Addressing overfitting in comparative study for deep learning-based classification

Jing-Yee Ong, Lee-Yeng Ong, Meng-Chew Leow
Faculty of Information Science and Technology, Multimedia University, Melaka, Malaysia

Article Info

Article history:

Received Jul 3, 2024
Revised Feb 28, 2025
Accepted Mar 11, 2025

Keywords:

Deep learning
Feature extraction
Few-shot learning
InceptionV3
Overfitting
Stanford dog
Xception

ABSTRACT

Despite significant advancements in deep learning methodologies for animal species classification, there remains a notable research gap in effectively addressing biases inherent in training datasets, combating overfitting during model training, and enhancing overall performance to ensure reliable and accurate classification results in real-world applications. Therefore, this study explores the complex challenges of dog species classification, with a specific focus on addressing biases, combatting overfitting, and enhancing overall performance using deep learning methodologies. Initially, the Stanford Dog dataset serves as the foundation for training, complemented by additional data from annotated datasets. The primary aim is to mitigate biases and reduce overfitting, which is essential for improving the performance of deep learning-based classification in terms of dataset size and computational time. Feature extraction and few-shot learning techniques are compared to assess and improve the model performance. The experimentation involves the utilization of optimal classifiers, specifically InceptionV3 and Xception. In order to tackle overfitting, a range of strategies are deployed, including data augmentation, early stopping, and the integration of dropout and freezing layers which particularly achieved a better performance with Xception on the augmented dataset.

Corresponding Author:

Lee-Yeng Ong
Faculty of Information Science and Technology, Multimedia University
Jalan Ayer Keroh Lama 75450 Melaka, Malaysia
Email: lyong@mmu.edu.my

1. INTRODUCTION

In today's world, the increasing number of animal species poses a challenge for humans to accurately differentiate between them, especially considering the similarities shared by some species. While classifying dogs can be done through an expert-based approach, where individuals with extensive knowledge of dog breeds make the classifications. However, this method is challenging due to the scarcity of experts. Another approach involves genetic testing, but it is both costly and time-consuming, especially considering the vast number of dog breeds worldwide, currently totaling 20,580 [1]. Hence, dog classification model has been developed to provide a more efficient and accessible solution to accurately differentiate between various dog breeds [2]-[4].

Overfitting is a common challenge in most of the existing research because a model for recognizing different types of dog species might focus too much on specific details from the available training data. This could cause it to perform poorly when attempting to identify unfamiliar creatures and make it even more difficult to differentiate between various species. Multiple studies highlighted the dataset imbalance, with a minimum of 10% of images featuring human interference has caused the model trained too long with too many parameters, which led to the reduction of validation accuracy [5]-[7]. The overfitting issue became worse when a model like InceptionV3 was originally trained on the ImageNet library, and applying this pre-trained knowledge to a new problem made the overfitting more severe [2].

There is a study that has shown that the integration of feature extraction technique can extract more information, making it beneficial for training models [8] but another research that relied solely on convolutional neural networks (CNN) for feature extraction may have overlooked essential information, potentially leading to inaccurate classification [9]. Besides, few-shot learning enables rapid learning from limited data, potentially reducing training time compared to traditional techniques requiring large datasets. The method tackles bias by adjusting decision boundaries for fair predictions. Through rigorous testing, the effectiveness of few-shot learning is proven in maintaining both accuracy and fairness across unseen tasks with limited training data [10].

This study aims to resolve the overfitting issue from classifier Xception and InceptionV3. After that, two techniques include feature extraction and few-shot learning are applied for evaluating the performance of model in terms of training time and handle overfitting issue. The contributions of this paper include: i) the methodology to address the overfitting issues in dog species classification and ii) comparing the impact of feature extraction and few-shot learning techniques to determine if these approaches improve or worsen the model's performance and overfitting.

The rest of this paper is organized as follows. In section 2, related work is introduced to explain the structure of Xception, Inception-V3, overfitting, feature extraction and few-shot learning. Section 3 discusses the methodology used in this study. In section 4, the performance of the proposed method is discussed and analyzed. Finally, section 5 provides a summary of the model's performance and offers suggestions for future work.

