International Journal of Informatics and Computation (IJICOM)
Vol. 8, No. 1, 2026
ISSN: 2685-8711, E-ISSN: 2714-5263 DOI: 10.35842/ijicom

Comparing Anime Character Classification with MobileNetV2, ResNet50V2, and Xception

Bulkis Kanata[1], A. Syamsul Irfan Akbara[2], Esas Rahmat Muharam[3]

Abstract
This study compares three pretrained CNN models including MobileNetV2, ResNet50V2, and Xception for anime character classification using transfer learning. We apply these models to anime images, which are challenging due to their high visual variation and stylized appearance. We also include dropout regularization to improve model generalization and reduce overfitting. We propose a simple evaluation framework using validation, test, and external internet datasets. The results show that all models perform well on anime character classification. ResNet50V2 with dropout achieves the best and most stable performance, reaching 96.36% validation accuracy, 96.00% test accuracy, and 96.53% on the internet dataset. Dropout improves performance across all models, especially for Xception, which is more prone to overfitting. MobileNetV2 delivers slightly lower accuracy but offers much higher efficiency, making it suitable for lightweight applications. We explore the main sources of classification errors and find that most mistakes occur in visually similar characters with high stylistic variation. We conclude that ResNet50V2 with dropout is the most reliable model for this task, while MobileNetV2 is the best option for efficient deployment. Future work can improve performance further using better augmentation strategies and ensemble learning.

Keywords:
Anime Character, Classification; MobileNetV2; ResNet50V2; Xception

1. Introduction

Anime content continues to expand rapidly in global entertainment markets, and we observe a significant rise in both production volume and audience consumption worldwide. The Association of Japanese Animations reports steady growth in anime production and international distribution, indicating that anime has become a dominant cultural and economic product rather than a niche medium [1]. At the same time, blockbuster releases such as *Demon Slayer: Kimetsu no Yaiba – The Movie: Mugen Train* demonstrate the global commercial impact of anime content, achieving record-breaking box office performance and reinforcing the importance of automated content understanding for large-scale media datasets [2]. However, despite this growth, we still lack robust and scalable computer vision systems that can accurately classify anime characters across diverse visual styles, poses, and artistic variations. This gap motivates the need for advanced deep learning-based classification approaches that can support large-scale anime data analysis.

We observe that image classification using deep learning has become a core research direction in computer vision, especially with the advancement of convolutional neural networks (CNNs). CNNs learn hierarchical feature representations directly from images and eliminate the need for manual feature engineering, making them highly effective for

Corresponding Author: Bulkis Kanata, University of Mataram (uqikanata@unram.ac.id)
1 Bulkis Kanata, University of Mataram, Indonesia
2 A. Syamsul Irfan Akbara, University of Mataram, Indonesia
3 Esas Rahmat Muharam, University of Mataram, Indonesia

visual recognition tasks [3]. Research shows that CNNs consistently outperform traditional machine learning approaches in image classification due to their ability to extract spatial and semantic features automatically [4]. However, despite these advantages, CNN models often struggle with anime images because anime artwork differs significantly from real-world images in terms of texture simplicity, exaggerated facial proportions, and stylistic variation. This creates a technical challenge in adapting standard CNN architectures for anime-specific classification tasks.

In anime-focused computer vision research, previous studies demonstrate growing interest in character recognition and style understanding. A benchmark study on anime style recognition highlights that anime datasets introduce high variability in drawing styles, making classification significantly more difficult compared to natural image datasets [5]. Some studies attempt to address this by using handcrafted features combined with classical machine learning methods, such as GLCM texture features and Random Forest classifiers for gender classification of anime characters [6]. However, these approaches rely heavily on manual feature extraction and fail to generalize across complex variations in anime art. Therefore, we still face a fundamental limitation in achieving high-accuracy and scalable anime character classification using traditional methods.

More recent research shifts toward deep learning-based approaches for anime-related tasks, particularly using CNN architectures. For instance, evolutionary deep learning methods using CNNs have been applied for anime storyboard recognition, showing improved adaptability in structured visual scenes [7]. Another study introduces intermediate feature aggregation techniques to improve anime character recognition performance by combining multi-level CNN features [8]. These studies indicate that deep feature representations significantly improve classification performance. However, they also reveal a recurring issue: performance varies widely depending on the chosen architecture, and no single model consistently dominates across different anime datasets and tasks.

Transfer learning has emerged as a key strategy to overcome data limitations in specialized image domains. Research in medical imaging shows that pretrained CNN models can be effectively adapted to small datasets using transfer learning, significantly improving classification accuracy [9]. Comprehensive reviews also confirm that transfer learning enhances model generalization and reduces training time, especially when datasets are limited or highly specialized [10]. However, anime datasets often exhibit domain shifts that are even more complex than medical imaging due to extreme stylistic diversity. This raises an important research challenge: determining which pretrained CNN architectures are most suitable for anime character classification and how effectively they can adapt to this domain.

Among CNN architectures, several well-established models such as MobileNetV2, ResNet50V2, and Xception have been widely used in different image classification tasks. MobileNetV2 is designed for lightweight and efficient computation using inverted residuals and linear bottlenecks, making it suitable for mobile and real-time applications [16]. ResNet-based models introduce residual learning to solve vanishing gradient problems and enable very deep networks to achieve strong performance in visual recognition tasks [17]. Xception improves performance by replacing standard convolutions with depthwise separable convolutions, enabling more efficient feature extraction [18]. Although these models are successful in general image domains, their comparative effectiveness for anime character classification remains underexplored.

Existing comparative studies of CNN architectures in other domains show that performance differences can be significant depending on dataset characteristics. For example, CNN comparisons in agricultural disease classification reveal that architecture choice strongly influences accuracy and generalization ability [23]. Similar findings appear in medical imaging, where different CNN models perform differently depending on data complexity and noise levels [25]. These results suggest that no single CNN architecture

universally performs best across all tasks. However, anime character datasets introduce unique challenges such as stylization inconsistency and limited labeled data, which require further investigation into model suitability and performance trade-offs.

