Journal of Engineering, Management and Information Technology

Vol. 04, No. 02 (2026) 205-212, doi: 10.61552/JEMIT.2026.02.009 - http://jemit.aspur.rs

EVALUATION OF TRANSFER LEARNING TECHNIQUE ON COMPUTER PART IMAGES CLASSIFICATION

Nam Tran Quy [1]

Received 05.10.2025.
Revised 28.11.2025.
Accepted 19.12.2025.

Keywords:

Transfer Learning, Image Classification, PC Hardware, EfficientNetB7, ResNet50, Xception, Deep Learning, Computer Vision.

Original research

A B S T R A C T

This study makes an evaluation of the transfer learning technique applied to the classification of computer part images using three state-of-the-art convolutional neural network architectures: EfficientNetB7, ResNet50, and Xception. The study utilizes a dataset of 3,279 images with 14 distinct hardware components. Each pre-trained model was fine-tuned and evaluated under consistent experimental conditions over 10 epochs with an input resolution of 256x256 pixels. The results demonstrate the strong effectiveness of transfer learning in this specialized domain, with all models achieving high classification accuracy. EfficientNetB7 attained the highest performance at 78%, while ResNet50 reached 73%, and Xception reached 72%. This results show the most suitable model, namely EfficientNetB7 to optimize accuracy and efficiency in this problem. Meanwhile, Xception, with its depthwise separable convolutions, and ResNet50, with its residual learning framework, also delivered robust and competitive results, each showcasing distinct advantages in terms of feature extraction capability and architectural efficiency.

© 2026 Journal of Engineering, Management and Information Technolog

1. INTRODUCTION

The application of deep learning and transfer learning for computer part and electronic component image classification has been an emerging topic, driven by demands in e-commerce, recycling, inventory management, and technical support automation (Zhang, 2021). The creation of high-quality, accurately labeled image datasets for computer components presents a practical challenge, often hindering the development of specialized classification models. This study utilizes the PC Parts Images Dataset (Asaniczka, 2023), a publicly available collection of 3,279 images across 14 distinct hardware categories (e.g., CPU, GPU, motherboard, RAM…), with each image standardized to a resolution of 256x256 pixels. This dataset provides a focused benchmark for evaluating fine-grained classification within the domain of computer hardware. However, with a limited total number of images and significant intra-class variance such as different brands, models, angles, it represents a non-trivial test for transfer learning techniques. The performance of advanced architectures like EfficientNetB7, ResNet50, and Xception on such a specialized, mid-scale dataset has not been extensively documented. This paper aims to fill that gap by conducting a systematic empirical evaluation, providing clear metrics on accuracy, efficiency, and convergence to guide practical implementation.

In this paper, we implement transfer learning with pre-trained models to evaluate and compare the performance of three advanced convolutional neural network architectures on a computer parts image classification task. Specifically, the study implements three pre-trained CNN models namely ResNet50, Xception, and EfficientNetB7, leveraging the PC Parts Images Dataset. The subsequent sections describe details of the methodology of our experimental setup, present a

[1] Corresponding author: Nam Tran Quy
Email: namtq@dainam.edu.vn

comparative analysis of the models' performance in terms of accuracy and computational efficiency, and discuss the implications of our findings for the fine-grained classification of hardware components.

2. LITERATURE REVIEW

In this part, the study reviews some other similar works which were using machine learning to identify the PC Parts Images. The advent of deep learning, particularly Convolutional Neural Networks (CNNs), has fundamentally transformed the field of computer vision (Yoo, 2015). In this section, we focus on reviewing some several recent studies that utilize machine learning and specifically transfer learning techniques to classify images of hardware components, with a focus on works that involve pre-trained models comparable to EfficientNetB7, ResNet50, and Xception.