2. RELATED WORK

Several studies revealed that numerous approaches have been attempted for dog classification using InceptionV3 and Xception. A comparative analysis of two deep learning models, InceptionV3 and Visual Geometry Group 16 (VGG16), for dog breed classification using the Stanford Dog dataset has addressed the challenge of differentiating between similar-looking breeds through the utilization of transfer learning and data augmentation techniques [1]. InceptionV3 achieved a significantly higher accuracy of 85% compared to VGG16's 69%, highlighting the effectiveness of transfer learning in enhancing model performance for this task [1].

Besides, the study on dog breed identification using Xception model also achieved a superior performance with a validation accuracy of 91.9% over VGG19, neural architecture search (NAS) network mobile (NetMobile), and EfficientNet version 2 medium (EfficientNetV2M) [11]. However, the model combining two pre-trained models from InceptionV3 and Xception, shown a superior performance with an accuracy of 92.4% [11]. The authors addressed the issue of overfitting by employing transfer learning and data augmentation techniques, demonstrating the effectiveness of combining pre-trained models and reducing overfitting problem [11]. Applying dropout and freeze layer are recognized as the effective ways to overcome overfitting problem [12]-[14]. These techniques, combined with careful dataset management and diagnostic tools like learning curves, can significantly reduce the risk of overfitting [15]-[18]. The learning curve illustrated in Figure 1 offers insights into the model's learning progression over time [19], [20]. If the training error, represented by the blue line, decreases while the validation error, represented by the red line, either remains stagnant or increases, it suggests a potential overfitting problem. In such case, the training process can be halted early using early stopping technique.

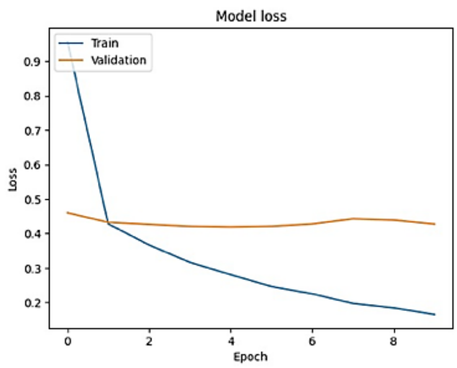


Figure 1. Sample of learning curve with overfitting problem

On the other hand, few-shot learning aims to recognize new classes with very few labeled examples, which is particularly useful when traditional techniques have limited datasets [21]-[24]. By using Vision Transformer to encode image patches, this approach captured more detailed and diverse features from limited data [25]. This model has shown a better performance after applying a combination of feature extraction and few-shot learning techniques.

3. METHOD

The method of this study begins by investigating the impact of combining the annotated dataset with or without augmentation. As depicted in Table 1, four experiments are carried out with Xception and InceptionV3 as classifiers. The first experiment involves using annotated dataset applied with and without augmentation to determine whether augmentation helps in solving the overfitting problem. The second experiment focuses on fine-tuning the model with various hyperparameters, which is intended to mitigate overfitting problem. This includes adjusting parameters like dropout and freezing layers to optimize the classification performance while minimizing overfitting problem. Hence, the optimal configuration that balances model complexity with generalization capability can be achieved.

Table 1. Experiment details

Experiment	Model development	Objectives
1	Annotation with or without augmentation	Solve overfitting
2	Hyperparameter fine-tuning	Solve overfitting
3	Feature extraction+fine-tuning	Enhance performance and prevent overfitting
4	Few-shot learning+fine-tuning	Enhance performance and prevent overfitting

Next, the third experiment explores the impact of combining feature extraction techniques from multiple pre-trained models. By combining these features, the model increases the capability at understanding complex patterns of features and improves the classification performance. However, the experiment also scrutinizes whether overfitting recurs despite the application of dropout and layer freezing techniques. This experiment seeks to determine if the integration of feature extraction will effectively enhance the model performance without compromising its ability to generalize to unseen data. The final experiment investigates the application of few-shot learning technique. This approach is evaluated to understand its effect on model performance and training efficiency. Few-shot learning aims to reduce the dependency on large amounts of training data and computational resources while potentially maintaining or improving accuracy. The integration of few-shot learning should be able to prevent overfitting problem while reducing the overall training time and computational cost.

