Cat Skin Disease Diagnosis Using EfficientNetV2 for Lightweight Processing on Low-Resource Devices

Fadila Rizka Nur Aminah, Mirza Alim Mutasodirin*, Muhammad Fikri Hidayattullah

Faculty of Science and Technology, Universitas Harkat Negeri
Pesurungan Lor, Margadana, Tegal City, Central Java, 52147, Indonesia
Corresponding author. Email: mirza.alim.m@harkatnegeri.ac.id

Abstract— Skin diseases are among the most common health issues in domestic cats. However, access to veterinarians is often limited, especially in low-resource settings. Automated image-based detection offers a fast and affordable alternative for early intervention. This paper presents a lightweight approach for diagnosing feline skin diseases using EfficientNetV2 optimized for low-resource devices. A balanced custom dataset consisting of 720 images across nine classes, namely Healthy, Mild/Severe Ringworm, Mild/Severe Acne, Mild/Severe Flea, and Mild/Severe Scabies, was compiled from Kaggle, Roboflow, and Google Images, ensuring ethical use of publicly available data. The images were augmented through rotations (0°, 90°, 180°, 270°) and horizontal flips, resulting in 5,760 images, to enhance model generalization. Five CNN architectures were benchmarked: DenseNet121, MobileNetV2, MobileNetV3, EfficientNetB0, and EfficientNetV2B0. Training was conducted with grid searches over batch sizes {64, 32, 16, 8} and learning rates {1e-3, 5e-4, 2e-4, 1e-4, 5e-5} for up to 300 epochs, and with the Adam optimizer and Reduce-LR-on-Plateau (decay factor 0.5). Early stopping (patience = 10) was used to mitigate overfitting. The best model was selected based on highest validation accuracy. The experiments were conducted on an Intel Xeon 6 CPU (2.2 GHz, 2 vCPUs) in Google Colab without GPU to simulate low-resource deployment. EfficientNetV2B0 achieved the best performance with 99.62% validation accuracy and 99.79% test accuracy, with an average inference latency of 78 ms/frame. Compared to previous studies focusing on heavyweight models or conventional ML using handcrafted features, this work highlights the feasibility of deploying an accurate real-time diagnostic pipeline on edge devices.

Keywords— cat skin; disease diagnosis; EfficientNetV2; feline skin; low-resource device

I. INTRODUCTION

Feline dermatological disorders are among the most frequently reported medical problems in domestic cats. Studies have shown that dermatological cases account for approximately 13-15% of all feline veterinary consultations [1], [2]. Parasites and bacterial infections are the most common causes for up to 55% of infectious skin diseases in cats [2]. These conditions not only cause discomfort and secondary infections but can also pose zoonotic risks to humans [3], underscoring the need for early and accurate diagnosis.

However, dermatological assessment in cats typically requires specialized equipment and expert visual inspection, which may not be available in low-resource environments or for individual pet owners. This diagnostic gap highlights the need for automated, accessible, and computationally efficient image-based systems that can assist in identifying feline skin diseases in real-time. Early and accurate diagnosis of these skin ailments is crucial to prevent progression, alleviate discomfort, and avoid secondary infections. Recent advances in Machine Learning and Deep Learning have enabled automated image-based diagnosis [4], offering rapid, non-invasive screening that can be deployed even in remote areas with minimal infrastructure.

The research landscape for automated feline skin disease diagnosis has undergone a significant evolution, establishing clear trends that justify the necessity of modern, lightweight Deep Learning architectures like EfficientNetV2 [5]. Early studies focused on establishing the feasibility of diagnosis using symptom-based expert systems, employing methods such as Certainty Factor [6], [7], Naïve Bayes [8], [9], and Support Vector Machine (SVM) [10]. While traditional classifiers like SVM achieved remarkably high numerical accuracy (up to 98.75%) based on textual symptom input, this approach was constrained by the need for expert knowledge encoding and the inherent user error in symptom identification [11]. Furthermore, while methods like Fuzzy Sugeno provided crucial actionable intelligence (disease severity percentage) [12], they remained reliant on subjective symptom scoring.

The field pivoted with the introduction of image-based diagnosis using Deep Learning, recognizing that human diagnosis is primarily visual. Papers began implementing Convolutional Neural Networks (CNNs) for image classification and object detection algorithms like YOLOv8 for real-time localization of disease areas [13], [14]. This shift satisfied the demand for a more intuitive user interface and confirmed the necessity of mobile application deployment [13], [15].

