Image Classification for Caltech-101

Qianyu Fan

1. Dataset and Features

1.1. Dataset Details

In this project, we used the Caltech-101 dataset to perform an image classification task across 101 distinct object categories. After excluding the BACKGROUND_Google class, the dataset contains a total of 8,677 images. Each image has an approximate resolution of 300×200 pixels. As shown in Figure 1, the distribution of images per category is imbalanced.

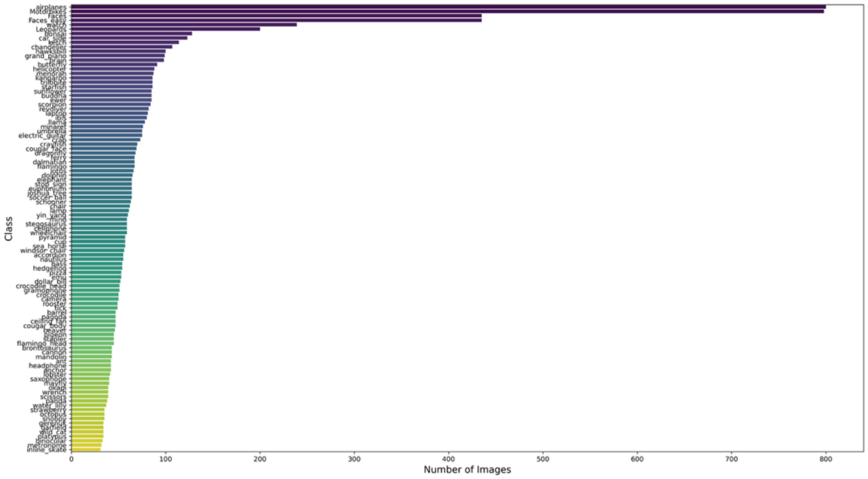


Figure 1. Caltech-101 Class Distribution (after removing BACKGROUND_Google)

1.2. Data Preprocessing

In preparation for model training and evaluation, we performed a series of data preprocessing steps. The dataset was loaded and organized into image-label pairs, with each image read using the Python Imaging Library (PIL) and convert to RGB format. The data was stratified and split into training, validation, and testing subsets, comprising 70%, 15%, and 15% of the data, respectively. To improve model generalization and reduce overfitting, the training set underwent data augmentation techniques, including random horizontal flipping and color jittering (brightness and contrast adjustments). The validation and test sets were only resized and normalized. All images were resized to 128x128 pixels and normalized using mean and standard deviation of the ImageNet dataset to match the input statistics expected by pretrained models. Finally, the processed datasets were loaded into PyTorch DataLoader objects to enable efficient mini-batch training and evaluation.

2. Methods

2.1. ResNet

Residual Network (ResNet), introduced by He et al. in 2015 [2], addressing the vanishing gradient problem commonly encountered in very deep networks. ResNet introduces residual (skip) connections that allow layers to learn residual functions relative to their inputs, improving gradient flow and enabling the training of very deep architectures. Each residual block consists of a series of convolutional layers followed by batch normalization and ReLU activation, with the input added back to the output through a shortcut connection. Due to its strong performance and stability, ResNet widely used for image classification.

In our approach, we employed the ResNet18, which consists of 18 layers organized into four residual blocks. Compared to deeper variants such as ResNet-50, ResNet-18 offers a favorable trade-off between computational efficiency and classification performance, making it particularly suitable for small-sized datasets like Caltech-101. To leverage transfer learning, the model was initialized with pre-trained ImageNet weights and fine-tuned it by unfreezing the final residual block and the fully connected layer. The final fully connected layer was replaced to output 101 logits corresponding to the number of object categories in the dataset. The overall architecture is illustrated in Figure 2.

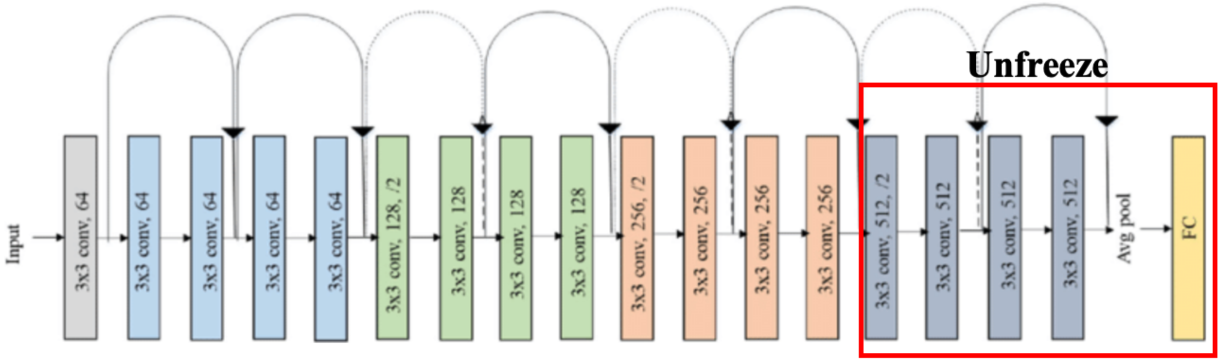


Figure 2. ResNet18 Architecture (with unfreeze the final residual block and the fully connected layer)

