CTU Journal of Innovation and Sustainable Development
ISSN 2588-1418 | e-ISSN 2815-6412

DOI:10.22144/ctujoisd.2026.016

A comparative deep learning approach for image classification and retrieval in scientific publications

Nguyen Hoang Anh[1], and Tran Thanh Dien[2*]

[1]*Fpt Software Hochiminh Co., Ltd - Can Tho Branch, Viet Nam*

[2]*Can Tho University Publishing House, Can Tho University, Viet Nam*

**Corresponding author (thanhdien@ctu.edu.vn)*

Article info.

Received 22 Sep 2025
Revised 10 Oct 2025
Accepted 15 Feb 2026

Keywords

Comparative analysis, Content-based image retrieval (CBIR), Deep learning, FAISS, Grad-CAM, Image classification

ABSTRACT

This study presents a comparative analysis of state-of-the-art deep learning models–EfficientNetB0, MobileNetV2, and ResNet101–for image classification and content-based retrieval in scientific publications. A dataset of 4,303 images from 11 categories was curated from the Can Tho University Journal of Science and enhanced through tailored data augmentation strategies. The models were fine-tuned using transfer learning with hyperparameters optimised via Grid Search. Features were extracted using GlobalAveragePooling2D, and cosine similarity was combined with the FAISS library for efficient similarity search. Experimental results demonstrate a clear performance-efficiency trade-off: ResNet101 achieved the highest classification accuracy, while EfficientNetB0 and MobileNetV2 offered significant advantages in inference speed. A user-friendly web interface was developed to support practical image retrieval applications. These findings highlight the potential of deep learning in enhancing the management and integrity of scientific image resources.

1. INTRODUCTION

The proliferation of digital scholarship has led to an exponential increase in the use of images within scientific publications. These images serve as vital evidence to support research findings, illustrate complex phenomena, and enhance the communicative power of scholarly articles. However, this growing reliance on visual data has also exacerbated the problem of image plagiarism, wherein images are reused without proper attribution or authorisation. Such practices not only violate academic integrity but also undermine the credibility and reproducibility of scientific research. While text-based plagiarism detection tools are now widely adopted, automated detection of image reuse remains a significant challenge due to the high variability, contextual ambiguity, and ease of manipulation inherent in visual content.

Recent advances in deep learning, particularly convolutional neural networks (CNNs), have demonstrated exceptional capabilities in image understanding, enabling automated feature extraction and semantic similarity measurement with human-like accuracy. Capitalising on these developments, this study investigates the application of three state-of-the-art CNN architectures–EfficientNetB0, MobileNetV2, and ResNet101–for the dual tasks of image classification and content-based image retrieval (CBIR) in scientific publications. The primary objective is to develop a robust system capable of categorising scientific images into semantically meaningful classes and retrieving visually similar

instances from a large corpus, thereby facilitating the detection of potential image reuse and supporting academic integrity efforts.

Beyond technical contributions, this research also emphasises practical applicability through the development of an end-to-end web-based retrieval system. By integrating optimised deep learning models with efficient similarity search via the FAISS library, the proposed system offers a scalable and user-friendly solution for real-world image retrieval tasks. Ultimately, this work aims to enhance the management, accessibility, and ethical use of image-based resources in academic publishing.

2. RELATED WORK

The task of detecting duplicated or manipulated images in scientific literature, a key application of content-based image retrieval (CBIR), has garnered significant attention in recent years. Early CBIR systems often relied on handcrafted features such as SIFT (Lowe, 2004) for similarity matching. However, the advent of deep learning, particularly Convolutional Neural Networks (CNNs), has revolutionised the field by enabling the learning of highly discriminative and semantic feature representations directly from data (Simonyan & Zisserman, 2014; He et al., 2016).

A seminal contribution to image forensics is the work of Jha et al. (2018), "Image Forensics: Detecting duplication of scientific images with manipulation-invariant image similarity". The authors introduced a Siamese CNN framework designed to learn features that remain consistent across common transformations such as rotation, cropping, and brightness adjustment. By training on pairs of similar and dissimilar images, their system achieved high accuracy in identifying duplicated content, demonstrating strong robustness against minor modifications.

Expanding on the integration of advanced neural network architectures, Gayadhankar et al. (2021) proposed an image plagiarism detection system that combined Generative Adversarial Networks (GANs) with CNNs. In their framework, CNNs were trained to distinguish authentic images from those generated or altered by GANs. They further developed a practical Flask-based web application that allows users to upload and compare images for similarity, demonstrating strong performance even against rotated, cropped, or brightness-adjusted images.

In the Vietnamese context, Tran et al. (2025) developed an image similarity detection system for scientific publications using a ResNet50-based deep learning model combined with image processing techniques. Their approach leveraged feature extraction and classification to identify visually similar images within a curated dataset. The study compared ResNet50 with AlexNet and VGG16, demonstrating ResNet50's superior generalisation capability for complex image recognition tasks.

For efficient similarity search in high-dimensional spaces, we leverage the FAISS library (Johnson et al., 2019). FAISS provides optimised algorithms for fast nearest neighbour search, which is crucial for scaling retrieval systems to large datasets, a challenge often encountered in academic image databases.

The present study builds upon these foundations by conducting a comprehensive comparative analysis of three modern CNN architectures–EfficientNetB0 (Tan & Le, 2019), MobileNetV2 (Sandler et al., 2018), and ResNet101 (He et al., 2016)–for both image classification and retrieval. Unlike prior work, which often focuses on a single model, we systematically evaluate trade-offs between accuracy and computational efficiency. Furthermore, we implement an end-to-end web-based retrieval system using Flask and ReactJS, integrated with FAISS for efficient search, demonstrating a practical application for enhancing academic integrity and resource management.

3. METHODS

3.1. System architecture

The overall architecture of the proposed system is illustrated in Figure 1 and consists of the following steps:

Step 1: Image extraction and annotation. Images and metadata were extracted from PDF files of scientific articles using the PyMuPDF library. Each image was annotated with its corresponding metadata, including filename, article title, author(s), approval date, and DOI.