3.1. Data preprocessing

The dataset Stanford Dogs is obtained from ImageNet, which includes 120 breeds of dog [25]. In data preprocessing, the dataset is divided into three partitions: 80% for training, 10% for testing, and 10% for validating the model’s performance, ensuring a diverse set of examples for learning and validation. To improve the quality of the training dataset, annotation technique is used. Subsequently, the preprocessed images undergo data augmentation techniques to enhance the diversity of the training dataset. The augmentation procedure includes horizontal flipping, where images are mirrored horizontally, aiding the model in learning features from various orientations.

3.2. Classification

The performance of InceptionV3 and Xception models are compared for image classification. Table 2 shows the parameter configuration for InceptionV3. Beginning with the base model in Figure 2, it outputs 7×7 spatial size and 2,048 channels. A global average pooling 2D layer reduces the dimensions, followed by a dense layer with 120 neurons as the classifier. Total parameters are 22,048,664, mainly from the non-trainable InceptionV3 which is displayed in Table 2. Additionally, the model utilizes the Adam optimizer, which typically employs a learning rate of 0.001. The architecture emphasizes an efficient transfer learning with specific task adaptation in the dense layer.

Table 3 and Figure 3 show the parameters configuration of Xception by applying global average pooling to condense its output to a 2,048-dimensional vector, preserving key features. A dense layer with 120 neurons and softmax activation then convert this vector into class probabilities for 120 target classes. This addition of a dense layer introduces 245,880 trainable parameters, enhancing the model’s classification capabilities while leveraging Xception’s strong feature extraction. Additionally, the model utilizes the Adam optimizer, which typically employs a learning rate of 0.001.

Table 2. Parameter list of inceptionV3 model

Layer (type)	Output shape	Parameter
Inception_v3 (Functional)	(None, 5, 5, 2,048)	21,802,784
Global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 2,048)	0
Dense_1 (Dense)	(None, 120)	245,880
Total parameter: 22,048,664		
Trainable parameters: 245,880		
Non-trainable parameters: 21,802,784		

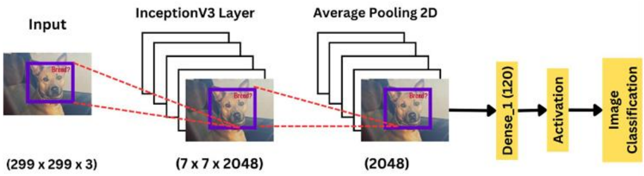


Figure 2. InceptionV3 architecture

Table 3. Parameter list of xception model

Layer (type)	Output shape	Parameter
Xception (Functional)	(None, 10, 10, 2048)	20861480
Global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0
Dense_1 (Dense)	(None, 120)	245880
Total parameter: 21107360		
Trainable parameters: 245880		
Non-trainable parameters: 20861480		

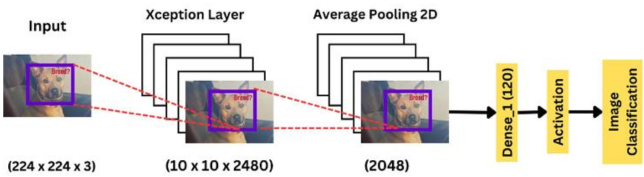


Figure 3. Xception architecture

3.3. Feature extraction

Then, the combination of feature extraction from various architectures like Xception, InceptionV3, VGG16, ResNet50, and MobileNetV2 with pre-trained models underscores a strategy focused on leveraging deep convolutional layers. This integration utilizes the strengths of architectures such as Xception and InceptionV3 to extract diverse visual features from input data. By incorporating pre-trained models with learned weights and parameters from extensive training on large-scale datasets, the methodology effectively initializes the model for specific tasks or domains. This initialization minimizes the requirement for additional training, enabling swift deployment in various computer vision applications across diverse datasets and tasks. After combining feature extraction from these architectures, the model undergoes training with classifiers specifically to Xception and InceptionV3.