Several researchers have leveraged transfer learning to overcome limitations posed by small, domain-specific datasets. Ravikumar et al. (2023) demonstrated the use of ResNet50 and VGG16 for classifying computer motherboard components, achieving high accuracy. Their work highlighted how pre-trained models could effectively recognize fine-grained features such as capacitors, slots, and chipsets without extensive data collection. This aligns with the present study’s use of ResNet50 for PC hardware recognition. In the domain of electronic waste (e-waste) sorting, Chen et al. (2022) employed EfficientNetB3 and MobileNetV2 to automatically classify images of printed circuit boards (PCBs) and various internal components. Their system achieved good accuracy, underscoring EfficientNet’s scalability and efficiency in resource-constrained industrial applications. Although their focus was on recycling automation, their methodology supports the relevance of EfficientNet variants for component-level classification.

Furthermore, Ge et al. (2016) explored the use of Xception and InceptionV3 for fine-grained classification of consumer electronics, including laptops, keyboards, and internal hardware. They reported that Xception outperformed InceptionV3 due to its depthwise separable convolution design, which efficiently captures spatial hierarchies, an architectural advantage also examined in our work with Xception. Similarly, Baobaid et al. (2022) conducted a comparative study of several CNN architectures including ResNet50, DenseNet121, and EfficientNetB0 for classifying server and networking hardware components. They found EfficientNet variants consistently delivered superior accuracy with fewer parameters, reinforcing the model's suitability for hardware recognition tasks and motivating our inclusion of EfficientNetB7.

Another relevant work by Abulfaraj and Binzagr, (2025) applied ensemble methods combining ResNet50, Xception, and Vision Transformers (ViTs) for multi-label classification of computer assembly parts. Their results indicated that hybrid CNN architectures significantly improve classification robustness, especially with occluded or partially visible components. This supports the rationale for evaluating diverse architectures like ResNet50 and Xception in our study. In addition, Thirumurthy et al. (2024) benchmarked several transfer learning models including VGG19, ResNet50, and EfficientNetB4 for GPU and CPU identification from retail images. Their work emphasized dataset augmentation and hyperparameter tuning to address inter-class similarity, a challenge also present in our PC parts dataset.

Collectively, these studies confirm that transfer learning with advanced CNNs is a proven and effective approach for hardware image classification (Shin et al., 2016). However, a direct comparative evaluation of EfficientNetB7, ResNet50, and Xception on an unified dataset of diverse PC components remains underexplored. Our research fills this gap by providing a systematic performance comparison, practical accuracy benchmarks, and deployment-oriented insights for real-world PC part recognition systems.

3. METHODOLOGY

This study employed a comparative transfer learning approach to evaluate the performance of three advanced Convolutional Neural Network (CNN) architectures for classifying images of computer hardware components. The methodology encompasses dataset preparation, model selection and adaptation, and the experimental training protocol.

3.1. Model Architectures and Transfer Learning Framework

This study implements and evaluates three state-of-the-art convolutional neural network architectures for the task of PC hardware component classification: EfficientNetB7, ResNet50, and Xception. Each model represents a distinct paradigm in deep learning design, compound scaling, residual learning, and depthwise separable convolutions, respectively. The following sections detail their architectural innovations, learning mechanisms, and the transfer learning strategy employed to adapt them to the domain-specific classification task.

EfficientNetB7: EfficientNetB7 belongs to the EfficientNet family, which introduced a novel compound scaling method that systematically scales network depth, width, and resolution using a set of empirically derived coefficients. Unlike conventional scaling approaches that arbitrarily increase one dimension, compound scaling optimizes all three dimensions simultaneously, leading to better accuracy-efficiency trade-offs. EfficientNetB7 is the largest variant in this family, designed to push the performance ceiling while maintaining computational feasibility (Ravikumar et al., 2023).

At its core, EfficientNetB7 employs Mobile Inverted Bottleneck Convolutions (MBConv), which integrate depthwise separable convolutions, squeeze-and-excitation attention blocks, and stochastic depth regularization. This design reduces parameter count and computational cost while enhancing feature representational capacity. The model is pre-trained on the ImageNet dataset, which enables it to capture a rich hierarchy of visual patterns, from edges and textures to complex object structures.