Step 2: Dataset preparation. The raw dataset was divided into training/validation (85%) and testing (15%) subsets. For the training/validation set, two different strategies were adopted:

- Method 1: Direct split of the original dataset, where 70% was used for training and 15% for validation.

• Method 2: Application of data augmentation before splitting, ensuring that the training set (85%) and validation set (15%) contained diverse image variants.

Data augmentation was applied to increase robustness against small transformations, such as rotation, brightness adjustment, and noise injection, following the practices suggested by Ameen and Mohammed (2023). The augmentation strategies were tailored for each image type to simulate realistic manipulations better.

Step 3: Model development. Three deep learning architectures were employed:

• EfficientNetB0 (Zhou & Ma, 2022), which balances depth, width, and resolution to achieve competitive accuracy with fewer parameters.

• MobileNetV2 (Tragoudaras & Siozios, 2022), a lightweight CNN optimized for mobile and embedded applications, incorporating inverted residuals and linear bottlenecks.

• ResNet101 (He et al., 2015), a deep residual network with 101 layers, designed to overcome gradient vanishing in very deep CNNs.

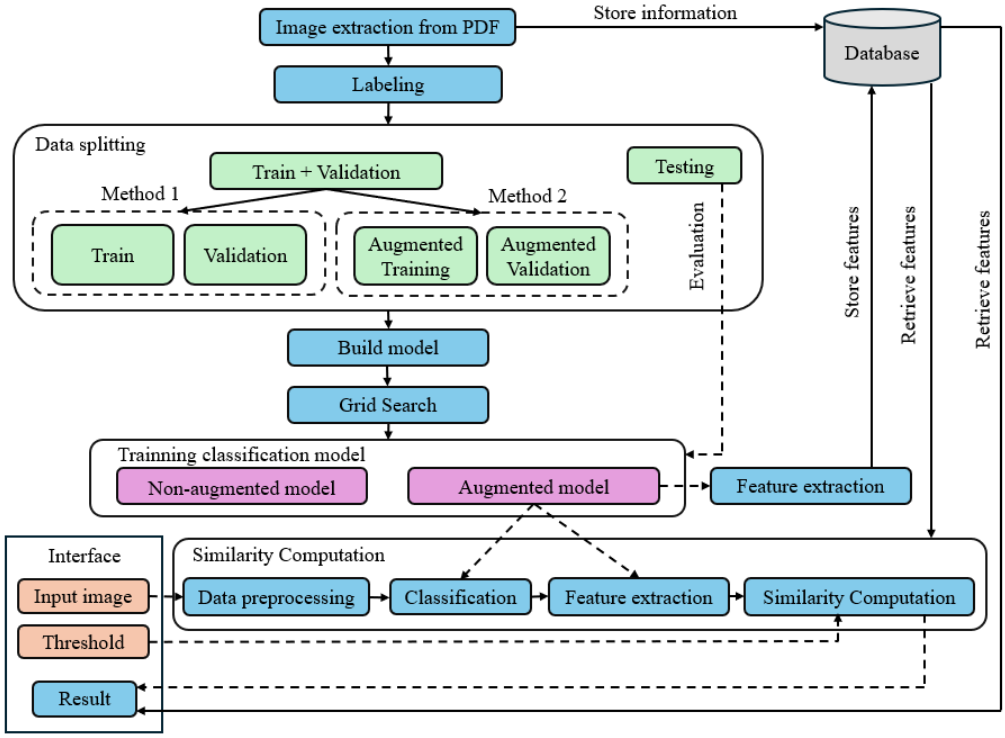

Figure 1. The system architecture

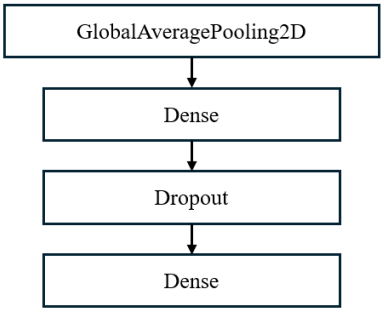

Figure 2. Output fully-connected layers of the model

All models were fine-tuned using transfer learning. After convolutional layers, features were aggregated through a GlobalAveragePooling2D layer (Lin et al., 2014), followed by fully connected layers with ReLU activation, dropout for regularisation, and a final softmax output layer with 11 units (Falaschetti et al., 2022). Models were compiled using categorical cross-entropy loss and the Adam optimiser. Figure 2 illustrates the output structure of the fully connected layers.

Step 4: Hyperparameter optimisation. Grid Search (Bergstra & Bengio, 2012) was used to tune dropout rates, dense layer sizes, and learning rates. The

parameter space included Dropout = {0.1, 0.2, 0.3, 0.4, 0.5}, Dense units = {64, 128, 256, 512, 1024}, and Learning rate = {1e-4, 1e-5}, resulting in 50 configurations (5 x 5 x 2). Each configuration was trained for up to 30 epochs, and the best-performing setup was selected based on validation accuracy (Ogunsanyaa & Ibekwe, 2023) (Table 1) and Table 2).

Step 5: Model training. Images were resized to 224×224 and normalised to [0,1]. Models were trained with a batch size of 16 for up to 30 epochs. Several callbacks were applied: ModelCheckpoint to save the best model, EarlyStopping to halt training if validation loss did not improve after five epochs, and ReduceLROnPlateau to adapt the learning rate when training plateaued.

Table 1. Optimal parameter tuning results with the raw dataset

Hyperparameters	ResNet101	EfficientNetB0	MobileNetV2
Dropout	0.5	0.5	0.2
Dense	512	128	64
Learning rate	0.0001	0.0001	0.0001

Table 2. Optimal parameter tuning results with the augmented dataset

Hyperparameters	ResNet101	EfficientNetB0	MobileNetV2
Dropout	0.4	0.4	0.1
Dense	128	512	128
Learning rate	0.0001	0.0001	0.0001

Step 6: Feature extraction. Once trained, the models were used to extract feature vectors by discarding the final softmax layer and retaining the output of the GlobalAveragePooling2D layer (Chang et al., 2024). These vectors capture semantic and spatial information from the images and are stored with metadata for retrieval. The FAISS library (Johnson et al., 2019) was employed for efficient similarity search in high-dimensional feature spaces (Figure 3).