However, this transition introduced a crucial trade-off. Deep learning models are computationally expensive [16]. While full-size CNNs are powerful, their complex calculations necessitate high computational power, making them challenging to deploy on resource-constrained mobile devices for real-time use [16]. This bottleneck was addressed in

Received 07 July 2025, Revised 10 November 2025, Accepted 11 November 2025.
DOI: https://doi.org/10.15294/jte.v17i2.29764

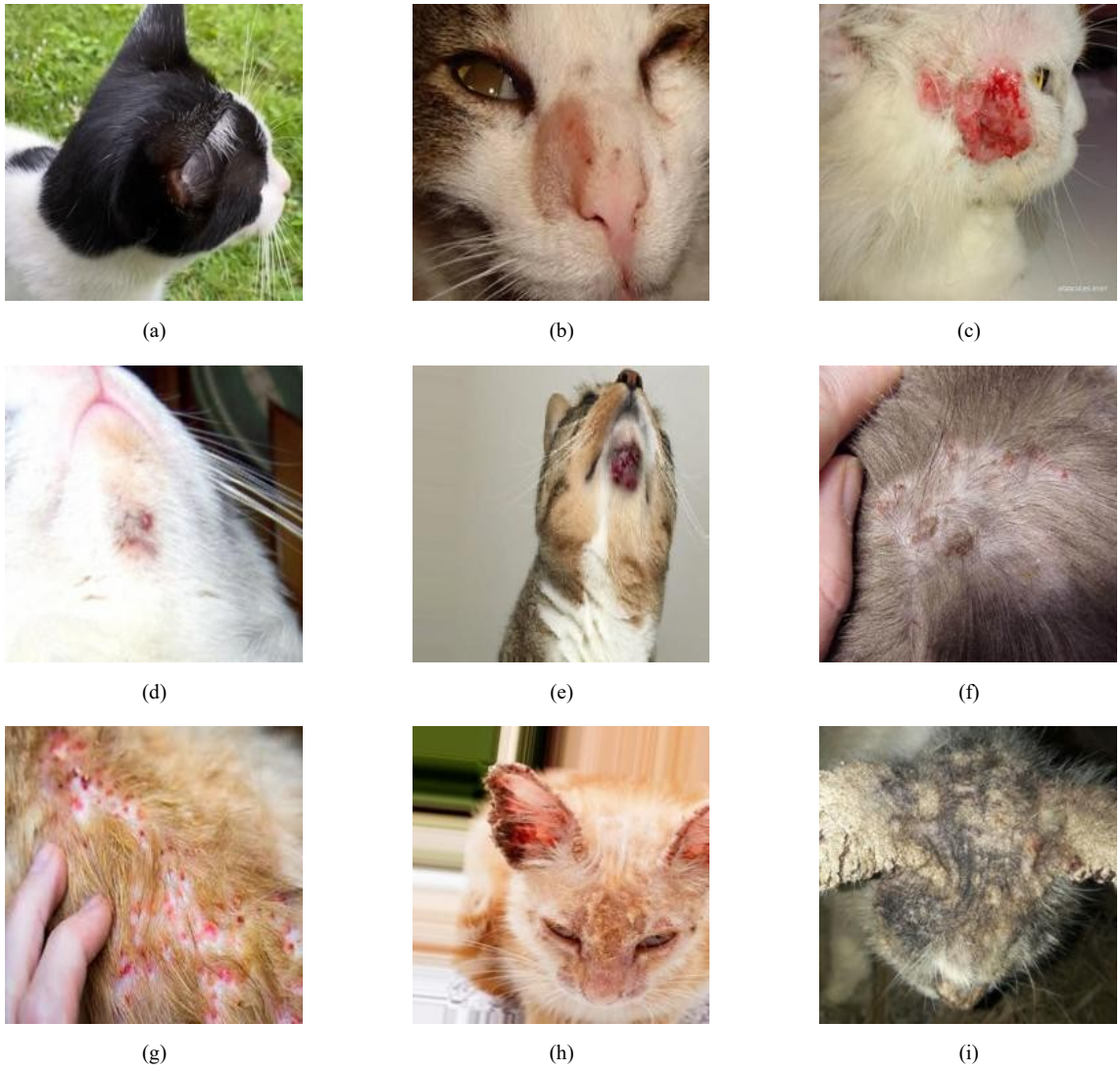

Figure 1. Example images from each diagnostic class (a) Healthy (b) Mild Ringworm (c) Severe Ringworm (d) Mild Acne (e) Severe Acne (f) Mild Flea (g) Severe Flea (h) Mild Scabies, and (i) Severe Scabies

subsequent work by exploring techniques like 8-bit quantization via TensorFlow Lite, which successfully reduced model size by 74.7% while maintaining high accuracy, thus ensuring practicality on smartphones [16]. Additionally, the use of hybrid models, such such as CNN combined with Random Forest, was explored to improve stability and generalization over standalone CNNs by leveraging robust traditional classifiers [17].