Finally, although previous studies explore individual CNN architectures and some hybrid or transfer learning approaches, there is still a clear research gap in systematic comparative analysis for anime character classification. Most studies focus on either general image datasets like ImageNet [22] or domain-specific applications such as medical or agricultural imaging, rather than stylized animation datasets. Reviews of CNN advancements also highlight the need for more domain-specific evaluations to understand model behavior under different visual conditions [20][21]. Therefore, this study addresses this gap by comparing MobileNetV2, ResNet50V2, and Xception for anime character classification. We aim to identify which architecture provides the best balance between accuracy, efficiency, and generalization in anime-specific visual recognition tasks.

2. Related Works

Several studies explored the rapid growth of anime content and its increasing demand for automated analysis. The Association of Japanese Animations reported continuous expansion in anime production and global distribution, which increased the need for computational tools to manage and analyze large-scale anime datasets [1]. Box office analysis of major anime productions such as *Demon Slayer: Mugen Train* further confirmed the global impact and commercial importance of anime content [2]. These works established the relevance of anime as a high-value visual domain. However, they did not address technical solutions for automatic character-level classification. Therefore, a clear gap remained in applying computer vision techniques to structured anime character recognition.

Convolutional Neural Networks (CNNs) formed the foundation of most modern image classification systems. Theoretical and empirical studies showed that CNNs learned hierarchical feature representations effectively and improved performance across many visual tasks [3][4]. Reviews of CNN architectures further confirmed their dominance in computer vision due to strong feature extraction capabilities and scalability [10][12]. However, these studies mainly focused on natural images and general datasets. They did not specifically address stylized domains such as anime, where visual abstraction and artistic variation reduced feature consistency.

Anime-specific recognition research showed early attempts to solve character classification problems using traditional and hybrid approaches. One study applied GLCM feature extraction with Random Forest classifiers for anime gender classification [6]. The method achieved moderate performance but relied heavily on handcrafted features. Another benchmark study on anime style recognition highlighted the difficulty of distinguishing characters due to strong stylistic variations across datasets [5]. These studies demonstrated the complexity of anime visual data. However, they failed to scale effectively and lacked deep learning-based generalization.

Deep learning approaches gradually replaced traditional methods in anime-related tasks. Evolutionary CNN-based models were applied for anime storyboard recognition and showed improved adaptability to structured visual sequences [7]. Another study used intermediate feature aggregation to enhance anime character recognition performance by combining multi-level CNN features [8]. These methods improved accuracy compared to classical approaches. However, they required complex architectures and still struggled with generalization across different anime styles and datasets.

Transfer learning became an important strategy to improve performance in limited-data scenarios. Studies in medical imaging demonstrated that pretrained CNN models significantly improved classification accuracy when fine-tuned on domain-specific datasets [9][11]. Reviews confirmed that transfer learning reduced training cost and improved

convergence stability [10][12]. Although these findings were strong, most experiments focused on medical or industrial datasets. They did not fully consider stylized image domains such as anime, where domain shift remained more severe.

Modern CNN architectures such as MobileNetV2, ResNet50V2, and Xception gained wide adoption due to their efficiency and accuracy trade-offs. MobileNetV2 introduced inverted residuals and linear bottlenecks to optimize lightweight deployment [16]. ResNet-based models used residual connections to improve deep network training and prevent gradient degradation [17][26]. Xception applied depthwise separable convolutions to enhance feature extraction efficiency [18]. These architectures showed strong performance in general image classification tasks. However, their comparative behavior in anime-specific datasets remained underexplored.

Comparative studies in other domains highlighted that CNN performance varied significantly depending on dataset characteristics. Agricultural disease classification studies showed that different CNN architectures produced different accuracy levels under the same conditions [23]. Medical imaging research also reported performance differences among deep learning models depending on image complexity and noise levels [25]. These findings confirmed that architecture selection strongly influenced final performance. However, anime datasets introduced unique challenges such as artistic abstraction and inconsistent visual rules, which were not fully represented in these studies.

Recent survey works emphasized the need for deeper evaluation of CNN models across diverse domains. Studies on deep residual networks and CNN surveys highlighted ongoing challenges in model generalization, computational efficiency, and domain adaptation [12][20][26]. Other research on transfer residual networks and fine-tuned deep models confirmed that model performance depended heavily on architecture and dataset alignment [14][27]. Despite these advancements, limited research focused specifically on anime character classification using modern CNN architectures. Therefore, a systematic comparison of MobileNetV2, ResNet50V2, and Xception remained necessary to identify the most suitable model for this domain.

3. Method

A. Dataset

This study applies an experimental approach to evaluate the performance of Convolutional Neural Network (CNN) architectures for anime character classification. We construct a structured research workflow that includes data collection, preprocessing, model training, and performance evaluation. We compile a standardized dataset consisting of 6,000 training images, 1,125 validation images, and two independent test sets with 375 images each, including a controlled dataset and an internet dataset to assess generalization ability. We preprocess all images uniformly to ensure consistent input size and format. This paper applies six CNN variants based on MobileNetV2, ResNet50V2, and Xception, both with and without dropout regularization. We use transfer learning with pretrained ImageNet weights and retain the best-performing model weights to ensure reproducibility. We evaluate performance on both test sets and harvest results that clearly show the impact of dropout on reducing overfitting, improving learning stability, and balancing model capacity with generalization performance.

We construct a dataset of anime character images from three series: My Hero Academia, Jujutsu Kaisen, and Demon Slayer: Kimetsu no Yaiba, with a total of 15 classes (five characters per series). We collect the main data by extracting frames automatically from anime episodes and movies using a Python script at one-second intervals, and this

paper harvests 7,500 images (500 per class). We add 375 images from the internet (25 per class) to evaluate model robustness against visual style variation. We apply preprocessing steps that include image quality filtering, character-centered cropping with a 1:1 ratio to reduce background noise, and resizing images to 224×224 pixels to match pretrained CNN input requirements. We split the dataset into 6,000 training images, 1,125 validation images, and two test sets of 375 images each (controlled and internet), ensuring balanced evaluation for both accuracy and generalization performance.