Step 7: Similarity computation and web deployment. Cosine similarity was used to measure the similarity between query images and database vectors.

A Flask–ReactJS web application was implemented, allowing users to upload images or PDFs, specify a similarity threshold, and retrieve relevant images with associated metadata, including similarity score, title, author, and DOI.

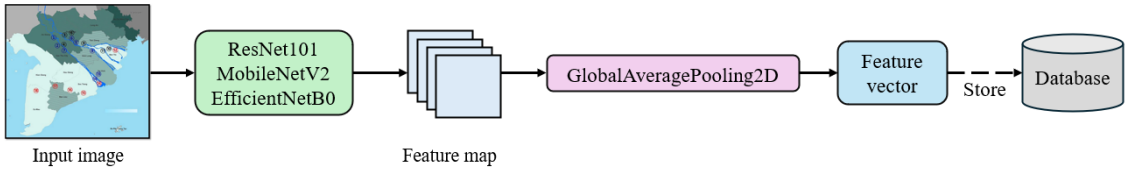

Figure 3. The process of feature extraction

3.2. Data description

The dataset used in this study consists of 4,303 images directly extracted, curated, and annotated from scientific articles published in the Can Tho University Journal of Science. The collection includes both Vietnamese- and English-language articles. Each extracted image was assigned to one of 11 categories: Animal Samples, Charts, Chemical Structures, Maps, Microbial Cultures, Micrographs, Models and Diagrams, PCR Gel Electrophoresis, Photographs, Plant Samples, and Others. Table 3 summarises the class distribution of the raw dataset.

Table 3. Raw dataset

Classes	Number
Animal Samples	154
Charts	1,668
Chemical Structure	126
Maps	210
Microbial Cultures	180
Micrographs	448
Models And Diagrams	409
PCR Gel Electrophoresis	129
Photographs	322
Plant Samples	375
Others	282
Sum	4,303

To enhance variability and improve model robustness, each original image was augmented into five variants using transformations such as rotation, brightness adjustment, and noise injection. Importantly, augmentation strategies were tailored to specific categories to preserve the visual characteristics of each image type. This process resulted in an augmented dataset of 22,548 images, as detailed in Table 4.

For both the raw and augmented datasets, data were partitioned into training (train), validation (val), and testing (test) sets to ensure fair evaluation. Table 5 provides the detailed distribution across these splits on the raw dataset.

Table 4. Augmented dataset

Classes	Number
Animal Samples	804
Charts	8,753
Chemical Structure	656
Maps	1,100
Microbial Cultures	940
Micrographs	2,348
Models And Diagrams	2,144
PCR Gel Electrophoresis	674
Photographs	1,687
Plant Samples	1,965
Others	1,477
Sum	22,548

Table 5. Split the raw dataset

Classes	Train	Val	Test
Animal Samples	107	23	24
Charts	1,167	250	251
Chemical Structure	88	18	20
Maps	147	31	32
Microbial Cultures	125	27	28
Micrographs	313	67	68
Models and Diagrams	286	61	62
PCR Gel Electrophoresis	90	19	20
Photographs	225	48	49
Plant Samples	262	56	57
Others	197	42	43
Sum	3,007	642	654

Table 6 provides the detailed distribution across these splits after applying data augmentation techniques. Additionally, the dataset has been made publicly available on the Kaggle platform to support further research in scientific image classification and knowledge discovery.

Table 6. Split the augmented dataset

Classes	Train	Val	Test
Animal Samples	663	117	24
Charts	7,226	1,276	251
Chemical Structure	540	96	20
Maps	907	161	32
Microbial Cultures	775	137	28
Micrographs	1,938	342	68
Models and Diagrams	1,769	313	62
PCR Gel Electrophoresis	555	99	20
Photographs	1,392	246	49
Plant Samples	1,621	287	57
Others	1,218	216	43
Sum	18,604	3,290	654

3.3. Evaluation metrics

Model performance was assessed using a range of classification and retrieval metrics. For image classification, the following measures were employed:

- Accuracy and Loss to evaluate overall correctness and convergence stability.
- Precision, Recall, and F1-score to capture class-wise performance, especially in imbalanced datasets.
- AUC (Area Under the ROC Curve) to assess the discriminative ability of the models under class imbalance conditions (Li, 2024).

For image retrieval, the evaluation focused on similarity-based measures:

- Average Precision (AP): calculated as the area under the Precision–Recall curve for each query, reflecting retrieval accuracy at different cutoff points.
- Mean Average Precision (mAP): the mean of AP scores across all queries, serving as the principal retrieval performance metric.
- Query Time: measured with and without FAISS integration to quantify computational efficiency during large-scale similarity searches.

These combined metrics provide a comprehensive assessment of classification effectiveness and

retrieval reliability, offering insights into trade-offs among accuracy, efficiency, and scalability.

4. EXPERIMENTAL RESULTS

4.1. Image Classification on the Raw Dataset

4.1.1. EfficientNetB0

Figure 4 presents the training performance of EfficientNetB0 on the raw dataset. The model achieved rapid convergence, reaching high accuracy and AUC after only a few epochs, while the loss decreased sharply and stabilised at a low level. The validation curves closely mirrored the training curves, indicating strong consistency and good generalisation. By epochs 4–5, performance metrics were already close to optimal. At its best performance (epoch 12), training accuracy reached approximately 0.986 and validation accuracy 0.933, with AUC values nearly perfect for both sets. Correspondingly, training loss was as low as 0.051 and validation loss was 0.310. Early stopping at epoch 14 successfully prevented overfitting and preserved generalisation.