2.2. EfficientNet

EfficientNet, proposed by Tan and Le in 2019 [4], is a family of convolutional neural networks optimized for both accuracy and efficiency across various computational budgets. Its key innovation lies in a compound scaling method, which uniformly scales the network's depth, width, and resolution using a set of fixed coefficients. At its core, EfficientNet leverages the Mobile Inverted Bottleneck Convolution (MBConv) and squeeze-and-excitation (SE) attention modules to enhance representational capacity without substantially increasing computational cost, making it well-suited for resource-constrained settings.

We adopted the EfficientNet-B1, which provides a balance between model complexity and classification accuracy. Compared to smaller models such as EfficientNet-B0, the B1 configuration operates on higher-resolution inputs and contains more channels per layer, allowing it to capture richer semantic representations. The model was initialized with pre-trained ImageNet weights to leverage transfer learning. To adapt the model to the target task, all layers were initially frozen, and the final two feature blocks and classification head were unfrozen for fine-tuning. Additionally, all Batch Normalization layers were explicitly frozen during fine-tuning to prevent updates to the running mean and variance statistics, which can become unstable when the target dataset is relatively small. This selective unfreezing strategy allows the network to refine its high-level feature extraction while retaining robust low-level representations from pretraining. The original classification head was replaced with a fully connected layer corresponding to the 101 output categories. This fine-tuning procedure enables EfficientNet-B1 to efficiently adapt pre-trained convolutional representations to the Caltech-101 dataset, achieving strong performance with minimal over-parameterization. The overall architecture is displayed in Figure 3.

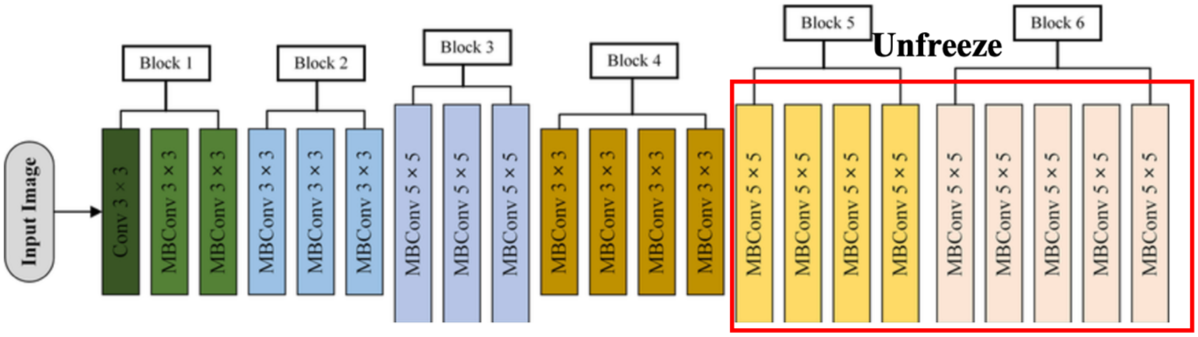


Figure 3. EfficientNet-B1 Architecture (with unfreeze the final block and classification head)

2.3. Vision Transformer (ViT)

Vision Transformer (ViT), introduced by Dosovitskiy et al. in 2020 [1], represents a paradigm shift in computer vision by directly applying the Transformer architecture, originally developed for natural language processing, to sequences of image patches. Instead of using convolutional operations, ViT divides an image (default is 224x224) into fixed-size non-overlapping patches (16x16 pixels), linearly embeds each patch, and adds learnable positional encodings to preserve spatial relationships. These embedded patch tokens, along with a special class token, are then processed by a stack of Transformer encoder layers that rely entirely on multi-head self-attention mechanisms. This design allows ViT to model long-range dependencies between image regions, leading to strong performance.

In our case, we utilized the ViT-B/16 model pre-trained on ImageNet-1K. However, unlike the standard ViT configuration that operates on (224x224) images (producing a 14x14 grid of 196 patches), we resized all images to 128x128 pixels at the beginning. This change alters the number of patches from 14x14 to 8x8, requiring careful adaptation of the positional encodings to maintain architectural compatibility. To address this, the original positional embeddings were resized via bilinear interpolation to match the new spatial resolution, ensuring consistent positional information for the transformer's attention mechanism. Without this adjustment, the model would encounter a dimensional mismatch and fail to process the modified input size.