Table. Dataset Distribution

Dataset Type	Number of Images	Description
Training	6,000	Model training data
Validation	1,125	Model tuning and validation
Test (Controlled)	375	Testing with controlled dataset
Test (Internet)	375	Testing with real-world variations
Total	7,875	Overall dataset size

B. CNN Architectures

This section compares three prominent CNN architectures including MobileNetV2, ResNet50V2, and Xception—for the task of anime character classification. The goal is to examine architectural design choices, computational efficiency, and classification performance on a dataset of anime character images. We present the core architectural motifs and provide mathematical formulations that support the forward pass and learning objectives used to evaluate these models in this context.

A typical CNN for image classification operates on an input tensor $X \in \mathbb{R}^{H\times W\times C}$ through a sequence of convolutional layers, nonlinear activations, normalization, and pooling. Convolutional layers apply learnable kernels to extract spatial features, while pooling reduces spatial resolution to aggregate information. For a standard 2D convolution with kernel $K \in \mathbb{R}^{k\times k\times C_{\text{in}}\times C_{\text{out}}}$, stride s, and padding p, the output feature map $Z \in \mathbb{R}^{H'\times W'\times C_{\text{out}}}$ is given by:

$$Z_{i,j,c} = \sum_{m=0}^{k-1}\sum_{n=0}^{k-1}\sum_{c'=1}^{C_{\text{in}}} X_{i\cdot s+m-p,\, j\cdot s+n-p,\, c'}\, K_{m,n,c',c} + b_c, \tag{1}$$

where $H' = \left\lfloor \frac{H+2p-k}{s} \right\rfloor + 1$ and W' is defined analogously. After nonlinear activation σ, a typical CNN may include Batch Normalization (BN) and non-linearities such as ReLU or ReLU6. Global average pooling (GAP) and a final fully connected (or 1×1 convolutional) layer map high-level features to class scores, followed by a softmax for:
probabilities:

$$\hat{y} = \text{softmax}(W_{\text{FC}}\, g + b_{\text{FC}}), \quad g_c = \frac{1}{H_G W_G}\sum_{i=1}^{H_G}\sum_{j=1}^{W_G} A_{i,j,c}, \tag{2}$$

where A is the activation map at the final convolutional stage and $H_G \times W_G$ are its spatial dimensions. The training objective is typically the cross-entropy loss over K classes:

$$\hat{y} = \text{softmax}(W_{\text{FC}}\, g + b_{\text{FC}}), \quad g_c = \frac{1}{H_G W_G} \sum_{i=1}^{H_G} \sum_{j=1}^{W_G} A_{i,j,c}, \tag{3}$$

$$L(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} y_{i,k} \log \hat{y}_{i,k},$$

with $\hat{y}_{i,k}$ the predicted probability for class k on example i and θ collecting all learnable parameters.

In this study, MobileNetV2 emphasizes lightweight efficiency via inverted residuals with linear bottlenecks and depthwise separable convolutions. Each building block first expands the number of channels by a factor t (the expansion factor), applies a depthwise convolution, and then projects back to a smaller number of channels. A residual (skip) connection is used when the block maintains the same resolution and number of channels. The key components are:

- Expansion: $X_{\text{exp}} = \text{ReLU6}(\text{BN}(W_{\text{expand}} * X + b_{\text{expand}}))$ with $W_{\text{expand}} \in \mathbb{R}^{k \times k \times C_{\text{in}} \times tC_{\text{in}}}$.
- Depthwise convolution: $Z = \text{DWConv}(X_{\text{exp}})$, performing a spatial convolution independently per channel: $Z^{(c)} = X_{\text{exp}}^{(c)} * K^{(c)}$.
- Projection (pointwise): $X_{\text{proj}} = \text{PWConv}(Z)$ with $W_{\text{proj}} \in \mathbb{R}^{1 \times 1 \times tC_{\text{in}} \times C_{\text{out}}}$, followed by BN and optional ReLU6.
- Residual connection: if the block has stride $s = 1$ and $C_{\text{out}} = C_{\text{in}}$, the output is $Y = X + X_{\text{proj}}$;
- otherwise, $Y = X_{\text{proj}}$.

In compact form, a MobileNetV2 inverted residual block can be expressed as a sequence of transformations culminating in a residual addition when the identity mapping is feasible. A typical forward mapping for a block with expansion factor t, input channels C_{in}, and output channels C_{out} is:

$$Y = \begin{cases} X + \text{PW}(\text{DW}(\text{ReLU6}(\text{BN}(W_{\text{expand}} * X)))), & \text{if } s = 1 \text{ and } C_{\text{out}} = \\ \text{PW}(\text{DW}(\text{ReLU6}(\text{BN}(W_{\text{expand}} * X)))), & \text{otherwise} \end{cases}$$

This design reduces computational cost while preserving representational capacity, making MobileNetV2 well-suited for on-device anime character classification where latency and memory are constrained. \

ResNet50V2

ResNet50V2 relies on residual learning with pre-activation bottleneck blocks to improve training dynamics and accuracy for deeper networks. Each bottleneck block uses a 1×1 reduction, a 3×3 convolution, and a 1×1 expansion, with skip connections that add the block output to the input when dimensions match. Pre-activation means BN and ReLU are applied before the convolutions within the block. A canonical bottleneck block can be described as:

- Pre-activation: $\hat{X} = \mathrm{ReLU}(\mathrm{BN}(X))$;
- 1×1 reduction: $X_r = \mathrm{Conv}_{1\times1}(\hat{X})$;
- 3×3 convolution: $X_{3\times3} = \mathrm{Conv}_{3\times3}(X_r)$;
- 1×1 expansion: $X_e = \mathrm{Conv}_{1\times1}(X_{3\times3})$;
- Skip connection: if the input and output dimensions match and stride is 1, then the block output is $Y = X + X_e$; otherwise, $Y = X_e$ after a projection to the appropriate dimension.

