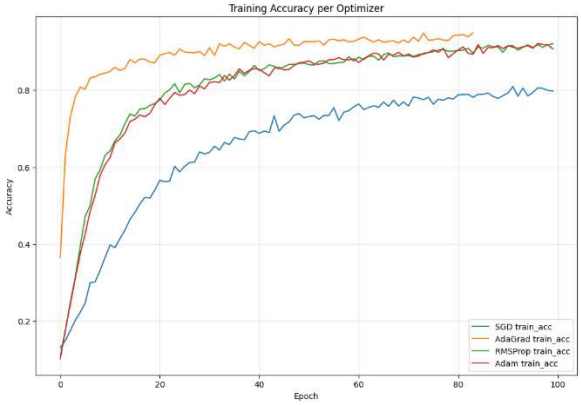


Figure 6. (a) Training Accuracy per Optimizer

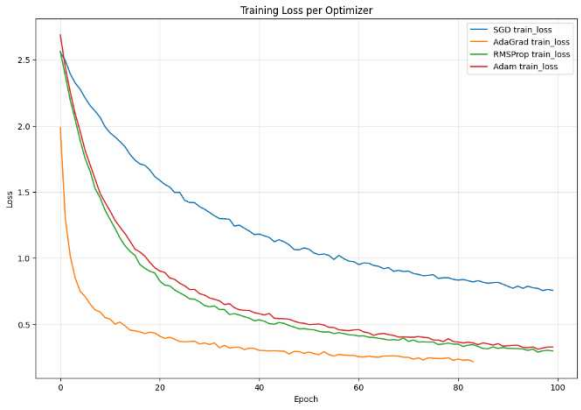


Figure 6. (b) Training Loss per Optimizer

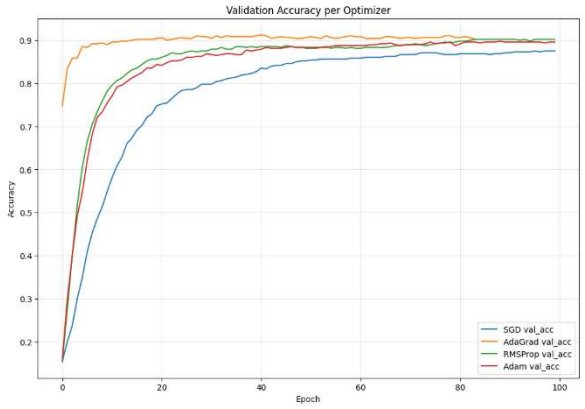


Figure 6. (c) Validation Accuracy per Optimizer

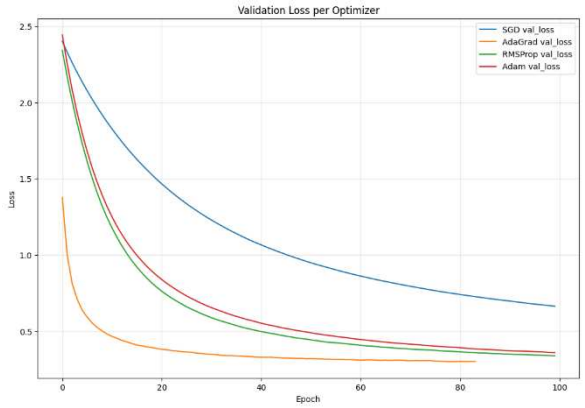


Figure 6. (d) Validation Loss per Optimize

Immediately following these visualizations, a summary of peak performance per optimizer is provided in Table 1, recorded items comprise training/validation accuracies, their respective loss measures,, and the epoch index corresponding to early stopping activation.

Table 1. Comparison of Training and Validation Performance by Optimizer

Optimizer	Best Train Acc	Best Val Acc	Train Loss	Val Loss	Epoch stop
SGD	80%	88%	0.78	0.66	100
AdaGrad	94%	91%	0.22	0.30	95
RMSProp	91%	90%	0.30	0.34	100
Adam	92%	89%	0.32	0.36	100

The results indicate that although AdaGrad reaches the highest training accuracy, its validation peak is closely matched by RMSProp and closely followed by Adam. SGD, while slower to converge, achieves a respectable validation accuracy but requires the full 100 epochs.