The selection of EfficientNetV2 is a strategic response to the combined findings of all sixteen papers, addressing the limitations of prior methods while embracing the best of the latest advancements. EfficientNetV2, as a Deep Learning architecture, overcomes the fundamental constraint of symptom-based models (SVM, Naïve Bayes, CF) by utilizing image data for diagnosis. It bypasses the need for the user to accurately select subjective symptoms, which was a persistent source of error in the traditional ML systems [11].

While earlier papers used basic CNN architectures [13], these are known to be computationally heavy and resource-intensive, requiring specialized optimization techniques like the 8-bit quantization to be deployable on mobile devices [16]. EfficientNetV2, by design, employs a highly effective compound scaling method and incorporates advanced features (like MBConv and Fused-MBConv) that make it lightweight and efficient out-of-the-box. This inherent efficiency is a direct advantage over the older, non-optimized CNN models.

EfficientNetV2 is designed for excellent performance on complex visual tasks. The ability to use a pre-trained model (transfer learning) allows the system to leverage features learned from massive datasets like ImageNet. It drastically reduces the training time and data requirements for the relatively smaller domain of cat skin diseases. This is a significant advantage over training a complex hybrid model (CNN + Random Forest) from scratch, which requires two separate stages of training and optimization [17].

While recent frameworks such as YOLO v8 [14] and CNN + RF hybrids [17] demonstrate strong accuracy, they demand substantial GPU resources and high-resolution datasets. Transformer-based lightweight models such as MobileViT [18] and Mobile-Former [19] deliver competitive accuracy, but often require complex token-mixing operations, larger parameter counts, and hardware acceleration (GPU/TPU) not universally available on edge devices. EfficientNetV2 was therefore selected in this study because it achieves an optimal accuracy-efficiency trade-off through progressive compound scaling and fused-MBConv layers.

Compared to previous CNNs and quantized networks [15], EfficientNetV2 offers smaller model size and faster convergence, enabling training on limited datasets, superior parameter efficiency, built-in regularization and gradient stability, and simpler pipeline than transformer-based models (MobileViT). Thus, EfficientNetV2 bridges the gap between

TABLE I. NUMBER OF IMAGES PER DIAGNOSTIC CLASS ALLOCATED TO TRAINING, VALIDATION, AND TESTING SETS AFTER AUGMENTATION

Split	Healthy	Ringworm		Acne		Flea		Scabies		Total	%
		Mild	Severe	Mild	Severe	Mild	Severe	Mild	Severe		
Train	480	480	480	480	480	480	480	480	480	**4320**	75.0
Val	80	80	80	80	80	80	80	80	80	**720**	12.5
Test	80	80	80	80	80	80	80	80	80	**720**	12.5
Total	**640**	**640**	**640**	**640**	**640**	**640**	**640**	**640**	**640**	**5760**	**100**

classical lightweight CNNs and modern hybrid vision models. It offers high precision with manageable computational cost, making it particularly suitable for edge-based veterinary diagnostic applications where power and memory are limited.

This study leveraged EfficientNetV2B0, a lightweight architecture designed for faster convergence and improved parameter efficiency, as the backbone for cat skin disease diagnosis on low-resource devices. By fine-tuning EfficientNetV2B0 alongside comparative baselines (DenseNet121, MobileNetV2, MobileNetV3, and EfficientNetB0) on an augmented dataset of nine skin condition classes, this study aims to demonstrate a balanced solution that achieves high diagnostic accuracy while maintaining sub-100 ms CPU inference times.

The remainder of this paper is organized as follows: Section II details the dataset preparation and model training methodology. Section III presents experimental results and discussion. Section IV concludes with insights.