Following positional encoding adjustment, all model parameters were initially frozen to retain the pretrained representations. The last two Transformer encoder blocks were unfrozen for fine-tuning, allowing high-level semantic features to adapt to the Caltech-101 domain. The classification head was replaced with a fully connected layer producing 101 output logits, corresponding to the dataset's object categories. This selective fine-tuning strategy enables efficient transfer learning, allowing ViT-B/16 to generalize effectively to smaller input resolutions while leveraging its strong representational capacity learned from large-scale pretraining. The overall architecture is showed in Figure 4.

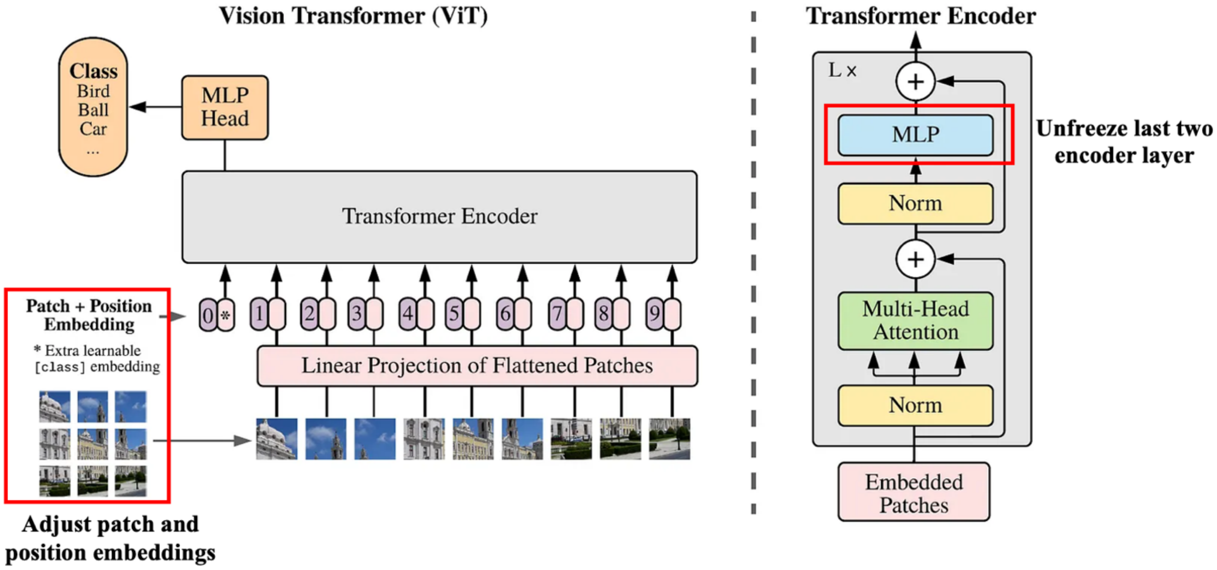

Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0
*
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches
Adjust patch and position embeddings
Transformer Encoder
L x
MLP
Unfreeze last two encoder layer
Norm
Multi-Head Attention
Norm
Embedded Patches

Figure 4. ViT Architecture (with adjusting patch and position embeddings and unfreeze last two encoder layers)

For these three models, we employed the cross-entropy loss [3] as the objective function for optimization. Cross-entropy is widely used in multi-class classification tasks as it measures the dissimilarity between the predicted probability distribution and the true class labels. By minimizing cross-entropy loss, the models are encouraged to assign higher probabilities to the correct classes, thus improving classification performance.

$$\mathcal{L} = -\sum_{i=1}^{C} y_i \log(\widehat{p_i})$$

3. Experiments

3.1. Evaluation Metrics

To rigorously evaluate model performance on the Caltech-101 dataset, we employed a set of standard classification metrics that assess both global accuracy and class-level behavior.

Overall accuracy served as the primary metric, calculated as the proportion of correctly predicted labels over all test samples. In addition, we reported Top 5 accuracy, which reflects the proportion of samples where the true label is among the model's five most confident predictions.

To further analyze performance across categories, we computed per-class accuracy, obtained by averaging classification accuracy across all individual classes. This highlights disparities in model performance and helps identify underrepresented or challenging categories.

A confusion matrix was also generated to visualize misclassification patterns and to identify systematic confusion between similar classes.