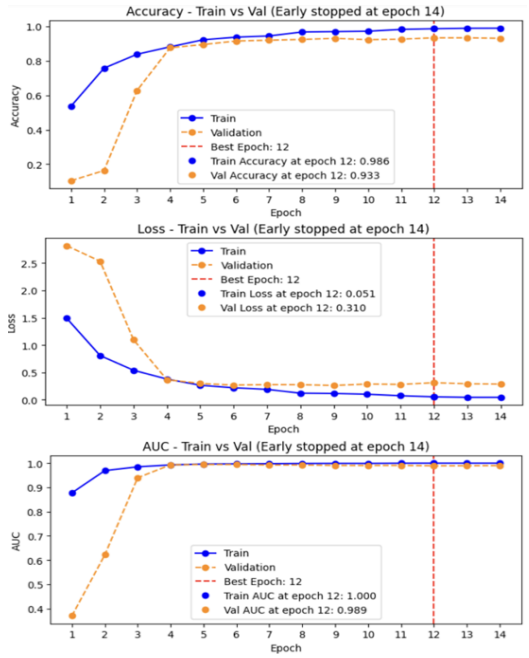

Figure 4. Performance visualization chart of EfficientNetB0 on the raw dataset

4.1.2. MobileNetV2

As shown in Figure 5 MobileNetV2 also converged quickly, with training and validation accuracy and AUC rising sharply within the first six epochs before stabilising. Training loss dropped from an initial value above 1.5 to 0.014, while validation loss decreased to 0.346 at the best-performing epoch. By epoch 23, training accuracy nearly reached 0.996 and validation accuracy remained high at 0.916. The AUC was close to 1.0 for training and 0.985 for validation, confirming stable classification performance. Notably, the validation curves closely followed the training curves throughout training, indicating limited overfitting. Early stopping at epoch 23 helped avoid overfitting and preserved long-term generalisation.

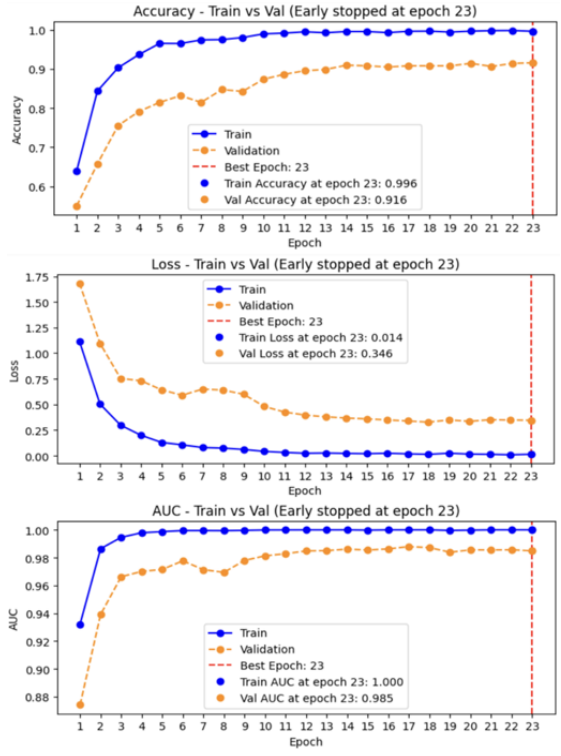

Figure 5. Performance visualisation chart of MobileNetV2 on the raw dataset

4.1.3. ResNet101

Figure 6 illustrates the training curve of ResNet101, which achieved high accuracy and AUC within the first few epochs. At its peak (epoch 21), training accuracy was nearly perfect, with an extremely low loss of 0.004, while validation accuracy remained high and validation loss reached 0.336. The training stopped at epoch 26 to mitigate overfitting, ensuring robust generalisation.

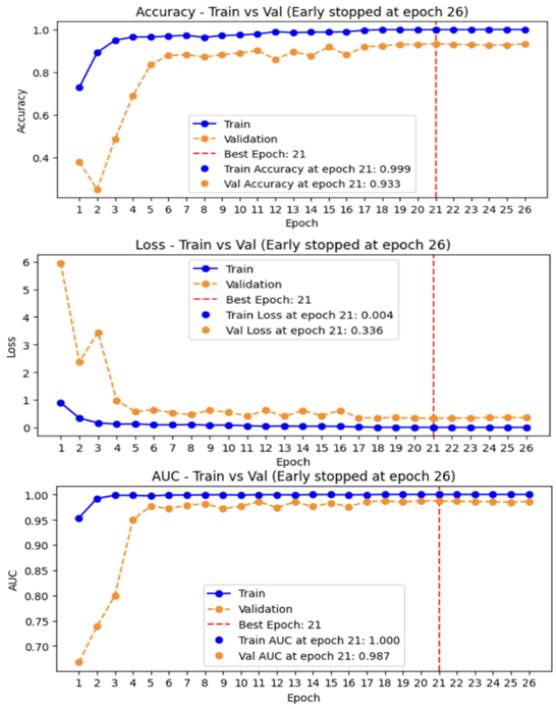

Figure 6. Performance visualization chart of ResNet101 on the raw dataset

4.2. Image Classification on the Augmented Dataset

4.2.1. EfficientNetB0

Training on the augmented dataset further improved performance (Figure 7). At epoch 15, EfficientNetB0 reached 0.996 training accuracy and 0.999 validation accuracy, with losses of 0.014 and 0.004, respectively. The extremely low validation loss compared to the raw dataset highlights the effectiveness of augmentation in enhancing model robustness. Both training and validation AUC values rapidly approached 1.0 after only a few epochs. Early stopping at epoch 20 indicated stable convergence without overfitting.

4.2.2. MobileNetV2

As shown in Figure 8, MobileNetV2 trained on augmented data, achieved excellent stability. From epoch 4 onwards, both training and validation accuracy exceeded 0.98, with loss values reduced to approximately 0.002. By epoch 10, validation accuracy had already reached 0.995 and validation loss had dropped below 0.01, indicating rapid and stable convergence. At epoch 14, validation accuracy reached 1.0 and training accuracy 0.999, while AUC was perfect (1.0) for both sets. The close overlap of training and validation curves throughout training further demonstrates the absence of overfitting. Early stopping at epoch 21 avoided unnecessary training and confirmed the model's generalization capability.