In this study, the pre-trained convolutional base of EfficientNetB7 was kept frozen to preserve its learned generic features. A custom classification head was appended, consisting of: (1) A global average pooling layer to reduce spatial dimensions; (2) Two fully connected dense layers (256 units each) with ReLU activation; (3) A final softmax output layer with 14 units corresponding to the PC part categories. Only the newly added layers were trained, allowing the model to specialize in hardware-related visual cues without catastrophic forgetting of general-purpose features.

ResNet50: ResNet50 is a milestone architecture that popularized deep residual learning to address the vanishing gradient problem in very deep networks. Its design is based on residual blocks, each containing a shortcut connection (skip connection) that adds the input of the block to its output. This identity mapping enables gradients to flow directly through the network, facilitating the training of architectures with 50 or more layers (He et al., 2016).

The residual blocks in ResNet50 are organized into four stages, with each stage comprising multiple bottleneck blocks that use 1×1 convolutions for dimensionality reduction and expansion, followed by 3×3 convolutions for spatial feature extraction. This design balances depth and computational efficiency, making ResNet50 a robust and widely adopted backbone for visual recognition tasks. For transfer learning, the original ImageNet classification head (a 1000-unit fully connected layer) was removed. In its place, a task-specific classifier was constructed, including: (1) A global average pooling layer; (2) Two dense layers (256 units each, ReLU activation); (3) A softmax layer with 14 output neurons.

During training, all convolutional layers remained frozen, while only the new classification layers were updated. This approach leverages ResNet50's powerful hierarchical feature extraction capabilities while adapting its top layers to the nuances of hardware component imagery.

Xception: Xception stands for "Extreme Inception" and represents an evolution of the Inception architecture. It replaces the standard Inception modules with depthwise separable convolutions, which factorize convolutional operations into two steps: depthwise spatial convolution followed by pointwise (1×1) convolution. This factorization significantly reduces parameter count and computational complexity while maintaining, or even improving, representational power (Chollet, 2017).

The Xception architecture consists of a series of depthwise separable convolutional blocks, each followed by batch normalization and ReLU activation. It also includes residual connections between some blocks to stabilize training and improve gradient flow. Pre-trained on ImageNet, Xception excels at capturing multi-scale features with high parameter efficiency, making it suitable for tasks where computational resources are constrained. In this work, the base convolutional layers of Xception were used as a fixed feature extractor. A custom classifier was attached, comprising: (1) Global average pooling; (2) Two dense ReLU layers (256 units each); (3) A softmax output layer for 14-class prediction. By keeping the base model frozen, we ensure that the rich, general-purpose visual features learned from ImageNet are retained, while the new layers are optimized to distinguish between fine-grained hardware categories.

3.2. Dataset and Preprocessing

The experimental dataset comprises 3,279 images systematically categorized into 14 distinct classes of PC components, including motherboard, GPU, RAM, CPU, etc. The dataset is named as "PC Parts Images Dataset" which was publicly available on Kaggle and published by Asaniczka (2023), which comprises a curated collection of images representing 14 distinct categories of computer hardware components.

This dataset is specifically designed for image classification tasks in the domain of PC hardware recognition, making it highly suitable for evaluating the performance of deep learning models in real-world technical applications. The 14 categories include: Cables, Case, CPU, GPU, HDD, Headset, Keyboard, Microphone, Monitor, Motherboard, Mouse, RAM, Speakers, Webcam.

The class distribution is visualized in Figure 1, which illustrates the number of samples per category. As shown, the dataset exhibits a moderate degree of class imbalance, with Speakers having the highest number of samples (approximately 70 images), while categories such as Mouse and CPU are relatively underrepresented (around 28–29 images). This imbalance reflects real-world availability and visual diversity of certain components, which may influence model learning and generalization.