3.4. Few-shot learning

On the other hand, the few-shot learning model employing InceptionV3 and Xception architectures with Prototypical Networks adopts a distinct strategy to address biases and overfitting. It begins by creating a custom dataset tailored for few-shot task, which is then divided into training, validation, and testing subsets. The model undergoes optimization using stochastic gradient descent (SGD) along with a MultiStepLR scheduler, and its performance is assessed on both validation and testing datasets. Functions for training epochs and evaluating tasks are implemented to ensure effective adaptation to new classes, with the model training fixed at 50 epochs.

In few-shot learning, the training and testing sets consist of entirely different classes, ensuring no overlapping samples. The training process involves selecting a query set and a support set from the training data. In a typical setup like 5-shot 1-way, the support set includes five different classes, which are randomly sampling from the training set. The query set is similarly constructed with five images, each from one of the five classes chosen in the support set. During training, the model analyzes an image from the query set and compares it with images in the support set to determine the closest matching class between the support set and query image. This comparison helps the model learn to make predictions based on minimal samples. Pre-trained on a large and diverse dataset, such as ImageNet, followed by this fine-tuning process on the specific few-shot task, allows the model to generalize well to new classes with only a few samples.

4. RESULTS AND DISCUSSION

In the first experiment, annotated dataset is combined with and without augmentation images to assess the effect of augmentation to mitigate overfitting problem. Table 4 presents the results, showing that performance of both classifiers applied on the annotated and augmentation datasets achieved a slightly better accuracy result, which are 90% and 84 % for Xception and InceptionV3 respectively. It is worth noting that the value of distance between training loss and validation loss of both models has reduced as compared to the values of distance for annotated dataset without augmentation. Nevertheless, even though the performance increases for accuracy, instances of overfitting persisted. Augmentation only led to a slight improvement of accuracy result as the diversity of dataset has expanded. Although the accuracy performance is the highest when using annotated dataset with augmentation, overfitting problem remains evident as there is a significant gap between the training loss and validation loss as illustrated in Figures 4(a) and (b).

Table 4. Comparative performance of both models for different types of datasets

Classifier	Augmentation	Accuracy (%)	Distance between loss
Xception	No	87	0.3194
	Yes	90	0.2478
InceptionV3	No	81	0.7824
	Yes	84	0.5244

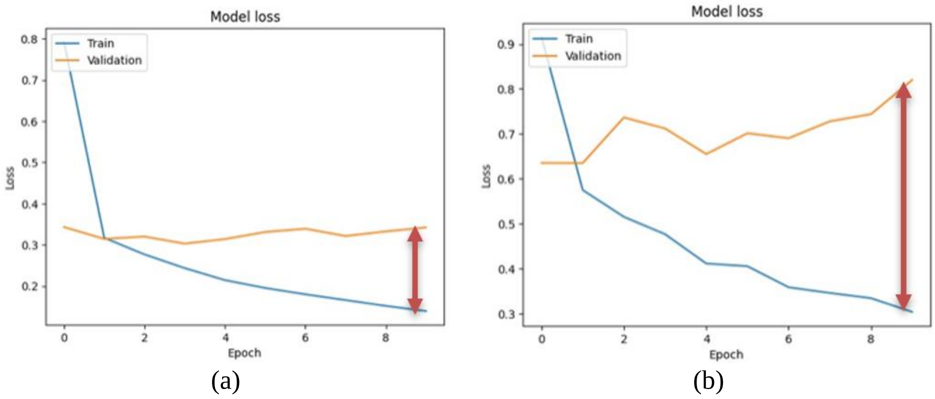
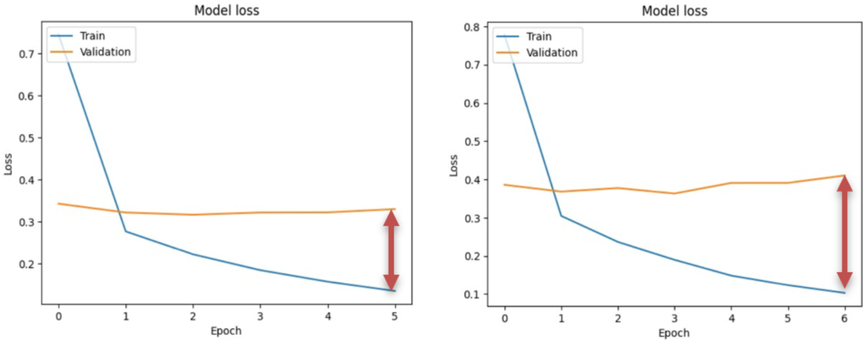