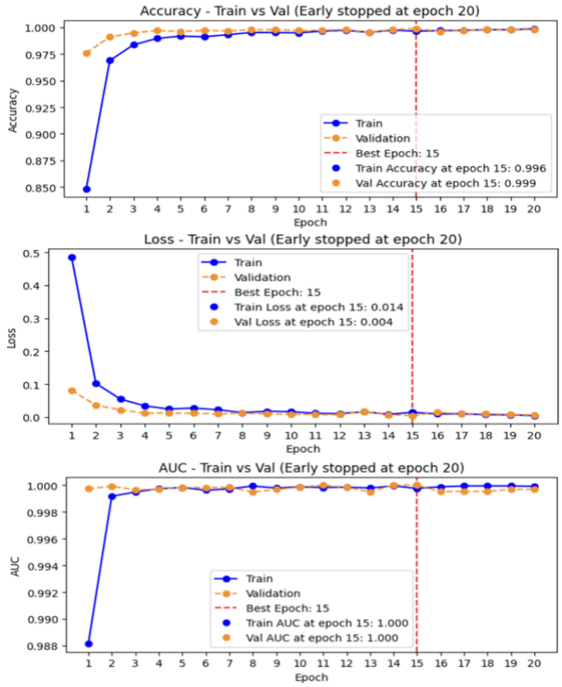

Figure 7. Performance visualization chart of EfficientNetB0 on the augmented dataset

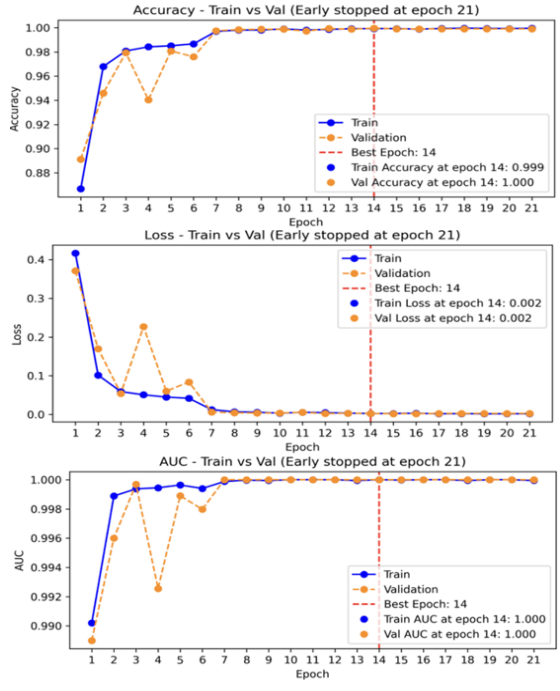

Figure 8. Performance visualization chart of MobileNetV2 on the augmented dataset

4.2.3. ResNet101

Figure 9 illustrates ResNet101 trained on the augmented dataset. Optimal performance was reached as early as epoch 9, with both training and validation accuracy at 0.999 and losses as low as 0.003. AUC values were already perfect by this stage. From epoch 4 onwards, validation accuracy exceeded 0.97 while loss dropped below 0.05, showing rapid convergence. By epoch 5, validation accuracy had already surpassed 0.98, and validation loss decreased to nearly 0.02, reflecting rapid improvement. At epoch 7, both training and validation AUC reached 1.0, confirming near-perfect separability at an early stage. Training was stopped at epoch 14, confirming outstanding classification accuracy and robustness without overfitting.

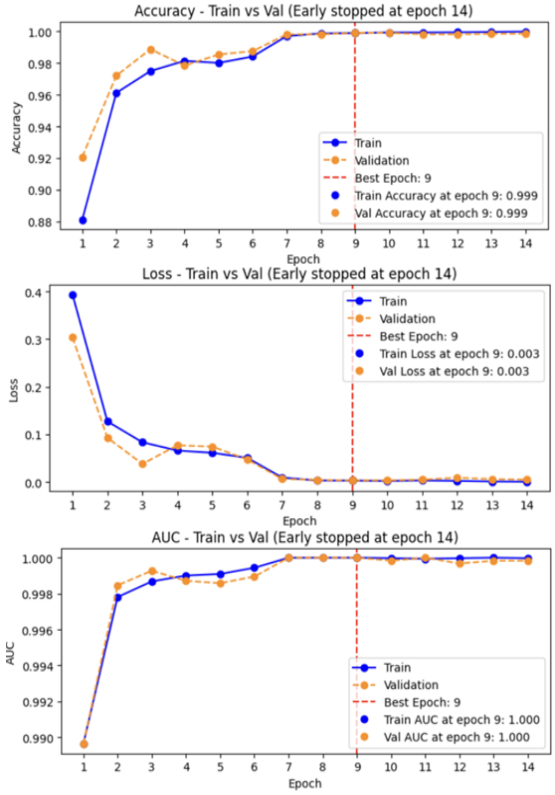

Figure 9. Performance visualisation chart of ResNet101 on the augmented dataset

4.3. Overall Classification Performance

Table 8 compares the three models across both raw and augmented datasets. With raw data, ResNet101 achieved the highest accuracy (0.92966) and F1-score (0.92999), outperforming EfficientNetB0 (Accuracy 0.92731, F1-score 0.92718) and MobileNetV2 (Accuracy 0.91896, F1-score 0.92378).

With augmented data, all models improved, with ResNet101 again leading (Accuracy 0.93972, F1-score 0.93849). EfficientNetB0 also performed strongly (Accuracy 0.93754, F1-score 0.93350), while MobileNetV2 showed moderate improvement (Accuracy 0.92966, F1-score 0.93169). The performance gains highlight the benefits of augmentation in reducing overfitting and enhancing generalisation.

In addition, Grad-CAM (Selvaraju et al., 2019) was employed to visualise the decision-making process of the models (Figure 10). For raw data, activation regions were limited to smaller areas, often missing contextual cues. In contrast, with augmented data, activation maps–particularly for ResNet101–covered broader, more relevant regions, demonstrating that augmentation enabled the models to leverage more comprehensive image information

4.4. Feature Extraction Results

Table 7 summarizes the feature dimensions. These feature vectors were extracted from the Global Average Pooling layer, which transforms the feature maps into compact representations. EfficientNetB0 and MobileNetV2 both generated 1,280-dimensional vectors, while ResNet101 produced 2,048-dimensional vectors, reflecting its deeper architecture.