To ensure consistency and compatibility with the pre-trained models, all images were resized to a uniform dimension of 256×256 pixels using bilinear interpolation. This resolution was selected as a trade-off between computational efficiency and preserving sufficient visual detail for component discrimination. Pixel values were normalized to the range [0, 1] by dividing by 255. Additionally, dataset-specific preprocessing functions corresponding to each pre-trained model (EfficientNetB7, ResNet50, Xception) were applied to align input data with the feature space expected by each network.

3.3. Experimental Setup and Training Protocol

The dataset was randomly partitioned into training (80%), validation (20% of training), and testing (20%) sets using a fixed random seed (random_state=2) to ensure reproducibility.

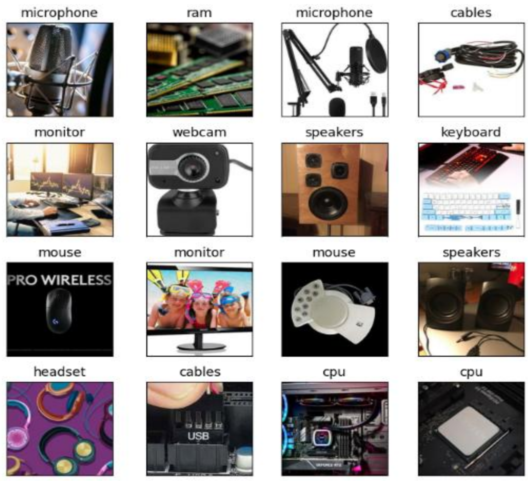

Figure 1. Example of PC part images dataset

The resulting splits contained: Training samples with 2,099 images, validation samples with 524 images, test samples with 656 images.

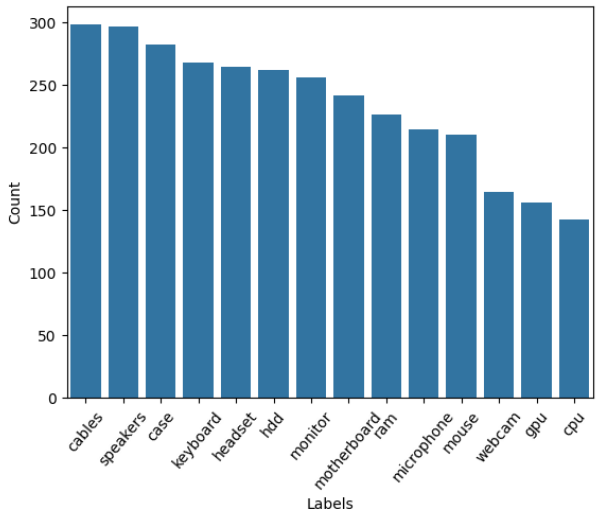

Figure 2. Distribution of Dataset by Labels

To mitigate overfitting and enhance model robustness, a comprehensive set of augmentation techniques was applied exclusively to the training set using the ImageDataGenerator class from Keras. The augmentations included: Random rotation (±30°); Zoom range (15%); Width and height shift (20%); Shear transformation (15%); Horizontal flipping; Fill mode: "nearest". These transformations simulate variations in viewpoint, scale, and orientation that may occur in real-world imaging conditions, thereby improving model generalization.

The preprocessed dataset served as the foundation for training and evaluating the three CNN architectures, enabling a fair and reproducible comparison of their classification capabilities (Figure 2).

The experimental setup was designed to evaluate and compare the performance of three pre-trained convolutional neural networks, EfficientNetB7, ResNet50, and Xception, on the task of PC component image classification. The dataset, consisting of 14 categories of PC parts, was split into training (80%) and testing (20%) sets using train_test_split from scikit-learn with random_state=2 to ensure reproducibility. The training set was further divided into training and validation subsets (80/20) via the validation_split parameter in Keras' ImageDataGenerator.