To quantitatively assess these models, a more granular analysis of their convergence patterns and generalization gaps is required:

1. Stabilization trend and fluctuation, as visualized in figure 6 (d), all the three adaptive optimizers (AdaGrad, RMSProp, and Adam) demonstrate highly stable convergence. Their validation loss curves (orange, green, red) are smooth, with minimal inter epoch fluctuation, and stabilize relatively early in the training process. In contrast, SGD (blue line) converges significantly slower and stabilizes at much higher validation loss (0.66).
2. Generalization gap, the critical differentiator is the generalization gap, calculated from Table 1. RMSProp achieves the smallest gap at just 1% (91% - 90%), indicating superior generalization on the validation data. AdaGrad and Adam follow with a similar, larger gap of 3%. SGD's negative gap confirms its tendency to underfit the data

This analysis confirms that while Adam provides the stable convergence it is known for, the validation data reveals a highly competitive trade off. AdaGrad achieves the highest peak accuracy (91%), while RMSProp demonstrates the strongest generalization with 1% gap. Given these nuanced results on the validation set, a definitive conclusion on the best optimizer is premature. The final, most reliable measure of performance will be determined by evaluating these models on the held out test set, which is presented in the following section.

Test Set Evaluation

Generalization capability was assessed with the held out test set, macro avaraged metrics (accuracy, precision, recall, F1-score) were computed for each optimizer, and confusion matrix for the leading model was produced to visualise misclassification trends. The results, summarised in Figure 5, reveal nuanced trade offs among the four optimizers algorithms when applied to EfficientNet-B0.

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i1.417
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

```
===== SGD — Classification Report =====
                  precision    recall  f1-score   support

      Abyssinian       0.86      0.93      0.89        40
          Bengal       0.88      0.75      0.81        40
          Birman       0.69      0.82      0.75        40
          Bombay       0.97      0.97      0.97        40
British Shorthair      0.64      0.85      0.73        40
     EgyptianMau       0.86      0.90      0.88        40
       MaineCoon       0.95      0.93      0.94        40
         Persian       0.92      0.90      0.91        40
         Ragdoll       0.73      0.75      0.74        40
     RussianBlue       0.81      0.53      0.64        40
         Siamese       0.97      0.88      0.92        40
          Sphynx       1.00      0.97      0.99        40

        accuracy                           0.85       480
       macro avg       0.86      0.85      0.85       480
    weighted avg       0.86      0.85      0.85       480
```

Figure 5. (a) SGD Classification Report on Test Set

```
===== AdaGrad — Classification Report =====
                  precision    recall  f1-score   support

      Abyssinian       0.90      0.93      0.91        40
          Bengal       0.89      0.78      0.83        40
          Birman       0.74      0.85      0.79        40
          Bombay       0.97      0.97      0.97        40
British Shorthair      0.72      0.85      0.78        40
     EgyptianMau       0.86      0.93      0.89        40
       MaineCoon       0.95      0.93      0.94        40
         Persian       0.95      0.90      0.92        40
         Ragdoll       0.79      0.82      0.80        40
     RussianBlue       0.82      0.68      0.74        40
         Siamese       0.95      0.88      0.91        40
          Sphynx       1.00      0.97      0.99        40

        accuracy                           0.87       480
       macro avg       0.88      0.87      0.87       480
    weighted avg       0.88      0.87      0.87       480
```

Figure 5. (b) AdaGrad Classification Report on Test Set

```
===== RMSProp — Classification Report =====
                  precision    recall  f1-score   support

      Abyssinian       0.93      0.93      0.93        40
          Bengal       0.88      0.75      0.81        40
          Birman       0.77      0.82      0.80        40
          Bombay       0.97      0.97      0.97        40
British Shorthair      0.69      0.88      0.77        40
     EgyptianMau       0.84      0.93      0.88        40
       MaineCoon       0.97      0.93      0.95        40
         Persian       0.92      0.90      0.91        40
         Ragdoll       0.76      0.88      0.81        40
     RussianBlue       0.86      0.60      0.71        40
         Siamese       0.95      0.88      0.91        40
          Sphynx       1.00      1.00      1.00        40

        accuracy                           0.87       480
       macro avg       0.88      0.87      0.87       480
    weighted avg       0.88      0.87      0.87       480
```