II. METHOD

This section details the end-to-end approach employed for feline skin disease diagnosis, from image collection and preprocessing through model selection, training procedures, and performance evaluation. First, this section describes how the dataset was assembled and augmented to ensure robustness across varied lesion presentations. Next, the five CNN architectures benchmarked in this work are introduced, highlighting their design motivations for lightweight, low-latency inference. The fine-tuning strategy is then outlined, including optimizer settings, learning rate scheduling, and hyperparameter search. Finally, the metrics and benchmarking protocol used to assess each model's diagnostic accuracy and CPU inference speed are specified, reflecting real-world deployment constraints.

A. Dataset and Preprocessing

Feline skin images were collected from Kaggle, Roboflow, and Google Images using class-specific search queries covering nine diagnostic categories: Healthy, Severe Ringworm, Mild Ringworm, Severe Acne, Mild Acne, Severe Flea, Mild Flea, Severe Scabies, and Mild Scabies. An initial set of 80 raw images per class (720 in total) was manually curated to ensure diversity in lighting, pose, background, fur color, and lesion appearance. Although this sample size is modest, it provides a balanced foundation for model comparison while allowing computational feasibility for extensive hyperparameter exploration. To reduce overfitting risk inherent to limited datasets, a systematic augmentation pipeline was applied to expand image diversity and improve generalization.

Each raw image underwent four rotations (0°, 90°, 180°, 270°) and horizontal flipping for each rotated variant. It results in a total data size of 640 samples per class (5,760 images in total). More complex augmentation methods, such as brightness and contrast adjustment, random cropping, or scaling, were intentionally excluded to maintain a manageable dataset size that facilitates efficient hyperparameter tuning and cross-model benchmarking. The chosen geometric transformations provide sufficient variation in spatial orientation and texture without substantially inflating the computational load or training time. It makes the dataset suitable for systematic architecture comparison on limited hardware resources.

All images were resized to 224x224 pixels, the canonical input resolution for modern CNN architectures. This size was selected to maintain compatibility with pretrained ImageNet [20] weights while minimizing computational overhead for deployment on low-resource hardware. Image intensities were normalized to the [0, 1] range to match the input requirements of modern CNN backbones.

Dataset partitioning followed a stratified random split at the class level using a fixed random seed (seed = 42) to ensure reproducibility and preserve class balance across subsets. Each class was divided into 480 training, 80 validation, and 80 testing samples, corresponding to 75%, 12.5%, and 12.5% ratio respectively. The final data splits are summarized in Table I and representative samples from each class are illustrated in Figure 1. Table I provides the exact counts used for training, validation, and testing per class. Figure 1 demonstrates visual variability and highlights common challenges such as varying backgrounds.

B. Model Architectures

Five convolutional neural network (CNN) architectures were selected to evaluate the trade-off between accuracy and computational cost on edge hardware. CNN-based models remain a dominant choice for mobile-scale image classification due to their mature optimization kernels and consistently low inference latency on commodity CPUs.

1. DenseNet121, a densely connected architecture that concatenates feature maps from all preceding layers to encourage feature reuse and mitigate vanishing gradients [21]. Its parameter efficiency makes it a popular choice for medical image tasks.
2. MobileNetV2, employs inverted residual blocks and linear bottlenecks to reduce memory footprint and multiply-accumulate (MAC) operations, targeting mobile and embedded applications [22].
3. MobileNetV3, builds upon MobileNetV2 by incorporating network architecture search (NAS), optimized modules and lightweight attention ("squeeze-and-excite") blocks for further latency reduction [23].
4. EfficientNetB0, utilizes a compound scaling method to uniformly scale depth, width, and resolution, achieving state-of-the-art accuracy with fewer parameters [24].
5. EfficientNetV2B0, an evolution of EfficientNet that improves training speed via fused MBConv layers and optimized scaling, delivering faster convergence and lower inference latency [5].

TABLE II. CLASSIFICATION ACCURACY OF CNN ARCHITECTURES ON VALIDATION AND TEST DATASET

No.	Model	Val. Acc.	Test. Acc.	Avg. Acc.
1	EfficientNetV2B0	99.62%	**99.79%**	**99.71%**
2	EfficientNetB0	99.62%	99.78%	99.70%
3	DenseNet121	**99.81%**	99.44%	99.63%
4	MobileNetV2	99.43%	99.38%	99.40%
5	MobileNetV3	12.35%	8.79%	10.57%