Data augmentation was applied during training to improve generalization, including random rotation (30°), zoom (15%), width and height shifts (20%), shear (15%), and horizontal flipping. All images were resized to 256×256 pixels. Each model was initialized with ImageNet weights, with its top classification layer removed. A custom classifier head was added, comprising two fully connected layers (256 units each, ReLU activation) and a softmax output layer with 14 units. All base model layers were frozen during training. Models were compiled using the Adam optimizer and categorical cross-entropy loss. Training was conducted for a maximum of 10 epochs with a batch size of 32. Early stopping was implemented with a patience of 5 epochs based on validation loss. Evaluation was performed on the held-out test set using accuracy, loss, precision, recall, F1-score, and confusion matrices. The following hyperparameters were kept consistent across all models Table 1.

Table 1. Same hyper-parameters settings

Types	**Values**
Input size	256×256×3
Batch size	32
Optimizer	Adam (default learning rate)
Loss function	Categorical cross-entropy
Early stopping patience	5 epochs
Maximum epochs	10 epochs
Train/Test split ratio	80%:20%
Train/Validation split ratio	80%:20% (from training set)
Data augmentation	Enabled (rotation, zoom, shift, flip)
Learning rate	Default (0.001)
Base model freezing	Enabled (transfer learning)

The uniform setup acrosses all three models, a consistent transfer learning protocol ensures a fair comparison, isolating architectural differences as the primary variable influencing performance. The use of frozen base models also accelerates training and reduces the risk of overfitting, given the moderate size of the PC parts dataset.

4. RESULTS AND DISCUSSION

The experimental results demonstrate that all three pre-trained models, namely EfficientNetB7, ResNet50, and Xception achieved competitive performance in classifying PC components. Among them, EfficientNetB7 outperformed the others with a test accuracy of 77.59%, followed by ResNet50 (72.56%) and Xception (71.80%) (see Table 2). The test loss values followed a similar trend, with EfficientNetB7 achieving the lowest loss (1.0159), indicating better generalization and robustness.

Table 2. Overall Performance of the Models

Model	**Test Accuracy (%)**
EfficientNetB7	77.59
ResNet50	72.56
Xception	71.80

The classification report reveals varying performance across the 14 PC component categories. EfficientNetB7 showed strong and balanced performance across most classes, particularly excelling in categories such as speakers (F1-score: 0.91), mouse (0.87), and keyboard (0.86). However, certain categories like CPU (0.49) and cables (0.71) remained challenging across all models, likely due to visual similarity with other components or limited training samples (see Table 3).

Table 3. EfficientNetB7 Performance on Test Dataset

Class		Precision	Recall	F1-score	Support
0	cables	0.73	0.70	0.71	57
1	case	0.58	0.67	0.62	52
2	cpu	0.60	0.41	0.49	29
3	gpu	0.68	0.62	0.65	40
4	hdd	0.74	0.74	0.74	39
5	headset	0.83	0.83	0.83	53
6	keyboard	0.85	0.87	0.86	54
7	microphone	0.80	0.80	0.80	50
8	monitor	0.80	0.88	0.84	56
9	motherboard	0.72	0.75	0.73	51
10	mouse	0.89	0.86	0.87	28
11	ram	0.83	0.81	0.82	42
12	speakers	0.90	0.91	0.91	70
13	webcam	0.82	0.80	0.81	35
Accuracy				0.78	656
Macro Avg		0.77	0.76	0.76	656
Weighted Avg		0.78	0.78	0.77	656

ResNet50 exhibited high precision in some categories (e.g., speakers: 0.96) but suffered from inconsistent recall, as seen in CPU (recall: 0.34) and microphone (recall: 0.60) (see Table 4). Xception showed a similar pattern, with strong precision in speakers (0.92) and microphone (0.92), but lower recall in CPU (0.28) and cables (0.53) (see Table 5).