Figure 5. (c) RMSProp Classification Report on Test Set

```
===== Adam — Classification Report =====
                  precision    recall  f1-score   support

      Abyssinian       0.93      0.95      0.94        40
          Bengal       0.91      0.80      0.85        40
          Birman       0.77      0.85      0.81        40
          Bombay       0.97      0.97      0.97        40
British Shorthair      0.72      0.85      0.78        40
     EgyptianMau       0.86      0.95      0.90        40
       MaineCoon       0.92      0.90      0.91        40
         Persian       0.90      0.90      0.90        40
         Ragdoll       0.80      0.88      0.83        40
     RussianBlue       0.83      0.62      0.71        40
         Siamese       0.97      0.88      0.92        40
          Sphynx       1.00      1.00      1.00        40

        accuracy                           0.88       480
       macro avg       0.88      0.88      0.88       480
    weighted avg       0.88      0.88      0.88       480
```

Figure 5. (d) Adam Classification Report on Test Set

Adam achieves the highest overall accuracy (88%), with uniformly strong precision and recall across breeds. AdaGrad and RMSProp achieve comparable accuracy, yet Adam holds a small lead in overall accuracy and attains the best F1 measure, which implies a more balanced decrease in both false positive and false negative errors.

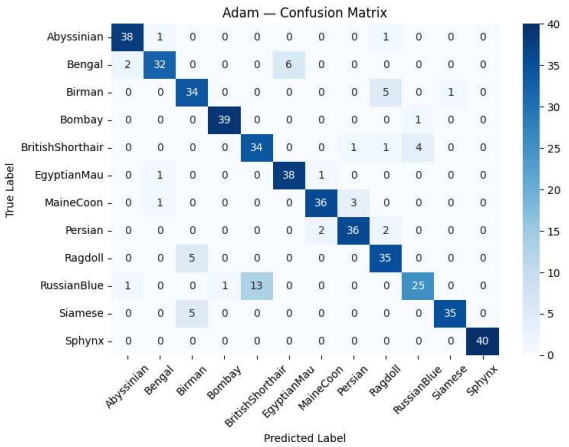


Figure 6. Confusion Matrix for Adam Optimizer on Test Set

Figure 6 illustrates the confusion matrix for the Adam based model. Most breeds such as Bombay and Sphynx exhibit near perfect true positive rates. At the same time, occasional misclassifications occur between visually similar categories (e.g., Birman vs. Ragdoll vs. Siamise, Russian Blue vs. other blue-toned coats). Adam's adaptive moment estimation stabilises gradient updates throughtout training, yielding robust per-class performance and minimising misclassification clusters.

These findings suggest that Adam's combination of momentum and adaptive learning rates confers superior stability during fine-tuning, enabling EfficientNet-B0 to generalise effectively on multiclass image data. For practical deployment, the Adam tuned model offers the best trade off between high accuracy and consistent performance across all twelve cat breeds.

System Testing

The final evaluation stage assesses the deployed EfficientNet-B0 model in a production like environment. After export and integration into a Flask based REST API, two key aspects were tested, functional correctness via black-box testing and real world accuracy using external images provided by cat owners and experts. Additionally, core user interfaces (homepage and history page) are shown to demonstrate how predictions and results are presented.

Functional verification was performed through black-box testing, where each feature, including uploading image, interface, authentication, history retrival, and deletion was exercised without inspecting the internal code. Table 2 summarises the test cases and their pass or fail outcomes.

Table 2. Black-Box Testing Result

Test Case	Input	Expected Output	Actual Result
Account Registration	Valid username, email, password	Redirect to sign-in	Pass
User login	Valid credentials	Access homepage	Pass
Image upload	JPEG/PNG < 5MB	Preview + "Detect active"	Pass
Breed detection	Uploaded cat image	Prediction + confidence	Pass
View history	-	Table of past detection	Pass
Delete record	History ID	Record removed	Pass
Oversize upload	>5MB image	Validation error	Pass
Logout	-	Redirect to sign-in	Pass

To contextualize how users interact with the application, Figure 7 and 8 presents the homepage, where images are uploaded and predictions are displayed, and the history page, which organises past results by user id and timestamp.