Mathematically, a residual block implements the identity mapping plus a learned residual function:

$$\hat{y} = \mathrm{softmax}(W_{\mathrm{FC}}\, g + b_{\mathrm{FC}}), \quad g_c = \frac{1}{H_G W_G} \sum_{i=1}^{H_G} \sum_{j=1}^{W_G} A_{i,j,c}, \tag{4}$$

$$H^{(l+1)} = H^{(l)} + \mathcal{F}(H^{(l)}, W^{(l)}),$$

where $\mathcal{F}$ represents the sequence of BN, ReLU, and convolutional operations inside the block. In the pre-activation variant, BN and ReLU are applied before the convolutions, which has been shown to stabilize training for deep networks. For anime character classification, ResNet50V2 offers a strong balance of depth, feature richness, and robust optimization dynamics.

In this paper, we adopted Xception with depthwise separable convolutions as its core building block, effectively replacing standard convolutions with a depthwise convolution followed by a pointwise convolution. This leads to strong representational power with reduced computational cost. A typical Xception block comprises:

- Depthwise convolution: $Z = \mathrm{DWConv}(X)$ with per-channel spatial filtering,
- Pointwise convolution: $Y = \mathrm{PWConv}(Z)$ to mix channel information,
- Nonlinearity and normalization between stages,
- Optional residual connections linking blocks with matching dimensions.

In mathematical terms, a depthwise separable convolution is the composition of a depthwise operation and a pointwise operation:

$$\mathrm{DSConv}(X) = \mathrm{PW}(\mathrm{DW}(X)),$$

where $\mathrm{DW}(X)$ applies a separate spatial kernel to each input channel and PW combines the channels via a 1×1 convolution. Xception stacks multiple DSConv blocks to form a deep, highly parameter-efficient network. This architecture often yields strong performance

on visual recognition tasks, including anime character classification, while maintaining relatively lower computational costs compared with very deep standard CNNs.

In training and evaluation process, across all three architectures, the forward pass maps an input image $x \in \mathbb{R}^{H \times W \times C}$ to a class probability distribution $\hat{y} \in \Delta^{K-1}$, where Δ^{K-1} is the $(K-1)$-simplex. The high-level forward relation can be expressed as:

$$\hat{y} = \text{softmax}(W_{\text{FC}}\, g + b_{\text{FC}}), \quad g_c = \frac{1}{H_G W_G} \sum_{i=1}^{H_G} \sum_{j=1}^{W_G} A_{i,j,c}, \tag{5}$$

$$\hat{y} = \text{softmax}(W_{\text{FC}}\, g(x;\theta) + b_{\text{FC}}),$$

where $g(x;\theta)$ denotes the global feature representation produced by the CNN (e.g., after GAP or a global pooling stage) and θ collects all learnable parameters from convolutional filters, BN, and the final classifier. The loss used to train the network is typically the crossentropy loss:

$$L(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} y_{i,k} \log \hat{y}_{i,k}, \tag{6}$$

with $y_{i,k}$ the ground-truth indicator for class k on example i.

For a rigorous comparison, one should standardize input preprocessing (e.g., resize to a common resolution such as 224×224), apply consistent data augmentation (random crops, flips, color jitter), and use identical training objectives (cross-entropy) and optimization protocols (e.g., SGD with momentum or Adam, learning rate schedules). Reportable metrics include top-1 accuracy, top-5 accuracy (if applicable), parameter counts, FLOPs, and measured latency on representative hardware. The mathematical framing above provides a uniform basis to compare architectures via their forward mappings, residual connections, and pooling strategies, facilitating a principled assessment of trade-offs between accuracy and efficiency in anime character classification tasks.

C. Model Flow

This paper utilizes three pretrained CNN architectures including MobileNetV2, ResNet50V2, and Xception to compare performance in anime character classification. We select these models based on their balance between computational efficiency and feature extraction capability. We construct two variants for each architecture, a standard model and a model with dropout regularization, resulting in six configurations. MobileNetV2

applies depthwise separable convolution to reduce computational complexity and uses inverted residual bottleneck blocks, making it efficient for lightweight deployment. ResNet50V2 introduces skip connections to address the vanishing gradient problem and supports deeper network training with strong accuracy performance. Xception separates spatial and channel correlations using depthwise separable convolution, allowing efficient parameter usage while maintaining high representation capacity. We utilize a classification head with GlobalAveragePooling2D followed by a Dense (15, softmax) layer for the standard model, while the dropout variant adds Dense (128, ReLU)–Dropout(0.4)–Dense(64, ReLU)–Dropout(0.3) before the output layer. This paper harvests that the chosen dropout configuration effectively balances model regularization and learning capability across all architectures.