Figure 4. The risk of overfitting problem for annotated dataset with augmentation; (a) Xception and (b) InceptionV3

4.1. Hyperparameter fine-tuning

To overcome the overfitting problem, the model incorporates early stopping, dropout layers, and layer freezing. All training models are trained using only 10 epochs. In such cases, the model is unlikely to have sufficient time to overfit excessively within this short training period. The primary purpose of early stopping is to identify the point at which further training does not lead to better validation performance, but with only 10 epochs, the model's performance will not have enough iterations to drastically worsen following an initial improvement.

Figure 5 shows that the overfitting problem maintains in both Xception and InceptionV3 models after early stopping. In Figure 5(a) the Xception model stopped at epoch 6 with accuracy 88% while in Figure 5(b) the InceptionV3 model stopped at epoch 7 with accuracy 90% which is notably impressive. This indicates that while the model fits the training data well, early stopping does not perform as well on the validation set, which is a hallmark of overfitting. Therefore, the benefits of early stopping are minimal in this scenario because the training is completed quickly, and the risk of overfitting is inherently lower due to the limited number of training epochs. While early stopping did not significantly reduce overfitting problem but shows a slight improvement in the performance of both models. So early stopping technique still not considered solving the overfitting problem.



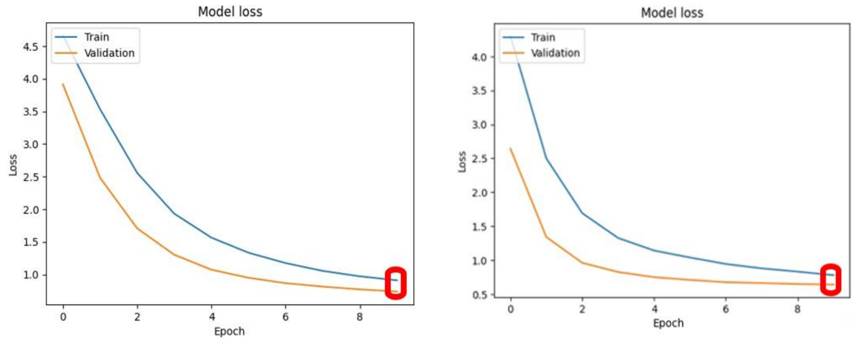
(a) (b)

Figure 5. Model loss after early stopping in hyperparameter tuning; (a) Xception (epoch 6) and (b) InceptionV3 (epoch 7)

Based on Table 5, Xception and InceptionV3 models achieved the accuracy of 88% and 87% respectively, by utilizing a 50% dropout rate and L2 weight regularization with an optimal approach involving freezing 50 layers to achieve the best performance. Figures 6(a) and (b) shows the results after combining dropout and layer freezing, demonstrating that this combination effectively balances the regularization of neural network. The gap between training loss and validation loss has significantly reduced compared to the model loss graph from Figures 5(a) and (b), thereby solving the overfitting problem. Therefore, by leveraging dropout and freezing layers strategically, the model retains robust features while adapting effectively to new data, enhancing overall performance and generalization of the neural network. The distance between loss values in Table 5 is also smaller compared to Table 4. The Xception model only suffers a distance between loss of 0.1409 compared to 0.2478 that not yet applied hyperparameter fine-tuning. Similarly, the InceptionV3 model shows a lower distance between loss of 0.1213 compared to 0.5244 from Table 4.

Table 5. Summary of fine tunning results

Fine tune	Classifier	Accuracy (%)	Training time	Distance between loss
Dropout=0.5	Xception	88	412 m 2.1 s	0.1409
Freeze layer=50	InceptionV3	87	530 m 9.4 s	0.1213



(a) (b)

Figure 6. Model loss for augmented dataset after fine tuning; (a) Xception and (b) InceptionV3

4.2. Feature extraction

To ensure good performance after addressing overfitting, different combinations of feature extraction techniques are investigated in the third experiment. This involves using a common classifier, Xception and InceptionV3, to classify images based on features extracted from various pre-trained models. Additionally, all classifier models undergo the optimal dropout and layer freezing from the second experiment, to prevent overfitting problem. Table 6 shows the results of various combinations of feature extraction techniques. Comparing these results to Table 5, the integration of various feature extraction techniques reveals a significant deterioration in the overall performance in terms of accuracy and training time.