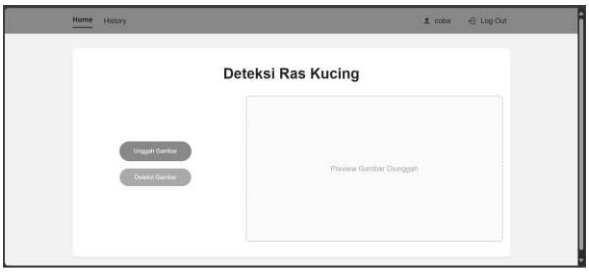


Figure 7. User Interface Homepage

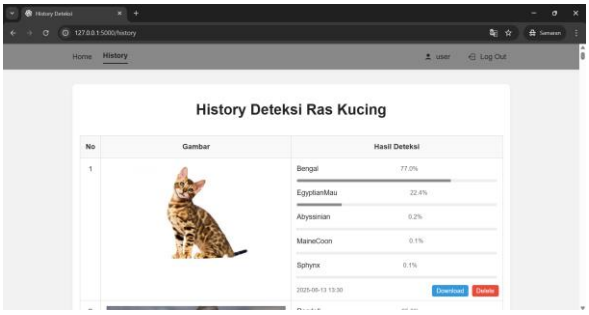
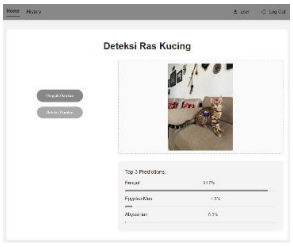


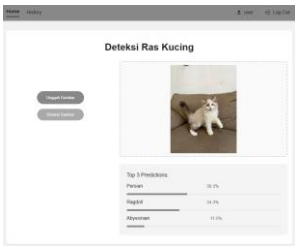
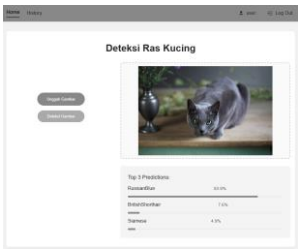
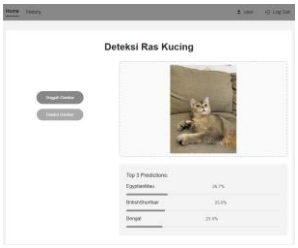
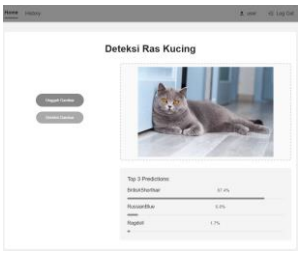
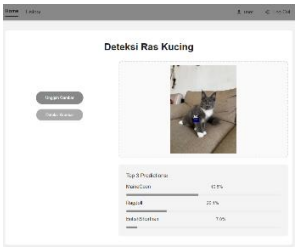
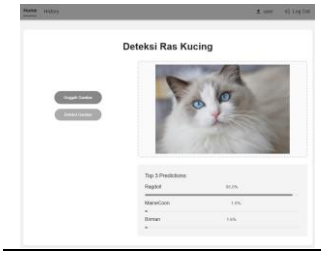
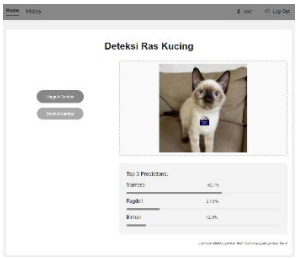
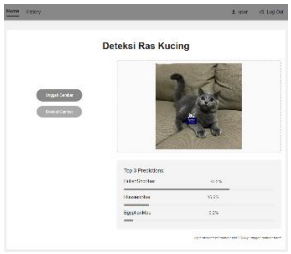
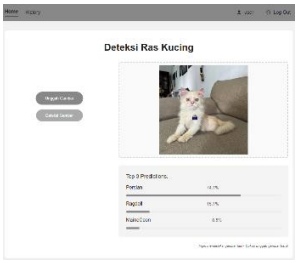
Figure 8. User Interface History Page

Finally, the model's real world accuracy was measured on ten external images of purebred cats not included in the original dataset. Table 3 reports the true breed, predicted breed, and correctness for each sample. The system correctly identified eight out of ten images (80%), confirming that the EfficientNet-B0 + Adam configuration generalizes well to new, real life inputs.