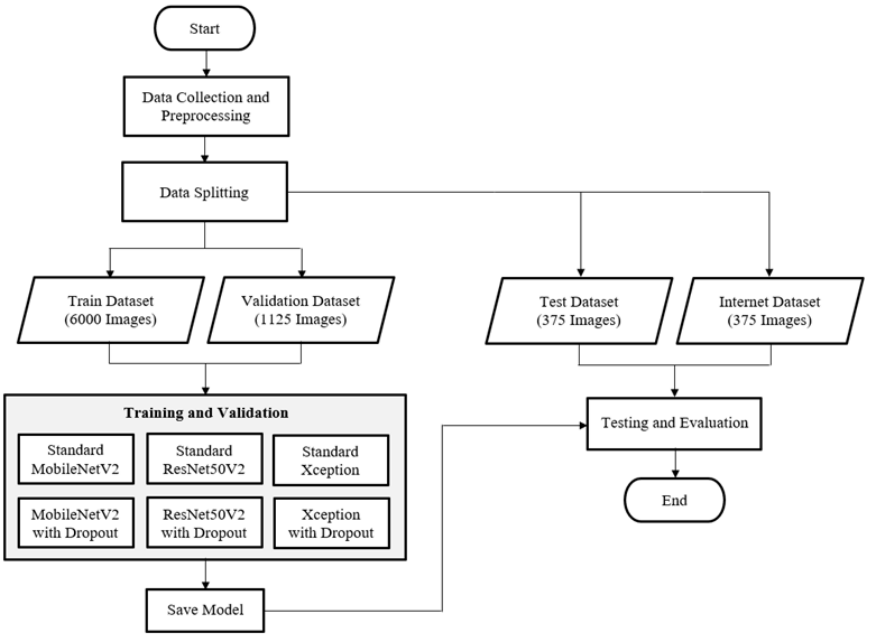


Fig. 1. Research flow

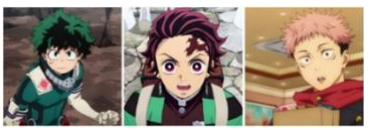


Fig. 2. Anime Characters

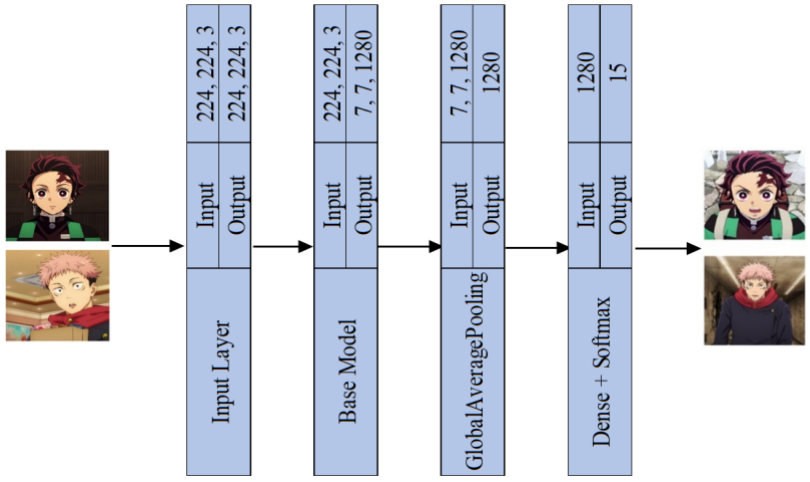


Fig. 3. Standard Models

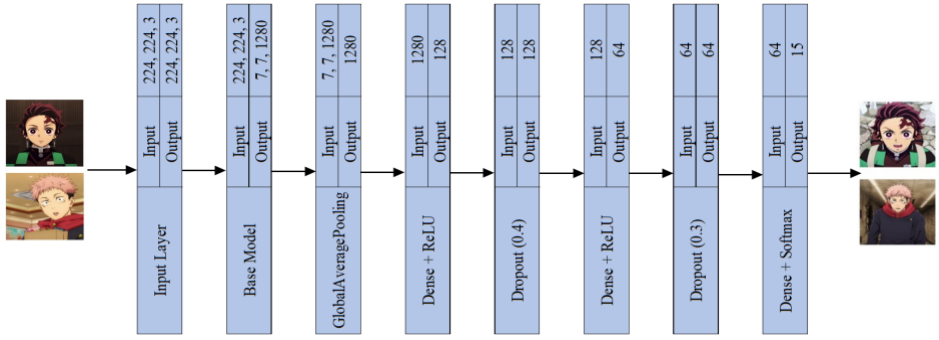


Fig. 4. Models with Dropout

D. Training Configuration

We apply a uniform hyperparameter configuration across all models to ensure fair and consistent performance comparison. This paper uses the Adam optimizer with a learning rate of 0.0003, which we select empirically due to its stability in transfer learning and fine-tuning scenarios with frozen base layers. We train each model for 10 epochs with a batch size of 32, as this setting provides sufficient convergence while reducing the risk of overfitting. We employ categorical cross-entropy as the loss function since it suits multiclass classification problems and produces stable gradients during backpropagation. To improve generalization, we apply dropout regularization with rates of 0.4 and 0.3 after the first and second dense layers. We determine this configuration experimentally to balance model learning capacity and regularization strength.

This paper applies a transfer learning approach by freezing all layers of the pre-trained base models (trainable = False) and training only the classification head. This strategy preserves general feature representations learned from ImageNet while adapting the model to specific anime visual characteristics. We implement all experiments using Python 3.12 in a Jupyter Notebook environment. The training process runs on a system equipped with an Intel Core i5-8265U processor, an NVIDIA GeForce MX250 2GB GPU, 20GB RAM, and a Windows 11 64-bit operating system. We obtain stable and reproducible training

results under this configuration, which supports reliable evaluation across different model architectures.

Table 1. Hyperparameters used for training

Name	Value
Input size	224 x 224
Batch size	32
Learning rate	0.0003
Epoch	10
Optimizer	Adam
Loss function	Categorical cross-entropy loss
Dropout rate 1	0.4
Dropout rate 2	0.3

4. Result and Analysis

5.1 Training Performance Analysis

The training results show consistent learning patterns across the six model variants, reflecting the influence of architecture selection and dropout regularization on convergence and final performance. Summary of training and validation metrics after 10 *epoch* shown in Table 2.

ResNet50V2 without dropout achieved the highest training accuracy of 99.14%, demonstrating strong representation capacity for the training data. However, the difference between training and validation accuracy of 2.70% indicates a tendency *overfitting* light even though the base layer is frozen. In contrast, ResNet50V2 with dropout shows a better learning balance, with a training accuracy of 93.11% and the highest validation accuracy of 96.36%, accompanied by *validation loss* lowest (0.1181), which indicates superior generalization ability.

MobileNetV2 demonstrated stable and efficient convergence, achieving a training accuracy of 97.28% with a relatively small training–validation difference (2.08%), indicating good generalization even with minimal explicit regularization. A MobileNetV2 variant with dropout yielded a lower training accuracy (90.62%) but maintained a competitive validation accuracy (94.58%), demonstrating the role of dropout in suppressing *overfitting*.