Table 6. Performance of different combination of feature extraction result

Feature extraction	Classifier	Accuracy (%)	Training time	Distance between loss
VGG16+	Xception	74	419 m 56.8 s	0.9660
Resnet50+	InceptionV3	63	468 m 20.7 s	1.0430
MobileNetV2				
Xception+	Xception	79	369 m 41.5 s	0.6749
Resnet50+	InceptionV3	78	1,895 m 8.2 s	0.7014
MobileNetV2				
Resnet50+	Xception	76	582 m 49.1 s	0.8750
MobileNetV2	InceptionV3	64	403 m 54.9 s	0.8250
MobileNetV2	Xception	78	589 m 9.4 s	0.8940
	InceptionV3	67	492 m 40.1 s	0.7170

The results from different combinations of feature extraction revealed several unexpected performance issues. Firstly, the accuracy of the all model with feature extraction is lower than previous experiment results of Xception and InceptionV3, which has minimized the overfitting problem. Despite implementing techniques like dropout and layer freezing, overfitting persists and the average time usage for all the combinations of feature extraction. These issues may be attributed to the complexity and redundancy introduced by combining multiple feature extraction techniques, which lead to an increased computational burden and noise in the extracted features. The expansive nature of a high-dimensional feature space can indeed contribute to overfitting, wherein the model is prone to fitting noise or irrelevant patterns, rather than capturing the essential relationships in the data. Furthermore, the inclusion of diverse features from various models might lead to inconsistencies and conflicts, reducing the overall performance of the model.

4.3. Few-shot learning

After that, few-shot learning is employed to assess its impact on the Xception and InceptionV3 models, specifically to improve the performance of classifiers and prevent the overfitting problem. Comparing to the results in Table 7, it can be observed that the performance of the few-shot learning model is significantly better than the third experiment. The overfitting problem remains resolved with few-shot learning, whereas the overfitting recurs when integrating with feature extraction techniques.

However, the results of few-shot learning in Table 7 reveal a slightly lower performance, which is 81% for Xception and 79% for InceptionV3 compared to Table 5. This difference can be attributed to few-shot learning's reliance on a smaller amount of training data, leading to a potential trade-off between model complexity and generalization of the neural network. Despite this limitation, few-shot learning remains valuable, particularly in resource-constrained scenarios. Although the accuracy of the few-shot learning model drops by 7% for Xception and 9% for InceptionV3, the training time has reduced by approximately 30% for both classifiers.

Table 7. Performance of few-shot learning with both classifiers

Feature extraction	Classifier	Accuracy (%)	Training time	Distance between loss
Few-shot	Xception	81	283 m 5.7 s	0.1373
	InceptionV3	79	300 m 6.7 s	0.1275

5. CONCLUSION

In conclusion, this study highlights the effectiveness of employing techniques such as data augmentation, dropout layers, and layer freezing in mitigating overfitting and enhancing the performance of deep learning-based classification for dog species. With this, the objective of solving overfitting is achieved by using data augmentation to introduce variability into the training dataset, thereby enhancing the model's ability to generalize to unseen data. Following augmentation, fine-tuning techniques such as dropout and freezing layers are applied. This dual strategy ensures the model learns robust features without memorizing noise or irrelevant details, striking a balance between complexity and generalization in deep learning. The use of Xception as the classifier demonstrated the best and more consistent performance compared to InceptionV3, likely due to its architectural advantage in handling spatial and channel-wise dependencies. Based on the experimental results involving dropout and freeze layers, the combination of feature extraction techniques did not lead to an improvement in performance. While few-shot learning offers a promising approach for the scenario with limited data, its effectiveness perhaps falls behind the big and augmented datasets. However, it requires less time for training, thereby reducing computational cost, and effectively mitigates overfitting problem.

FUNDING INFORMATION

This work was supported by the Telekom Malaysia Research and Development under Grant RDTC/241125 (MMUE/240066).