Table 4. ResNet50 Performance on Test Dataset

Class		Precision	Recall	F1-score	Support
0	cables	0.47	0.82	0.60	57
1	case	0.56	0.60	0.58	52
2	cpu	0.77	0.34	0.48	29
3	gpu	0.79	0.75	0.77	40
4	hdd	0.67	0.72	0.69	39
5	headset	0.72	0.77	0.75	53
6	keyboard	0.88	0.80	0.83	54
7	microphone	0.94	0.60	0.73	50
8	monitor	0.82	0.84	0.83	56
9	motherboard	0.76	0.63	0.69	51
10	mouse	0.61	0.79	0.69	28
11	ram	0.82	0.74	0.78	42
12	speakers	0.96	0.76	0.85	70
13	webcam	0.74	0.89	0.81	35
Accuracy				0.73	656
Macro Avg		0.75	0.72	0.72	656
Weighted Avg		0.76	0.73	0.73	656

The training curves indicate that all models converged within the first 10 epochs, with EfficientNetB7 showing the steepest increase in training accuracy, reaching over 95% by epoch 8 (Figure 3). However, validation accuracy plateaued earlier, suggesting potential overfitting despite the use of data augmentation and early stopping. This is particularly evident in EfficientNetB7, where the gap between training and validation accuracy widened after epoch 3.

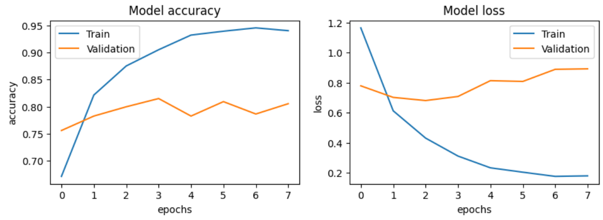

Figure 3. EfficientNetB7 Training Performance

ResNet50 and Xception showed more stable validation performance but slower improvement in training accuracy, indicating a more conservative learning dynamic, possibly due to their architectural depth and complexity (Figure 4).

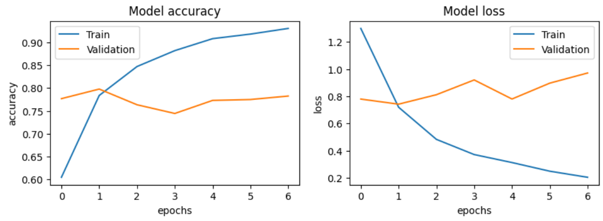

Figure 4. ResNet50 Training Performance

The superior performance of EfficientNetB7 can be attributed to its advanced compound scaling mechanism, which optimizes network depth, width, and resolution simultaneously. This allows the model to capture finer visual details, a critical factor in distinguishing between similar-looking PC parts.

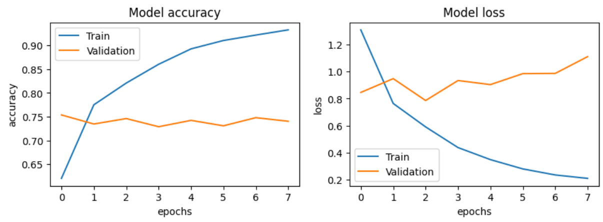

Figure 5. Xception Training Performance

ResNet50, while robust, may suffer from residual block saturation in deeper layers when applied to a domain-specific task with limited data. Xception, with its depthwise separable convolutions, is efficient but may require more data to achieve competitive performance in fine-grained classification (Figure 5). The results also highlight the challenge of class imbalance and intra-class variation in real-world PC component images, which may be addressed in future work through advanced augmentation, attention mechanisms, or metric learning. From a practical standpoint, EfficientNetB7 offers a promising balance between accuracy and computational cost, making it suitable for deployment in automated PC part identification systems, such as in e-commerce, recycling, or inventory management. However, its larger model size and inference time should be considered in resource-constrained environments.