In contrast, Xception exhibits more challenging training dynamics. The standard model only achieves a validation accuracy of 89.16% despite a training accuracy of 92.63%, indicating limited generalization. Applying dropout improves the validation accuracy to 92.27% despite decreasing the training accuracy to 86.95%, indicating that regularization contributes significantly to learning stability. These findings indicate that the Xception architecture tends to be less optimal in representing the specific visual characteristics of anime images compared to ResNet50V2 and MobileNetV2.

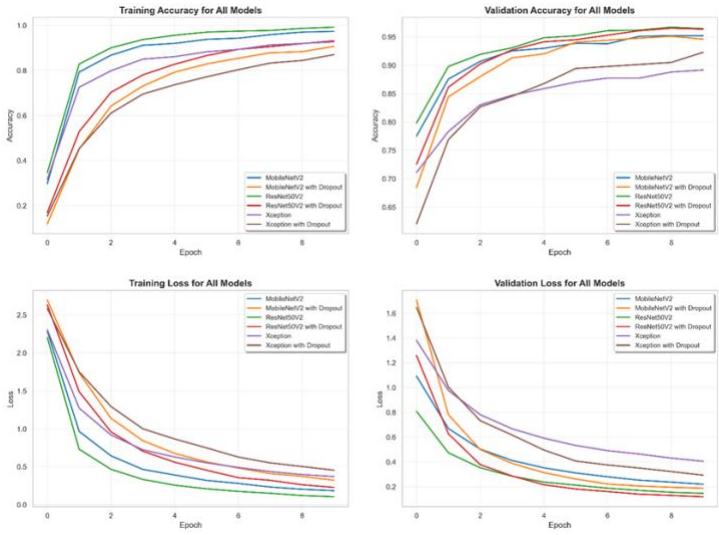


Fig. 5. Training and Validation Curves for All Models

Table 2. Training and Validation Performance of All Models

Model	Train Accuracy	Train Loss	Validation Accuracy	Validation Loss
MobileNetV2	0.9728	0.1838	0.9520	0.2193
MobileNetV2 with Dropout	0.9062	0.3212	0.9458	0.1865
ResNet50V2	0.9914	0.1036	0.9644	0.1458
ResNet50V2 with Dropout	0.9311	0.2270	0.9636	0.1181
Xception	0.9263	0.3705	0.8916	0.4050
Xception with Dropout	0.8695	0.4538	0.9227	0.2927

5.2 Test Performance Evaluation

We evaluate the models on two independent test datasets and this paper obtains clear evidence of generalization under different visual conditions. We use a controlled dataset (375 images) with consistent quality and an internet dataset (375 images) with diverse styles, backgrounds, and noise. We can observe that ResNet50V2 with dropout achieves the best performance, reaching 96.53% accuracy and the lowest loss of 0.1311 on the internet dataset. Its performance remains stable compared to the controlled test result of 96.00%, which shows strong generalization ability and confirms the effectiveness of combining residual learning with dropout regularization. We also find that MobileNetV2 experiences a performance drop from 96.00% to 91.73% in the standard model, while the dropout variant reduces this gap to 93.07%. These results indicate that we can rely on MobileNetV2 for efficient, resource-constrained applications, and this paper obtains that dropout plays an important role in handling distribution shifts and improving model robustness.

Table 3. Model Performance on Test and Internet Datasets

Model	Test Accuracy	Test Loss	Internet Accuracy	Internet Loss

MobileNetV2	0.9600	0.1953	0.9173	0.2682
MobileNetV2 with Dropout	0.9573	0.1396	0.9307	0.2278
ResNet50V2	0.9653	0.1532	0.9573	0.1977
ResNet50V2 with Dropout	0.9600	0.1357	0.9653	0.1311
Xception	0.8907	0.4144	0.8827	0.4189
Xception with Dropout	0.9253	0.2984	0.9040	0.3079

5.3 Model Performance and Confusion Matrix

We conduct a comprehensive evaluation of six CNN variants and this paper obtains clear differences in architectural performance as well as the strong impact of dropout regularization on generalization. On the controlled test dataset, we can observe that ResNet50V2 with dropout achieves the highest accuracy of 96.53%, followed by MobileNetV2 standard at 96.00% and its dropout variant at 95.73%. The small performance gap in MobileNetV2 suggests that we can attribute part of its stability to implicit regularization within its architecture. In contrast, Xception records the lowest performance, but this paper obtains a notable improvement after applying dropout, increasing accuracy from 89.07% to 92.53%. This result shows that Xception is more sensitive to overfitting on anime data and requires explicit regularization to improve learning stability.

We further evaluate the models on the internet dataset and we can confirm consistent findings across different data distributions. ResNet50V2 with dropout maintains the highest accuracy at 96.53% and shows nearly identical performance to the controlled dataset, which indicates strong generalization capability. Standard ResNet50V2 follows closely at 95.73%, while MobileNetV2 shows a larger performance drop, although its dropout variant remains more robust to distribution shifts. This paper obtains additional insight from the confusion matrix analysis, where strong diagonal dominance appears, with several characters achieving perfect classification (25/25).

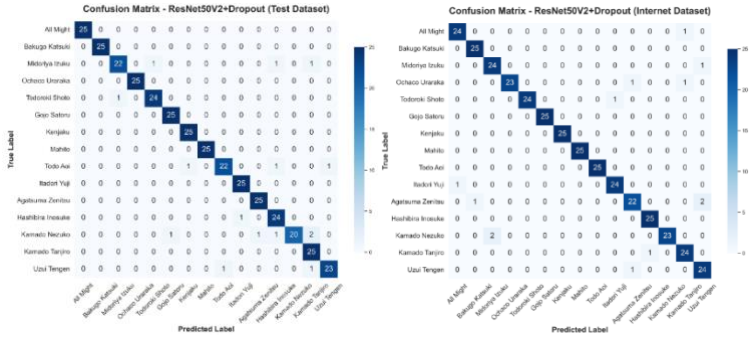
Results *confusion matrix*The internet dataset (Fig. 7) shows high consistency, with seven characters achieving perfect classification (25/25) and most other characters showing near-perfect precision. The performance stability between the test dataset and the internet dataset confirms the superior generalization ability of ResNet50V2 with dropout.