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Jing-Yee Ong	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓			
Lee-Yeng Ong					✓	✓	✓	✓		✓	✓	✓	✓	
Meng-Chew Leow		✓			✓		✓			✓				✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are openly available in [Stanford Dogs Dataset] at https://www.kaggle.com/datasets/jessicali9530/stanford-dogs-dataset, reference number [25].

REFERENCES

[1] A. Varshney, A. Katiyar, A. K. Singh, and S. S. Chauhan, "Dog Breed Classification Using Deep Learning," in *2021 International Conference on Intelligent Technologies, CONIT 2021*, Institute of Electrical and Electronics Engineers Inc., Jun. 2021. doi: 10.1109/CONIT51480.2021.9498338.

[2] P. Borwarnginn, W. Kusakunniran, S. Karnjanapreechakorn, and K. Thongkanchorn, "Knowing Your Dog Breed: Identifying a Dog Breed with Deep Learning," *International Journal of Automation and Computing*, vol. 18, no. 1, pp. 45–54, Feb. 2021, doi: 10.1007/s11633-020-1261-0.

[3] K. Morrill *et al.*, "Ancestry-inclusive dog genomics challenges popular breed stereotypes," *Science*, vol. 376, no. 6592, Apr. 2022, doi: 10.1126/science.abk0639.

[4] R. Poojary and A. Pai, "Comparative Study of Model Optimization Techniques in Fine-Tuned CNN Models," in *2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, IEEE, Nov. 2019, pp. 1–4, doi: 10.1109/ICECTA48151.2019.8959681.

[5] A. Vabalas, E. Gowen, E. Poliakoff, and A. J. Casson, "Machine learning algorithm validation with a limited sample size," *PLoS One*, vol. 14, no. 11, Nov. 2019, doi: 10.1371/journal.pone.0224365.

[6] M. Ahmad, M. Abdullah, H. Moon, and D. Han, "Plant Disease Detection in Imbalanced Datasets Using Efficient Convolutional Neural Networks with Stepwise Transfer Learning," *IEEE Access*, vol. 9, pp. 140565–140580, 2021, doi: 10.1109/ACCESS.2021.3119655.

[7] P. Borwarnginn, K. Thongkanchorn, S. Kanchanapreechakorn, and W. Kusakunniran, "Breakthrough Conventional Based Approach for Dog Breed Classification Using CNN with Transfer Learning," in *2019 11th International Conference on Information Technology and Electrical Engineering (ICITEE)*, IEEE, Oct. 2019, pp. 1–5, doi: 10.1109/ICITEED.2019.8929955.

[8] J. Orozco, V. Manian, E. Alfaro, H. Walia, and B. K. Dhatt, "Graph Convolutional Network Using Adaptive Neighborhood Laplacian Matrix for Hyperspectral Images with Application to Rice Seed Image Classification," *Sensors*, vol. 23, no. 7, Apr. 2023, doi: 10.3390/s23073515.

[9] R. Kumar, M. Sharma, K. Dhawale, and G. Singal, "Identification of Dog Breeds Using Deep Learning," in *2019 IEEE 9th International Conference on Advanced Computing (IACC)*, IEEE, Dec. 2019, pp. 193–198, doi: 10.1109/IACC48062.2019.8971604.

[10] C. Zhao, C. Li, J. Li, and F. Chen, "Fair meta-learning for few-shot classification," in *Proceedings - 11th IEEE International Conference on Knowledge Graph, ICKG 2020*, Institute of Electrical and Electronics Engineers Inc., Aug. 2020, pp. 275–282, doi: 10.1109/ICBK50248.2020.00047.

[11] B. Valarmathi, N. S. Gupta, G. Prakash, R. H. Reddy, S. Saravanan, and P. Shanmugasundaram, "Hybrid Deep Learning Algorithms for Dog Breed Identification—A Comparative Analysis," *IEEE Access*, vol. 11, pp. 77228–77239, Oct. 2023, doi: 10.1109/ACCESS.2023.3297440.

[12] S. H. Wang and Y. Chen, "Fruit category classification via an eight-layer convolutional neural network with parametric rectified linear unit and dropout technique," *Multimedia Tools and Applications*, vol. 79, no. 21–22, pp. 15117–15133, Jun. 2020, doi: 10.1007/s11042-018-6661-6.