Table 7. Feature map sizes and feature vector dimensions of the models

Models	Feature map	Feature vector
ResNet101	7 × 7 × 2,048	2,048
EfficientNetB0	7 × 7 × 1,280	1,280
MobileNetV2	7 × 7 × 1,280	1,280

Visualisation of the top-5 channels with the highest Global Average Pooling (GAP) values (Figure 11) highlighted architectural differences. ResNet101 exhibited sharp, localised activations with high GAP values (>5), confirming its ability to capture deep semantic features. In contrast, EfficientNetB0 showed weaker, more diffuse activations (<2.5), consistent with its lightweight design. MobileNetV2 achieved intermediate results, effectively balancing efficiency with robust feature extraction.

Table 8. Performance comparison of ResNet101, EfficientNetB0, and MobileNetV2

Models	Data	Accuracy	Loss	Precision	Recall	F1-score
ResNet101		**0.92966**	**0.31898**	**0.93811**	**0.92202**	**0.92999**
EfficientNetB0	Raw	0.92731	0.32891	0.93344	0.92102	0.92718
MobileNetV2		0.91896	0.33782	0.93023	0.91743	0.92378
ResNet101		**0.93972**	**0.30495**	**0.93954**	**0.93745**	**0.93849**
EfficientNetB0	Aug	0.93754	0.31586	0.93584	0.93119	0.93350
MobileNetV2		0.92966	0.31226	0.93529	0.92813	0.93169

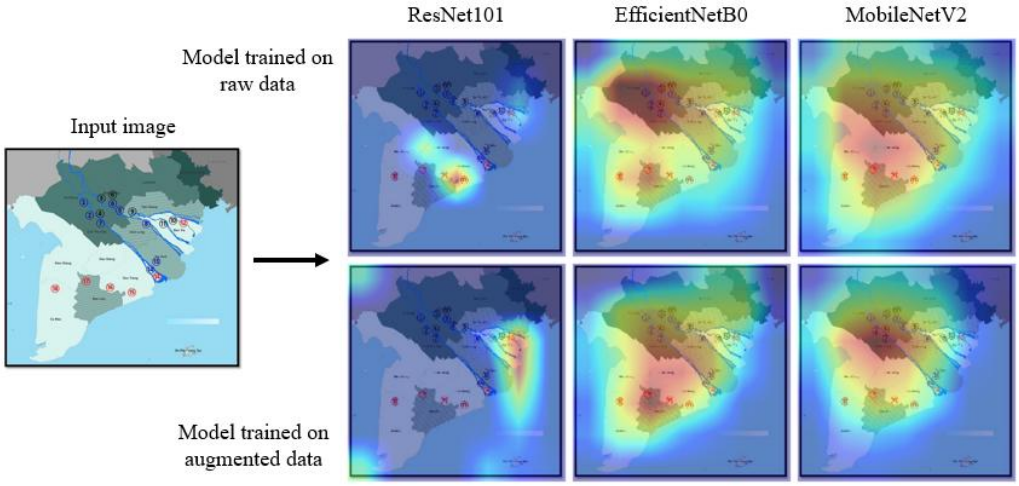

Figure 10. Visualisation of Grad-CAM results for the models

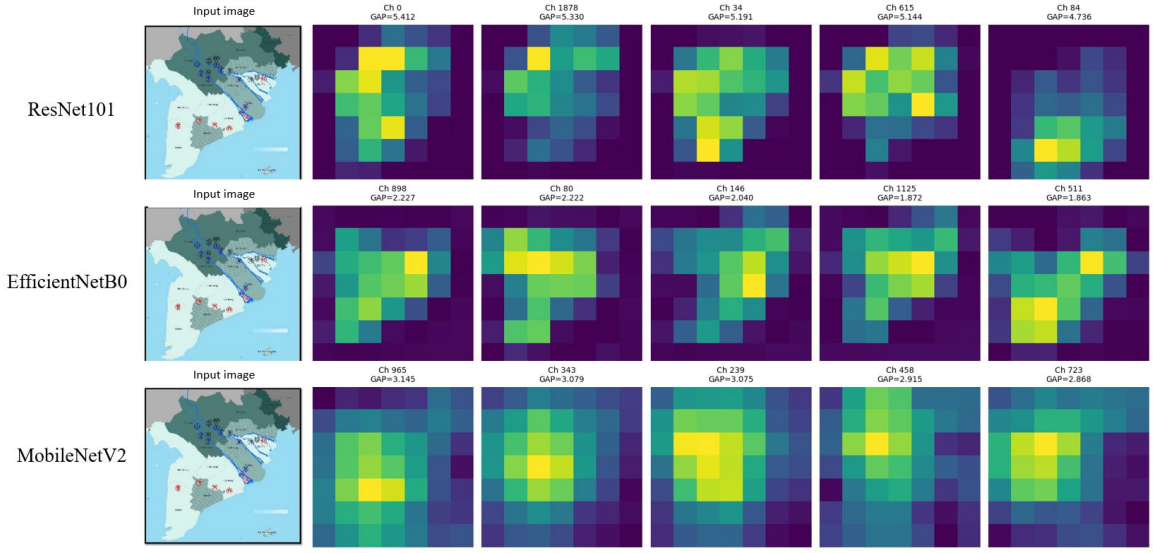

Figure 11. Feature visualisation of the models

4.5. Results of image similarity computation

Similarity retrieval was evaluated using cosine similarity at multiple thresholds. A clear trade-off was observed: Precision increased with higher thresholds, while Recall decreased sharply. At a threshold of 0.5, ResNet101 and EfficientNetB0 achieved Precision values of 0.9540 and 0.9502, but Recall was limited (0.3487 and 0.3096). At a threshold of 0.9, all models reached perfect Precision (1.0), but Recall dropped close to zero.