Table 4. Accuracy, Precision, Recall, and F1-Score on Test Dataset

Model	Accuracy	Precision	Recall	F1-Score
MobileNetV2	0.9600	0.9626	0.9600	0.9595
MobileNetV2 with Dropout	0.9573	0.9591	0.9573	0.9568
ResNet50V2	0.9653	0.9663	0.9653	0.9648
ResNet50V2 with Dropout	0.9600	0.9619	0.9600	0.9594
Xception	0.8907	0.8974	0.8907	0.8885
Xception with Dropout	0.9253	0.9339	0.9253	0.9251

Table 5. Accuracy, Precision, Recall, and F1-Score on Internet Dataset

Model	Accuracy	Precision	Recall	F1-Score
MobileNetV2	0.9173	0.9237	0.9173	0.9182
MobileNetV2 with Dropout	0.9307	0.9354	0.9307	0.9311
ResNet50V2	0.9573	0.9582	0.9573	0.9571
ResNet50V2 with Dropout	0.9653	0.9663	0.9653	0.9654
Xception	0.8827	0.8883	0.8827	0.8820
Xception with Dropout	0.9040	0.9106	0.9040	0.9051



(a) Test Dataset (b) Internet Dataset

Fig. 6. Confusion Matrix of ResNet50V2 with Dropout

The confusion matrix in Fig. 6(a), ResNet50V2 with Dropout for the test dataset reveals a strong diagonal structure, supporting perfect classification accuracy for all character classes. Nine characters showed perfect classification accuracy (25/25 correct classifications) predicted correctly: All Might, Bakugo Katsuki, Ochaco Uraraka, Gojo Satoru, Kenjaku, Mahito, Itadori Yuji, Agatsuma Zenitsu, and Kamado Tanjiro. Todoroki Shoto and Uzui Tengen showed near-perfect classification accuracy of 24/25 and 23/25, respectively, with some misclassifications. Midoriya Izuku was confused at a moderate level (22/25 correct) with errors allocated to Hashibira Inosuke, Kamado Nezuko, and Todoroki Shoto. Kamado Nezuko was the most confused (20/25 correct), and misclassifications ranged across the series, including Gojo Satoru, Agatsuma Zenitsu, Hashibira Inosuke, and Kamado Tanjiro. There were fewer misclassifications across the series, indicating the model correctly represents the visual information related to the series. The internet dataset confusion matrix in Fig. 6(b), demonstrates the remarkable capacity of ResNet50V2 with Dropout to maintain high accuracy in demanding real-world tasks. Correct classification of 25/25 is maintained for the seven main protagonists: Bakugo Katsuki, Gojo Satoru, Kenjaku, Mahito, Todo Aoi, Hashibira Inosuke, and Kamado Tanjiro. Very close to perfect precision is maintained for All Might (24/25), Midoriya Izuku (24/25), Todoroki Shoto (24/25), Itadori Yuji (24/25), and. Ochaco Uraraka and Kamado Nezuko (23/25) show slightly more variation, and Agatsuma Zenitsu records higher difficulty (22/25 accurate) with errors to Bakugo Katsuki and Uzui Tengen. The consistency of performance on the test and internet datasets, where the model recorded the same accuracy on internet data (96.53%).

This paper applies dropout regularization to improve the generalization performance of all evaluated architectures, and we observe that the improvement increases with model complexity. We obtain the most stable results from ResNet50V2 with dropout, while MobileNetV2 shows moderate gains with minimal performance degradation across different data distributions. Xception benefits the most from this approach, where we record a 3.46% increase in test accuracy and a 2.13% improvement in internet-based accuracy.

We further analyze character-specific performance and find that characters with strong and consistent visual features, such as Gojo Satoru, Kenjaku, and Todo Aoi, achieve high classification accuracy across all models. In contrast, characters with high visual variability, such as Midoriya Izuku, Itadori Yuji, and Kamado Nezuko, are more difficult to classify. We identify Kamado Nezuko as the most challenging case due to significant visual transformations. Most classification errors occur within the same series, which indicates that the models successfully learn artistic styles but still struggle to distinguish characters with similar visual patterns.

5. Conclusion

In this study, we applied a comparative evaluation of three pretrained convolutional neural network architectures for the task of anime character classification. We assessed their ability to learn and generalize from stylized anime images that exhibit high visual variability and abstraction. We also incorporated dropout regularization into the experimental setup to examine its effect on improving model robustness. The results confirmed that transfer learning from pretrained CNN models effectively adapts to the unique characteristics of anime datasets.

We propose that ResNet50V2 with dropout provides the most reliable and stable performance among the evaluated models. It achieved the highest results across all evaluation settings, with a validation accuracy of 96.36%, a test accuracy of 96.00%, and an internet dataset accuracy of 96.53%. This consistent performance indicates strong generalization across different data distributions. We also observed that dropout regularization consistently improved model performance, particularly by reducing overfitting. The improvement was most significant in Xception, which initially showed higher sensitivity to dataset variation.

We further explore the trade-off between accuracy and computational efficiency across the evaluated architectures. MobileNetV2 demonstrated competitive classification performance while maintaining significantly lower computational cost, making it suitable for deployment in resource-limited environments. In contrast, deeper architectures such as ResNet50V2 and Xception achieved higher representational capacity but required more computational resources. Error analysis revealed that misclassifications mainly occurred in characters with high intra-class variation and stylistic ambiguity. This finding suggests that future research should focus on advanced data augmentation techniques and ensemble learning strategies to further improve robustness and classification accuracy.