[13] H. Moayed and E. G. Mansoori, “Skipout: An Adaptive Layer-Level Regularization Framework for Deep Neural Networks,” *IEEE Access*, vol. 10, pp. 62391–62401, May 2022, doi: 10.1109/ACCESS.2022.3178091.
[14] B. Zhao, C. Cheng, Z. Peng, Q. He, and G. Meng, “Hybrid Pre-Training Strategy for Deep Denoising Neural Networks and Its Application in Machine Fault Diagnosis,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, 2021, doi: 10.1109/TIM.2021.3126019.
[15] I. Castiglioni *et al.*, “AI applications to medical images: From machine learning to deep learning,” *Physica Medica*, vol. 83. Associazione Italiana di Fisica Medica, pp. 9–24, Mar. 2021, doi: 10.1016/j.ejmp.2021.02.006.
[16] K. Goutam, S. Balasubramanian, D. Gera, and R. R. Sarma, “LayerOut: Freezing Layers in Deep Neural Networks,” *SN Comput Sci*, vol. 1, no. 5, Sep. 2020, doi: 10.1007/s42979-020-00312-x.
[17] P. P. Mitra, “Understanding overfitting peaks in generalization error: Analytical risk curves for l_2 and l_1 penalized interpolation,” Jun. 2019.
[18] S. Sanchez-Martinez *et al.*, “Machine Learning for Clinical Decision-Making: Challenges and Opportunities in Cardiovascular Imaging,” *Frontiers in Cardiovascular Medicine*, vol. 8, 2021, doi: 10.3389/fcvm.2021.765693.
[19] X. Ying, “An Overview of Overfitting and its Solutions,” *J Phys Conf Ser*, vol. 1168, p. 022022, Feb. 2019, doi: 10.1088/1742-6596/1168/2/022022.
[20] Galgotias University, Universitatea “Aurel Vlaicu” din Arad, IEEE Industry Applications Society, and Institute of Electrical and Electronics Engineers, *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA): Galgotias University, Greater Noida, UP, India*, 2020, doi: 10.1109/ICCCA49541.2020.
[21] X. Sun, B. Wang, Z. Wang, H. Li, H. Li, and K. Fu, “Research Progress on Few-Shot Learning for Remote Sensing Image Interpretation,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 2387–2402, 2021, doi: 10.1109/JSTARS.2021.3052869.
[22] D. Wertheimer and B. Hariharan, “Few-Shot Learning with Localization in Realistic Settings,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2019, pp. 6551–6560, doi: 10.1109/CVPR.2019.00672.
[23] M. Yan, “Adaptive Learning Knowledge Networks for Few-Shot Learning,” *IEEE Access*, vol. 7, pp. 119041–119051, 2019, doi: 10.1109/ACCESS.2019.2934694.
[24] A. Fritzler, V. Logacheva, and M. Kretov, “Few-shot classification in named entity recognition task,” in *Proceedings of the ACM Symposium on Applied Computing*, Association for Computing Machinery, 2019, pp. 993–1000, doi: 10.1145/3297280.3297378.
[25] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li, "Novel dataset for fine-grained image categorization: Stanford dogs", Proc. CVPR workshop on fine-grained visual categorization (FGVC), vol. 2, 2011.

BIOGRAPHIES OF AUTHORS

Jing-Yee Ong currently studying Bachelor of Computer Science (Hons.) Artificial Intelligence in Multimedia Unversity, Melaka, Malaysia. Her area of interest includes machine learning, deep learning, and sustainable AI. She can be contacted at email: ojy0282@gmail.com.

Lee-Yeng Ong received Ph.D. degree in computer vision from Multimedia University, Malaysia. She is currently working as an assistant professor with Faculty of Information Science and Technology, Multimedia University, Malaysia. Her research interests include object tracking, computer vision, data science, and big data analytics. She can be contacted at email: lyong@mmu.edu.my.

Meng-Chew Leow received his Doctor of Philosophy from Multimedia University. His research interest is in game-based learning, specifically in role-playing game-based learning. He is also interested in system science, practical spirituality, and philosophy. He can be contacted at email: mcleow@mmu.edu.my.