Table 3. Real-World Accuracy on External Images

Images	True Breed	Correct (Y/N)
	Bengal	Y

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i1.417
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

	Ragdoll	N		Russian Blue	Y
	British Shorthair	N		British Shorthair	Y
	Maine Coon	Y		Ragdoll	Y
	Siamise	Y			
	British Shorthair	Y			
	Persian	Y			

The results indicate that the deployed model achieved both functional requirements but also maintains high classification accuracy in real-world scenarios, validating its practical applicability.

CONCLUSIONS AND SUGGESTIONS

Conclusion

This study developed and evaluated a transfer learning approach for cat breed identification using EfficientNet-B0 and a comparative examination of four optimizers. Results show that the choice of optimizer materially influenced training dynamics and generalization. AdaGrad achieved the highest training accuracy but exhibited earlier plateauing and signs of overfitting. At the same time, Adam provided the best balance of convergence, stability, and test set performance, achieving 88% accuracy on the held out set. Confusion matrix analysis highlighted strong recognition for breeds such as Bombay and Sphynx and recurring confusion among visually similar categories. End to end verification demonstrated that the model can be integrated into a lightweight Flask service with reliable functionality. A small real world trial (10 external images) produced 80%

correct identifications, supporting the system's practical applicability for in distribution images.

Suggestion

For future work and practical deployment, it is recommended to enlarge and diversify the evaluation corpus to better capture real world variablitiy (lighting, pose, background, and mixed breeds) and to investigate more substantial augmentation or generative data augmentation methods to reduce class confussion. Additionally, broader comparison accross backbones (larger EfficientNet variants, lightweight CNNs or transformer hybrids) and more systematic hyperparameter/optimizer searches (including recent optimizers and regularization strategies) could clarify trade offs between accuracy, latency and model size. Operationally, adopting basic MLOps practices such as single time model loading, model versioning, monitoring, and automated testing and conducting user studies with cat owners or veterinarians would further validate the utility and guide improvements for real world adoption.

REFERENCES

Abdulkadirov, R., Lyakhov, P., & Nagornov, N. (2023). Survey of Optimization Algorithms in Modern Neural Networks. *Mathematics*, *11*(11), 1–37. https://doi.org/10.3390/math11112466

Albelaihi, A., & Ibrahim, D. M. (2024). DeepDiabetic: An Identification System of Diabetic Eye Diseases Using Deep Neural Networks. *IEEE Access*, *12*(November 2023), 10769–10789. https://doi.org/10.1109/ACCESS.2024.3354854

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., … Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. In *Journal of Big Data* (Vol. 8). Springer International Publishing. https://doi.org/10.1186/s40537-021-00444-8

Bashetty, S., Raja, K., Adepu, S., & Jain, A. (2022). Optimizers in Deep Learning: A Comparative Study and Analysis. *International Journal for Research in Applied Science and Engineering Technology*, *10*(12), 1032–1039. https://doi.org/10.22214/ijraset.2022.48050

Bayram, F., & Ahmed, B. S. (2025). Towards Trustworthy Machine Learning in Production: An Overview of the Robustness in MLOps Approach. *ACM Computing Surveys*, *57*(5). https://doi.org/10.1145/3708497

Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., … Ghayvat, H. (2021). Cnn variants for computer vision: History, architecture, application, challenges and future scope. *Electronics (Switzerland)*, *10*(20), 1–28. https://doi.org/10.3390/electronics10202470

Dogo, E. M., Afolabi, O. J., & Twala, B. (2022). On the Relative Impact of Optimizers on Convolutional Neural Networks with Varying Depth and Width for Image Classification. *Applied Sciences (Switzerland)*, *12*(23). https://doi.org/10.3390/app122311976