References

[1] The Association of Japanese Animations, *"Japan Anime Data,"* 2024. [Online]. Available: https://aja.gr.jp/english/japan-anime-data

[2] Box Office Mojo, *"Demon Slayer: Kimetsu no Yaiba – The Movie: Mugen Train,"* 2020. [Online]. Available: https://www.boxofficemojo.com/title/tt11032374/

[3] M. M. Taye, "Theoretical understanding of convolutional neural network: Concepts, architectures, applications, and future directions," *Computation*, vol. 11, no. 3, p. 52, 2023, doi: 10.3390/computation11030052.

[4] N. Sharma, V. Jain, and A. Mishra, "An analysis of convolutional neural networks for image classification," *Procedia Computer Science*, vol. 132, pp. 377–384, 2018, doi: 10.1016/j.procs.2018.05.198.

[5] H. Li *et al.*, "A challenging benchmark of anime style recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2022, pp. 4720–4729, doi: 10.1109/CVPRW56347.2022.00518.

[6] D. I. Mulyana and V. V. Pramansah, "Gender classification for anime character face image using random forest classifier method and GLCM feature extraction," *JUITA: Jurnal Informatika*, vol. 10, no. 2, pp. 243–250, 2022, doi: 10.30595/juita.v10i2.13833.

[7] S. Fujino, T. Hatanaka, N. Mori, and K. Matsumoto, "Evolutionary deep learning based on deep convolutional neural network for anime storyboard recognition," *Neurocomputing*, vol. 338, pp. 393–398, 2019, doi: 10.1016/j.neucom.2018.05.124.

[8] E. A. Rios, M. C. Hu, and B. C. Lai, "Anime character recognition using intermediate features aggregation," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2022, pp. 424–428, doi: 10.1109/ISCAS48785.2022.9937519.

[9] S. Minaee *et al.*, "Deep-COVID: Predicting COVID-19 from chest X-ray images using deep transfer learning," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 10, pp. 2897–2905, 2020, doi: 10.1109/JBHI.2020.3036341.

[10] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, “A review of convolutional neural networks in computer vision,” *Artificial Intelligence Review*, vol. 57, no. 4, 2024, doi: 10.1007/s10462-024-10721-6.
[11] K. Suzuki, “Overview of deep learning in medical imaging,” *Radiological Physics and Technology*, vol. 10, no. 3, pp. 257–273, 2017, doi: 10.1007/s12194-017-0406-5.
[12] L. Alzubaidi *et al.*, “Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions,” *Journal of Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-021-00444-8.
[13] Howard *et al.*, “Searching for MobileNetV3,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 1314–1324, doi: 10.1109/ICCV.2019.00140.
[14] M. Rahimzadeh and A. Attar, “A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2,” *Informatics in Medicine Unlocked*, vol. 19, p. 100360, 2020, doi: 10.1016/j.imu.2020.100360.
[15] N. Jinsakul, C. F. Tsai, C. E. Tsai, and P. Wu, “Enhancement of deep learning in image classification performance using Xception with the Swish activation function for colorectal polyp preliminary screening,” *Mathematics*, vol. 7, no. 12, p. 1170, 2019, doi: 10.3390/math7121170.
[16] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.
[17] L. Zhang, Y. Bian, P. Jiang, and F. Zhang, “A transfer residual neural network based on ResNet-50 for detection of steel surface defects,” *Applied Sciences*, vol. 13, no. 9, p. 5260, 2023, doi: 10.3390/app13095260.
[18] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 1251–1258, doi: 10.1109/CVPR.2017.195.
[19] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks: An overview and application in radiology,” *Insights into Imaging*, vol. 9, no. 4, pp. 611–629, 2018, doi: 10.1007/s13244-018-0639-9.
[20] K. L. Kermanidis, M. Maragoudakis, and M. Krichen, “Convolutional neural networks: A survey,” *Computers*, vol. 12, no. 8, p. 151, 2023, doi: 10.3390/computers12080151.
[21] Y. Gonzalez Tejeda and H. A. Mayer, “Deep learning with convolutional neural networks: A compact holistic tutorial with focus on supervised regression,” *Machine Learning and Knowledge Extraction*, vol. 6, no. 4, pp. 2753–2782, 2024, doi: 10.3390/make6040132.
[22] J. Deng *et al.*, “ImageNet: A large-scale hierarchical image database,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.
[23] M. T. Ahad, Y. Li, B. Song, and T. Bhuiyan, “Comparison of CNN-based deep learning architectures for rice diseases classification,” *Artificial Intelligence in Agriculture*, vol. 9, pp. 22–35, 2023, doi: 10.1016/j.aiia.2023.07.001.
[24] S. Hussain, M. Roshanzamir, T. Ekmekyapar, and B. Ta, “Exemplar MobileNetV2-based artificial intelligence for robust and accurate diagnosis of multiple sclerosis,” *Diagnostics*, vol. 13, no. 19, p. 3030, 2023, doi: 10.3390/diagnostics13193030.
[25] E. Acar, E. Şahin, and İ. Yılmaz, “Improving effectiveness of different deep learning-based models for detecting COVID-19 from computed tomography images,” *Neural Computing and Applications*, vol. 33, no. 24, pp. 17589–17609, 2021, doi: 10.1007/s00521-021-06344-5.
[26] M. Shafiq and Z. Gu, “Deep residual learning for image recognition: A survey,” *Applied Sciences*, vol. 12, no. 18, p. 8972, 2022, doi: 10.3390/app12188972.
[27] M. A. Talukder *et al.*, “An efficient deep learning model to categorize brain tumor using reconstruction and fine-tuning,” *Expert Systems with Applications*, vol. 230, p. 120534, 2023, doi: 10.1016/j.eswa.2023.120534.
[28] Markoulidakis *et al.*, “Multiclass confusion matrix reduction method and its application on Net Promoter Score classification problem,” *Technologies*, vol. 9, no. 4, p. 81, 2021, doi: 10.3390/technologies9040081.