Table 5. Xception Performance on Test Dataset

Class		Precision	Recall	F1-score	Support
0	cables	0.70	0.53	0.60	57
1	case	0.62	0.58	0.60	52
2	cpu	0.57	0.28	0.37	29
3	gpu	0.65	0.70	0.67	40
4	hdd	0.51	0.79	0.62	39
5	headset	0.75	0.75	0.75	53
6	keyboard	0.89	0.76	0.82	54
7	microphone	0.92	0.66	0.77	50
8	monitor	0.82	0.89	0.85	56
9	motherboard	0.60	0.71	0.65	51
10	mouse	0.47	0.79	0.59	28
11	ram	0.80	0.79	0.80	42
12	speakers	0.92	0.87	0.90	70
13	webcam	0.76	0.80	0.78	35
ccuracy				0.72	656
Macro Avg		0.71	0.71	0.70	656
Weighted Avg		0.74	0.72	0.72	656

5. CONCLUSION

In this study, we conducted a comprehensive comparative analysis of three state-of-the-art pre-trained CNN architectures, EfficientNetB7, ResNet50, and Xception, for the task of PC component image classification. Through a systematic experimental protocol involving data augmentation, transfer learning, and rigorous evaluation, we demonstrated that EfficientNetB7 achieved the highest classification performance with a test accuracy of 77.59%, outperforming both ResNet50 (72.56%) and Xception (71.80%). The model also exhibited the strongest per-class F1-scores and the lowest test loss, indicating robust generalization across the 14 component categories.

The results underscore the effectiveness of transfer learning in domain-specific computer vision tasks, even with a moderate-sized dataset. EfficientNetB7's advanced scaling methodology proved particularly advantageous in capturing discriminative visual features essential for fine-grained classification. Despite challenges in distinguishing visually similar components such as CPUs and motherboards, the proposed approach shows practical promise for real-world applications including automated inventory management, e-commerce product categorization, and PC assembly assistance systems.

Future research should focus on expanding datasets with diverse viewpoints and lighting conditions, integrating attention mechanisms like Vision Transformers for improved interpretability, developing lightweight models such as EfficientNet-Lite for edge deployment, and exploring multi-modal learning that combines visual and textual metadata. These advancements would enhance real-world applicability in inventory management, recycling systems, and interactive hardware recognition tasks.

References:

Abulfaraj, A. W., & Binzagr, F. (2025). A Deep Ensemble Learning Approach Based on a Vision Transformer and Neural Network for Multi-Label Image Classification. *Big Data and Cognitive Computing*, *9*(2), 39.

Asaniczka. (2023). PC Parts Images Dataset - Classification [Data set]. Kaggle. https://www.kaggle.com/datasets/asaniczka/pc-parts-images-dataset-classification

Baobaid, A., Meribout, M., Tiwari, V. K., & Pena, J. P. (2022). Hardware accelerators for real-time face recognition: A survey. *Ieee Access*, *10*, 83723-83739.

Chen, S., Liu, D., Pu, Y., & Zhong, Y. (2022). Advances in deep learning-based image recognition of product packaging. *Image and Vision Computing*, *128*, 104571.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1251-1258). https://arxiv.org/abs/1610.02357.

Ge, Z., Bewley, A., McCool, C., Corke, P., Upcroft, B., & Sanderson, C. (2016, March). Fine-grained classification via mixture of deep convolutional neural networks. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)* (pp. 1-6). IEEE.

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. 770-778).

Ravikumar, A., Sriraman, H., Saleena, B., & Prakash, B. (2023). Selecting the optimal transfer learning model for precise breast cancer diagnosis utilizing pre-trained deep learning models and histopathology images. *Health and Technology*, *13*(5), 721-745.

Shin, H. C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., ... & Summers, R. M. (2016). Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, *35*(5), 1285-1298.

Thirumurthy, B., Parameswaran, L., & Vaiapury, K. (2024). A deep learning and transfer learning model for intra-change detection in images. *The Journal of Supercomputing*, *80*(8), 11640-11660.

Yoo, H. J. (2015). Deep Convolution Neural Networks in Computer Vision: a Review: a Review. *IEIE Transactions on Smart Processing & Computing*, *4*(1), 35-43.

Zhang, P. (2021). E-commerce products recognition based on a deep learning architecture: Theory and implementation. *Future Generation Computer Systems*, *125*, 672-676.

Nam Tran Quy
Dai Nam University,
Vietnam.
namtq@dainam.edu.vn
ORCID: 0009-0002-5671-2747