Gupta, J., Pathak, S., & Kumar, G. (2022). Deep Learning (CNN) and Transfer Learning: A Review. *Journal of Physics: Conference Series*, *2273*(1). https://doi.org/10.1088/1742-6596/2273/1/012029

Hassan, E., Shams, M. Y., Hikal, N. A., & Elmougy, S. (2023). The effect of choosing optimizer algorithms to improve computer vision tasks: a comparative study. In *Multimedia Tools and Applications* (Vol. 82). Multimedia Tools and Applications. https://doi.org/10.1007/s11042-022-13820-0

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Springer*. https://doi.org/10.1007/s12525-021-00475-2/Published

Kulkarni, A., Chong, D., & Batarseh, F. A. (2020). Foundations of data imbalance and solutions for a data democracy. In *Data Democracy: At the Nexus of Artificial Intelligence, Software Development, and Knowledge Engineering*. Elsevier Inc. https://doi.org/10.1016/B978-0-12-818366-3.00005-8

Kumar, T., Brennan, R., Mileo, A., & Bendechache, M. (2024). Image Data Augmentation Approaches: A Comprehensive Survey and Future Directions. *IEEE Access*, *12*, 187536–187571. https://doi.org/10.1109/ACCESS.2024.3470122

Mboweni, T., Masombuka, T., & Dongmo, C. (2022). A Systematic Review of Machine Learning DevOps. *2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, 1–6. https://doi.org/10.1109/ICECET55527.2022.9872968

Pang, Bo, Nijkamp, Erik, & Wu, Ying Nian. (2019). Deep Learning With TensorFlow: A Review. *Journal of Educational and Behavioral Statistics*, *45*(2), 227–248.

https://doi.org/10.3102/107699861987276 1

Parkhi, O. M., Vedaldi, A., Zisserman, A., & Jawahar, C. V. (2012). *Cats and Dogs*. Retrieved from robots.ox.ac.uk

Rahhal, M. M. Al, Bazi, Y., Jomaa, R. M., Zuair, M., & Melgani, F. (2022). Contrasting EfficientNet, ViT, and gMLP for COVID-19 Detection in Ultrasound Imagery. *Journal of Personalized Medicine*, *12*(10). https://doi.org/10.3390/jpm12101707

Rakuten Insight. (2021, February 27). Pet Ownership in Asia. Retrieved from insight.rakuten.com website: https://insight.rakuten.com/pet-ownership-in-asia/

Reddy Aleti, S., & Kurakula, K. (2024). *Evaluation of Lightweight CNN Architectures for Multi-Species Animal Image Classification*. Retrieved from www.bth.se

Royal Canin. (2024). Cat Nutrition Products. Retrieved December 17, 2024, from royalcanin.com website: https://www.royalcanin.com/id/cats/products/retail-products

Salonen, M., Vapalahti, K., Tiira, K., Mäki-Tanila, A., & Lohi, H. (2019). Breed differences of heritable behaviour traits in cats. *Scientific Reports*, *9*(1), 1–10. https://doi.org/10.1038/s41598-019-44324-x

Tan, M., & Le, Q. V. (2019). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. California.

Vapalahti, K., Virtala, A. M., Joensuu, T. A., Tiira, K., Tähtinen, J., & Lohi, H. (2016). Health and behavioral survey of over 8000 finnish cats. *Frontiers in Veterinary Science*, *3*(AUG). https://doi.org/10.3389/fvets.2016.00070

Xu, M., Yoon, S., Fuentes, A., & Park, D. S. (2023). A Comprehensive Survey of Image Augmentation Techniques for Deep Learning. *Pattern Recognition*, *137*, 109347. https://doi.org/10.1016/j.patcog.2023.109347

Yang, Z., Sinnott, R. O., Bailey, J., & Ke, Q. (2023). A survey of automated data augmentation algorithms for deep learning-based image classification tasks. In *Knowledge and Information Systems* (Vol. 65). Springer London. https://doi.org/10.1007/s10115-023-01853-2

Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M., & Parmar, M. (2024). A review of convolutional neural networks in computer vision. In *Artificial Intelligence Review* (Vol. 57). Springer Netherlands. https://doi.org/10.1007/s10462-024-10721-6