TABLE III. TOP FOUR CNN ARCHITECTURES SIZE COMPARISON

No.	Model	Param.	Size	FLOPS	Latency
1	EfficientNetV2B0	5.9 M	22.64 MB	1.4 G	78 ms/frame
2	MobileNetV2	2.2 M	8.68 MB	0.6 G	80 ms/frame
3	EfficientNetB0	4.0 M	15.51 MB	0.8 G	140 ms/frame
4	DenseNet121	7.0 M	26.90 MB	5.7 G	190 ms/frame

Although Vision Transformers (e.g., MobileViT) and hybrid CNN-Transformer architectures have recently gained popularity, their self-attention mechanisms typically incur higher computational overhead and memory pressure on CPU-only environments. Existing mobile-oriented transformer variants often depend on optimized GPU or NPU acceleration to realize advertised efficiency gains, which are not consistently available on low-resource edge devices targeted in this work.

The selected architectures therefore align with the research gap identified earlier: designing and benchmarking classification models that remain accurate, computationally efficient, and practical for deployment on CPU-constrained edge hardware. By spanning a spectrum from highly compact (MobileNet variants) to moderately deep but still efficient (DenseNet, EfficientNet families), this evaluation enables a comprehensive comparison under identical training and inference constraints.

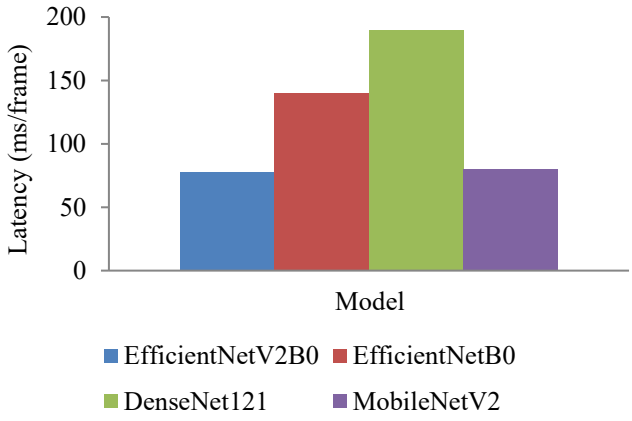

Figure 2. Bar chart showing average per-frame CPU inference latency (in milliseconds) for each model evaluated

C. Fine-Tuning Strategy

All models were initialized with ImageNet-pretrained weights and fine-tuned on the cat skin disease dataset. Data augmentation was performed offline (pregenerated) prior to training to ensure consistent image variations across experimental runs and to facilitate reproducibility. The Adam optimizer [25] was employed for its adaptive learning rate capabilities, with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 1e\text{-}7$, no weight decay, and exponential moving average (EMA) momentum = 0.99. A Reduce-LR-on-Plateau scheduler monitored validation loss and reduced the learning rate by a factor of 0.5 upon stagnation for five consecutive epochs, preventing oscillations around suboptimal minima.

The models were trained using Sparse Categorical Cross-Entropy loss, the standard objective for multiclass image classification. Validation performance was monitored on a stratified hold-out validation set (12.5% of total data), as described in the Data and Preprocessing subsection. No k-fold cross-validation was used to keep training computationally tractable during extensive model and hyperparameter benchmarking. Early stopping that monitors validation loss with a patience of 10 epochs and minimum delta of 0 was employed to halt training when validation performance ceased improving, preventing overfitting and unnecessary training time and computation.

A grid search was conducted over batch sizes {64, 32, 16, 8} and initial learning rates {1e-3, 5e-4, 2e-4, 1e-4, 5e-5} for each architecture to identify optimal hyperparameters. Each configuration was trained for up to 300 epochs and halted by early stopping, with the best checkpoint determined by the highest validation accuracy. Additional regularizations were handled implicitly through data augmentation, adding batch normalization layer and dropout layer with rate of 0.2, and early stopping. Training was performed on an NVIDIA T4 GPU with 16 GB memory.

D. Evaluation Metrics and Inference Benchmarking

Model performance was assessed using classification accuracy on both validation and held-out test sets. Accuracy was computed as the number of correctly classified samples divided by the total sample count per split. Inference latency was measured on an Intel Xeon 6 CPU (2.2 GHz, 2 vCPUs) of Google Colab without GPU acceleration to simulate low-resource deployment. Each model processed 720 frames of testing data in inference mode, and average per-frame latency (in milliseconds) was recorded. This CPU-based performance benchmarking aims to simulate a constrained computing environment that represents edge devices. The combination of accuracy and latency metrics provides a holistic view of each model's suitability for on-device cat skin disease diagnosis.