MobileNetV2 achieved the highest F1-score (0.8271) at threshold 0.5 due to its superior Recall

(0.7974), whereas ResNet101 and EfficientNetB0 were more conservative. Mean Average Precision (mAP) remained consistently high across models and thresholds, ranging from 0.9934–1.0000 for ResNet101, 0.9825–1.0000 for EfficientNetB0, and 0.9487–1.0000 for MobileNetV2 (Figure 12).

Efficiency was significantly improved by integrating FAISS. Without FAISS, similarity search took tens of milliseconds per query, with the time increasing with dataset size. With FAISS, retrieval time was reduced to just 4–6 ms, even for large datasets (Figure 13, Table 9). Although ResNet101 vectors required slightly longer retrieval times (6.5 ms) than EfficientNetB0 (3.9 ms) or MobileNetV2 (4.5 ms), the differences were negligible in practice.

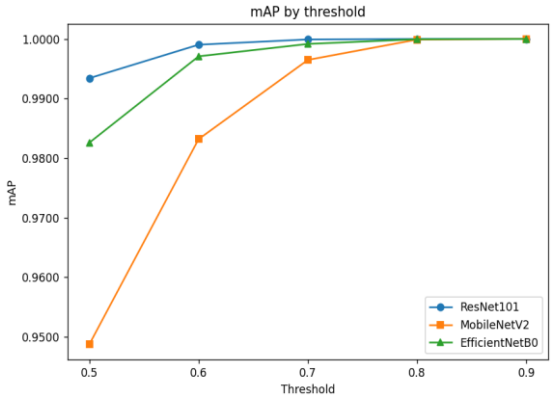

Figure 12. mAP chart across different thresholds

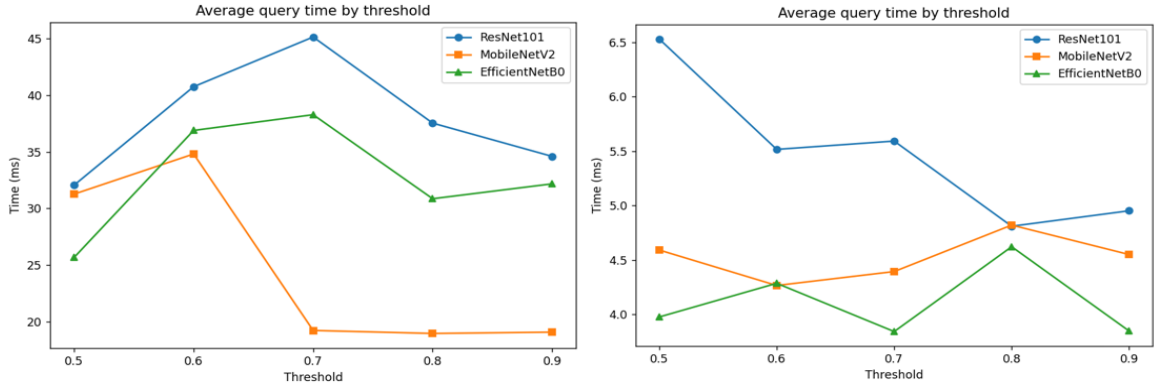

Figure 13. Average query time without FAISS (left) and with FAISS (right)

Table 9. Performance of similar image retrieval

Models	Threshold	Precision	Recall	F1-score	mAP	Time (ms)	FAISS (ms)
ResNet101	0.5	0.9540	0.3487	0.4376	0.9934	32.0	6.5
	0.6	0.9786	0.2149	0.2910	0.9990	40.7	5.5
	0.7	0.9946	0.1108	0.1669	0.9999	45.1	5.5
	0.8	**0.9986**	**0.0346**	**0.0610**	**1.0000**	**37.5**	**4.8**
	0.9	**0.9992**	**0.0064**	**0.0124**	**1.0000**	**34.6**	**4.9**
EfficientNetB0	0.5	0.9502	0.3096	0.3993	0.9825	25.6	3.9
	0.6	0.9837	0.1556	0.2117	0.9970	36.8	4.2
	0.7	0.9954	0.0684	0.0928	0.9991	38.2	3.8
	0.8	0.9985	0.0231	0.0365	0.9999	30.8	4.6
	0.9	**1.0000**	**0.0047**	**0.0093**	**1.0000**	**32.1**	**3.8**
MobileNetV2	0.5	0.8998	0.7974	0.8271	0.9487	31.2	4.5
	0.6	0.9440	0.6106	0.6856	0.9831	34.8	4.2
	0.7	0.9726	0.3677	0.4528	0.9964	19.2	4.3
	0.8	0.9912	0.1306	0.1857	0.9999	18.9	4.8
	0.9	**1.0000**	**0.0104**	**0.0193**	**1.0000**	**19.0**	**4.5**

4.6. System interfaces

Finally, a web-based prototype was implemented to demonstrate real-world usability. As shown in Figures 14 to 16, the interface allows users to upload an image or a PDF, specify a similarity threshold, and retrieve the most relevant images, along with metadata such as the similarity score, title, author, and DOI. The system supports both single-image and multi-image queries, providing interactive, real-time exploration of the dataset. Integration with FAISS ensured that retrieval remained efficient and scalable.