III. RESULTS AND DISCUSSION

This section presents the quantitative outcomes of the experiments and the analysis of the implications. The classification performance of each model on both validation

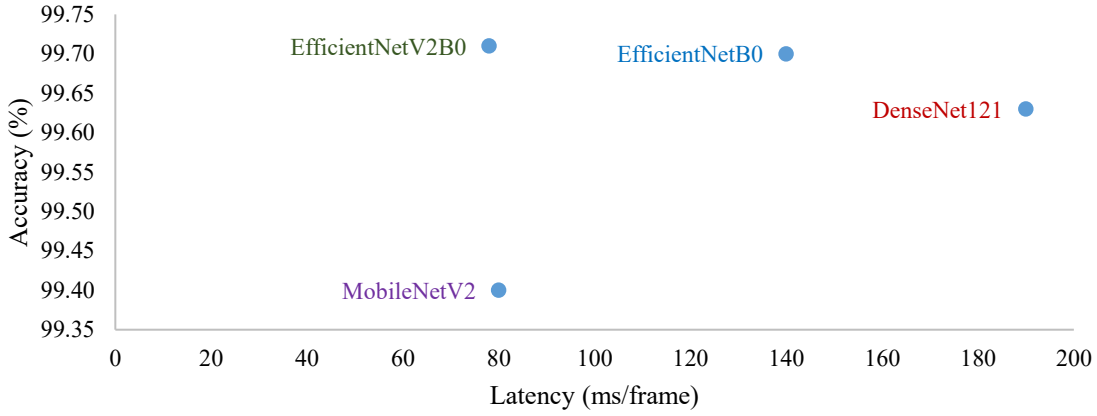

Figure 3. Scatter plot illustrating the trade-off between test accuracy (%) and average CPU inference latency (ms/frame) for the evaluated models.

and test sets is first examined, then inference latency on a CPU is evaluated to assess real-world deployment feasibility. Comparative analyses highlight trade-offs between accuracy and computational efficiency, guiding model selection for low-resource devices.

A. Classification Performance

The validation and test accuracies for all five architectures are summarized in Table II. Four of the five models achieved exceptionally high accuracies (> 99 %) on both splits, indicating that the augmentation strategy and fine-tuning protocol successfully captured discriminative features across nine dermatological categories. EfficientNetV2B0 and EfficientNetB0 delivered near-identical test performance (99.79 % vs. 99.78 %), consistent with their shared compound-scaling design. DenseNet121 achieved the highest validation accuracy (99.81 %) but exhibited a noticeable decline on the test set (99.44 %), suggesting mild overfitting despite early stopping and dynamic learning-rate adjustments. In contrast, MobileNetV2 performed slightly worse overall but maintained strong generalization (99.38 % test accuracy), reflecting the effectiveness of its inverted-residual bottlenecks under constrained computational budgets.

A key architectural insight emerges when comparing these models. EfficientNetV2B0 benefits from fused MBConv layers, which simplify depthwise-separable operations into more CPU-friendly fused convolutions, accelerating both training and inference. EfficientNetB0 lacks this optimization, which likely contributes to its slower inference despite comparable accuracy. DenseNet121's dense connectivity enhances gradient flow but significantly increases feature-map concatenations, leading to higher memory access costs, an effect amplified on CPU-only execution.

In stark contrast, MobileNetV3 failed to generalize, yielding only 12.35 % validation and 8.79 % test accuracy. This dramatic underperformance likely stems from architectural incompatibility with specific dataset characteristics of this study or suboptimal hyperparameter convergence during fine-tuning. Its lightweight "NAS optimized" modules, while beneficial for certain mobile vision tasks, may insufficiently capture the complex texture and color variations present in dermatological images. Further hyperparameter exploration or custom module adaptation might be required to salvage MobileNetV3 for this domain.

Overall, EfficientNetV2B0 strikes the best balance between high accuracy and model robustness. Its marginal accuracy gains over EfficientNetB0 and MobileNetV2, coupled with more stable validation-test consistency than DenseNet121, position it as the preferred backbone for cat skin disease classification under resource constraints.

B. Inference Latency on CPU

Inference times, measured on an Intel Xeon 6 CPU (2.2 GHz, 2 vCPUs) of Google Colab for 720 frames of testing data, are plotted in Figure 2. EfficientNetV2B0 processes each image in 78 ms, slightly faster than MobileNetV2's 80 ms, demonstrating its optimized fused MBConv layers and effective compound scaling. MobileNetV2, despite its inverted-residual design, trails by only 2 ms/frame, confirming its continued relevance as a lightweight backbone. In contrast, EfficientNetB0's 140 ms/frame and DenseNet121's 190 ms/frame latencies reveal substantial computational overhead, with DenseNet121 nearly tripling the inference time of EfficientNetV2B0.

These results underscore that, on typical CPU hardware, only EfficientNetV2B0 and MobileNetV2 achieve near-real-time throughput (≈12-13 fps), whereas EfficientNetB0 and DenseNet121 fall below 10 fps; rendering them less suitable for interactive or on-device applications without further optimization. EfficientNetV2B0's sub-80 ms latency thus remains the best balance of speed and accuracy for low-resource deployment.

Crucially, sub-100 ms per frame performance ensures that even with modest processing budgets, such as Raspberry Pi or mid-range smartphones, the model may deliver over 10 fps, enabling instantaneous feedback. This capability transforms the diagnostic pipeline from a batch-oriented analysis into an interactive tool for veterinarians and pet owners.

C. Trade-off Analysis and Practical Implications

Balancing accuracy and latency are vital for deployment on low-resource devices. Figure 3 schematically illustrates this trade-off. While DenseNet121 slightly edges out in validation accuracy, its doubled inference time relative to EfficientNetV2B0 undermines its practicality in time-sensitive or energy-constrained settings. Conversely, MobileNetV3, despite its minimal parameters, fails to meet basic accuracy requirements, highlighting that extreme compression without domain-specific adaptation can be counterproductive.

EfficientNetV2B0 emerges as the optimal compromise, delivering near-peak accuracy with minimal latency. Its efficient scaling strategy preserves high-resolution feature sensitivity, which is valuable for subtle dermatological patterns, while maintaining low inference cost. MobileNetV2 remains a

strong candidate for devices with tighter memory budgets due to its smaller parameter count despite slightly lower accuracy.

In future work, extending evaluation to other hardware platforms (e.g., ARM-based processors) and integrating quantization techniques could further enhance efficiency. Additionally, exploring lightweight attention mechanisms or knowledge-distillation strategies may unlock further performance gains without compromising speed. Nonetheless, current results of this study establish EfficientNetV2B0 as a compelling solution for rapid, accurate cat skin disease diagnosis in low-resource environments.

IV. CONCLUSION

In this work, EfficientNetV2B0 was evaluated for the classification of nine feline dermatological conditions under edge-device constraints. When fine-tuned on an augmented dataset, EfficientNetV2B0 achieved a test accuracy of 99.79 % while sustaining sub-100 ms per-frame inference latency on a standard CPU-only environment. Through comprehensive evaluations against established architectures (DenseNet121, MobileNetV2, MobileNetV3, and EfficientNetB0), EfficientNetV2B0 consistently occupied the Pareto-optimal frontier, balancing representational capacity and computational efficiency. The systematic augmentation pipeline and hyperparameter grid search ensured model robustness across varied lighting, pose, and lesion presentations, addressing common challenges in veterinary dermatological imaging.

REFERENCES

[1] D. W. Scott and M. Paradis, "A survey of canine and feline skin disorders seen in a university practice: Small Animal Clinic, University of Montréal, Saint-Hyacinthe, Québec (1987-1988)," *Can Vet J*, vol. 31, no. 12, pp. 830–835, Dec. 1990.

[2] P. B. Hill *et al.*, "Survey of the prevalence, diagnosis and treatment of dermatological conditions in small animals in general practice," *Veterinary Record*, vol. 158, no. 16, pp. 533–539, Apr. 2006.

[3] M. Hermoso de Mendoza *et al.*, "A zoonotic ringworm outbreak caused by a dysgonic strain of Microsporum canis from stray cats," *Rev Iberoam Micol*, vol. 27, no. 2, pp. 62–65, 2010.

[4] Z. Li, K. C. Koban, T. L. Schenck, R. E. Giunta, Q. Li, and Y. Sun, "Artificial Intelligence in Dermatology Image Analysis: Current Developments and Future Trends," *J Clin Med*, vol. 11, no. 22, Nov. 2022.