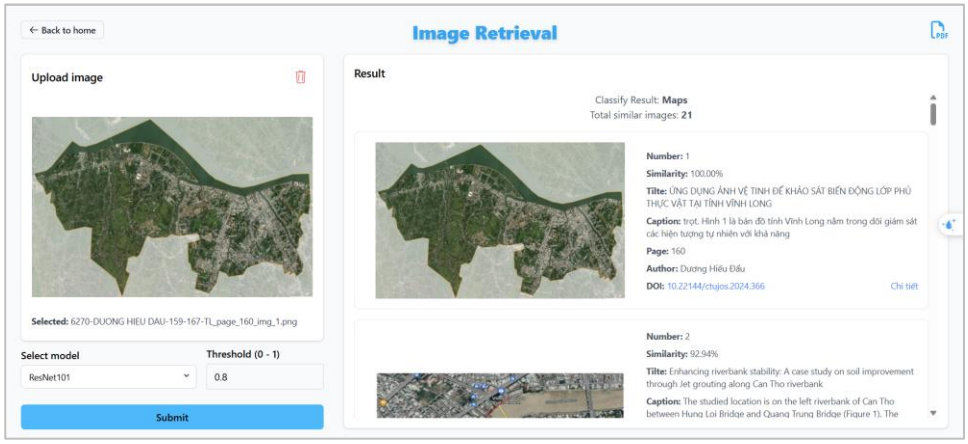

Figure 14. An interface for similar image retrieval with an input image

Figure 15. Interface for similar image retrieval with an input PDF file

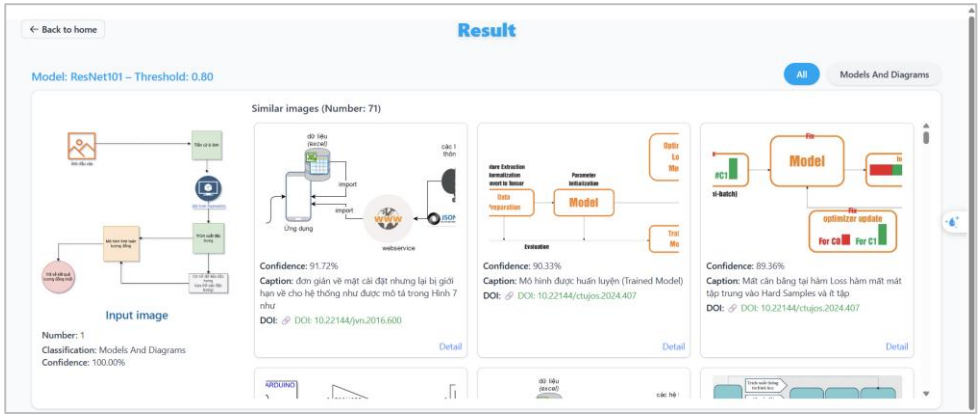

Figure 16. Similarity search results with the input PDF file

5. CONCLUSION

This study established a robust framework for scientific image analysis by three deep learning architectures: EfficientNetB0, MobileNetV2, and ResNet101. The results delineate a clear performance-efficiency trade-off: ResNet101 achieved superior classification accuracy and high-precision retrieval, whereas EfficientNetB0 and MobileNetV2 provided optimal efficiency for rapid inference tasks.

The integration of tailored data augmentation, validated through Grad-CAM visualisations, significantly enhanced model robustness and feature learning. Coupled with hyperparameter optimisation and FAISS for efficient indexing, our approach reduced retrieval latency to milliseconds, enabling real-time performance.

The practical viability of the system was demonstrated through an intuitive web interface, positioning it as a valuable tool for academic integrity monitoring and scholarly resource management. Future work will explore vision transformers, cross-modal retrieval, and larger multi-disciplinary datasets to further advance the capabilities of automated scientific image analysis.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest regarding the publication of this paper. The authors have no financial, professional, or personal relationships that could have influenced the research results or the interpretation of the findings.

REFERENCES

Ameen, Y. A., & Mohammed, D. (2023). Which data subset should be augmented for deep learning? A simulation study using urothelial cell carcinoma histopathology images. *BMC Bioinformatics*. https://doi.org/10.1186/s12859-023-05199-y

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research, 13*, 281–305. https://dl.acm.org/doi/10.5555/2188385.2188395

Chang, P., Sun, H., Lee, J. & Kim, H. (2024). Extraction and evaluation of features of preterm patent ductus arteriosus in chest X-ray images using deep learning. *Scientific Reports*. https://doi.org/10.1038/s41598-024-79361-8

Falaschetti, L., Manoni, L., Leoa, D., Paub, D., Tomasellic, V., & Turchettia, C. (2022). A CNN-based image detector for plant leaf diseases classification. *HardwareX, 12*. https://doi.org/10.1016/j.ohx.2022.e00363

Gayadhankar, K., Patil, R., Chavan, P., Channe, P., & Patil, S. (2021). Image plagiarism detection using GAN - (Generative Adversarial Network). *ITM Web of Conferences, 40*, 03013.https://doi.org/10.1051/itmconf/20214003013

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-78). https://doi.org/10.1109/CVPR.2016.90

Jha, A., Uppal, M., & Pelz, J. B. (2018). Image forensics: Detecting duplication of scientific images with manipulation-invariant image similarity. *arXiv preprint arXiv:1802.06515*. https://arxiv.org/abs/1802.06515

Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data, 7*(3), 535-547. https://doi.org/10.1109/TBDATA.2019.2921572

Li, J. (2024). Area under the ROC Curve has the most consistent evaluation for binary classification. *PLOS ONE*. https://doi.org/10.1371/journal.pone.0316019

Lin, M., Chen, Q., & Yan, S. (2014). Network In Network. *arXiv preprint arXiv:1312.4400*. https://arxiv.org/abs/1312.4400

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision, 60*(2), 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

Ogunsanyaa, M., & Ibekwe, J. (2023). Grid Search Hyperparameter Tuning in Additive Manufacturing Processes. *Manufacturing Letters, 36*, 1031-1042. https://doi.org/10.1016/j.mfglet.2023.08.056

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520). https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00474

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2019). Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *International Journal of Computer Vision, 128*, 336-359. https://doi.org/10.1007/s11263-019-01228-7

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. https://arxiv.org/abs/1409.1556

Tan, M., & Le, Q. V. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105-6114). PMLR. https://arxiv.org/abs/1905.11946

Tragoudaras, A., & Siozios, P. (2022). Design Space Exploration of a Sparse MobileNetV2 Using High-Level Synthesis and Sparse Matrix Techniques on FPGAs. *Sensors, 22*(12), 4318. https://doi.org/10.3390/s22124318

Zhou, A., & Ma, Y. (2022). Multi-head attention-based two-stream EfficientNet for action recognition. *Multimedia Systems, 28*, 487-498. http://dx.doi.org/10.1007/s00530-2022-00961-3