[5] M. Tan and Q. Le, "EfficientNetV2: Smaller Models and Faster Training," in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, Eds., in Proceedings of Machine Learning Research, vol. 139. PMLR, Aug. 2021, pp. 10096–10106.

[6] N. Kurniati, Y. Yanitasari, D. Lantana, I. Karima, and E. Susanto, "Sistem pakar untuk mendiagnosa penyakit kulit pada kucing menggunakan Certainty Factor," *ILKOM Jurnal Ilmiah*, vol. 9, no. 1, pp. 34–41, 2017.

[7] Novi, P. B. R. Putri, and Richiandrianto, "Penerapan sistem pakar dalam mendiagnosis penyakit kulit pada kucing menggunakan metode Certainty Factor (CF) berbasis web," *Jurnal Ilmu Komputer*, vol. 8, no. 6, pp. 86–87, 2024.

[8] R. Pratiwi, "Web-Based Expert System for Early Diagnosis of Skin Diseases in Cats Using the Naïve Bayes Method," *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, vol. 3, no. 1, pp. 491–497, Oct. 2023.

[9] F. Dwiramadhan, M. I. Wahyuddin, and D. Hidayatullah, "Sistem Pakar Diagnosa Penyakit Kulit Kucing Menggunakan Metode Naive Bayes Berbasis Web," *Jurnal JTIK (Jurnal Teknologi Informasi Dan Komunikasi)*, vol. 6, no. 3, pp. 429–437, 2022.

[10] Y. Wilantikasari, "Klasifikasi Penyakit Kulit Kucing Menggunakan Metode Support Vector Machine," Universitas Brawijaya, 2019.

[11] T. Widiyaningtyas, I. Made Wirawan, and S. Halimatus Mahmud, "Diagnosis of Feline Skin Disease Using C4.5 Algorithm," in *2020 Fifth International Conference on Informatics and Computing (ICIC)*, 2020, pp. 1–5.

[12] M. Salsabila, "Sistem pakar diagnosis penyakit kulit pada kucing jenis Persia dengan metode fuzzy Sugeno," Universitas Sriwijaya, Palembang, Indonesia, 2022.

[13] I. Y. Pangestu and S. R. Ramadhani, "Perancangan Sistem Deteksi Penyakit Kulit Pada Kucing Menggunakan Deep Learning Berbasis Android," *Teknika*, vol. 12, no. 3, pp. 173–182, Nov. 2023.

[14] B. Meilita and W. Yustanti, "Cat Skin Disease Detection System Using You Only Look Once (YOLO) v8 Algorithm: Sistem Deteksi Penyakit Kulit Kucing Menggunakan Algoritma You Only Look Once (YOLO) v8," *Journal of Emerging Information Systems and Business Intelligence*, vol. 5, no. 2, pp. 178–188, Jun. 2024.

[15] M. Juwita and M. H. Ambar, "Sistem Pakar Untuk Mendiagnosa Penyakit Kulit Pada Kucing Berasis Android Menggunakan Metode Forward Chaining," *Journal of Information System and Artificial Intelligence*, vol. 5, no. 1, pp. 192–201, Nov. 2024.

[16] K. Trinanda Putra, H. Zidni, R. Turrizka, H.-T. Chu, D.-T. Vu, and Prayitno, "An 8-bit Quantized Globalization Model for Cat Skin Disease Detection Using Convolutional Neural Networks and TensorFlow Lite," in *2024 International Conference on Information Technology and Computing (ICITCOM)*, 2024, pp. 323–327.

[17] A. Suryavanshi, V. Kukreja, P. Srivastava, A. Bhattacherjee, and R. S. Rawat, "Felis catus disease detection in the digital era: Combining CNN and Random Forest," in *2023 International Conference on Artificial Intelligence for Innovations in Healthcare Industries (ICAIIHI)*, 2023, pp. 1–7.

[18] S. Mehta and M. Rastegari, "MobileViT: Light-weight, General-purpose, and Mobile-friendly Vision Transformer," in *International Conference on Learning Representations*, 2022.

[19] Y. Chen *et al.*, "Mobile-Former: Bridging MobileNet and Transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 5270–5279.

[20] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[21] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269.

[22] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[23] A. Howard *et al.*, "Searching for MobileNetV3," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1314–1324.

[24] M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, Eds., in Proceedings of Machine Learning Research, vol. 97. PMLR, Aug. 2019, pp. 6105–6114.

[25] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *The 3rd International Conference for Learning Representations (ICLR)*, San Diego, 2015.