Additionally, we evaluated precision, recall, and F1-score to provide a more comprehensive assessment. These were computed using both macro averaging (equal weight per class) and weighted averaging (proportional to class size), balancing the need to reflect both fairness and dataset composition.

3.2. Experiment Setup

To ensure the consistency and repeatability of our experiments, we run them all on the Google Colab platform with an A100 GPU. Each model was trained using the Adam optimizer with a learning rate of 1e-4 and a weight decay of 1e-4 to prevent overfitting. We trained each model for up to 30 epochs with a mini-batch size of 32. Early stopping was implemented with a patience of 5 epochs, halting training when the validation loss showed no improvement, thereby promoting efficient convergence. Under this setup, ResNet18 terminates at epoch 19, EfficientNet-B1 terminates at epoch 19, and ViT-B/16 terminates at epoch 14.

4. Results

4.1. Accuracy, Top 5 Accuracy, and Macro & Weighted Metrics

Table 1 summarizes the performance of the three models, including ResNet18, EfficientNet-B1, and ViT-B/16 on the test dataset. Overall, all three models achieve strong results, with accuracies exceeding 92%. Among them, ViT-B/16 demonstrates the best performance, achieving the highest overall accuracy (0.943) and weighted F1-score (0.942). This indicates that the Vision Transformer is most effective at capturing diverse object features in the dataset.

In terms of macro-averaged metrics, which treat all classes equally, ViT-B/16 again outperforms the CNN-based models with a macro F1-score of 0.914, suggesting better balance across classes and improved handling of class imbalance. ResNet18 and EfficientNet-B1 show comparable results, with slightly lower macro F1-scores (0.893 and 0.895, respectively).

For Top 5 accuracy, all models achieve excellent results above 0.98, indicating that the correct class is almost always among the top five predictions. This suggests that even when the Top 1 prediction is incorrect, the model's understanding of object categories remains close to accurate.

Overall, ViT-B/16 provides the most robust and balanced performance, likely due to its self-attention mechanism that captures long-range dependencies and global image context more effectively than traditional convolutional architectures.

Table 1. Model results

	ResNet18	EfficientNet-B1	ViT-B/16
Accuracy	0.929	0.929	0.943
Top 5 Accuracy	0.993	0.986	0.987
Macro Averages			
Precision	0.916	0.908	0.927
Recall	0.892	0.895	0.911
F1-Score	0.893	0.895	0.914
Weighted Averages			
Precision	0.938	0.933	0.947
Recall	0.929	0.929	0.943
F1-Score	0.928	0.928	0.942

4.2. Per-class Accuracy

The per-class accuracy results in Table 2 (See Appendice) reveal the presence of class imbalance effects across all three models; nevertheless, each model demonstrates consistently strong performance for the majority of categories, which suggests that adopted data augmentation and regularization strategies effectively mitigated bias. Notably, 21 categories are perfectly classified (100% accuracy) by all models, including *Leopards, Motorbikes, accordion, binocular, bonsai, buddha, car_side, dalmatian, dollar_bill, ferry, Garfield, inline_skate, laptop, metronome, minaret, sea_horse, soccer_ball, stop_sign, strawberry, sunflower, trilobite, and yin_yang*. These categories tend to have consistent visual patterns, clear object boundaries, and minimal background clutter, making them easier to recognize across architectures.

ResNet18 achieves stable and reliable performance. It excels in classes with strong texture and edge information (e.g., *Faces_easy*, *butterfly*, *cannon*, *cougar_body*, *crab*, *cup*, *dolphin*, *euphonium, lotus, mayfly, octopus, pigeon, pizza, revolver, schooner, scorpion, tick*), though its performance decreases slightly for irregular or fine-grained objects such as *anchor* (0.667), *emu* (0.625), and *crocodile* (0.375). EfficientNet-B1 exhibits similar accuracy but demonstrates slightly better performance on scale-sensitive or complex shapes like *anchor* (0.833), *chandelier* (1), *crocodile* (0.875), *ant* (0.833), and *platypus* (1), likely due to its compound scaling strategy that balances network depth and resolution. However, it occasionally underperforms in classes with less distinctive color or shape cues, such as *lobster* (0.333).

In contrast, ViT-B/16 delivers the most balanced and robust per-class accuracy overall. It achieves perfect classification across a broader set of categories, including *Faces, ant, bass, brontosaurus, ewer, grand_piano, hawksbill, lobster, ketch, rooster, and scissors*, where CNN-based models occasionally misclassify. The transformer's self-attention mechanism allows it to capture *global context and long-range dependencies*, leading to improved recognition of objects with varied poses or backgrounds. Nonetheless, ViT-B/16 shows slight drops in certain small or low-texture classes, such as *anchor* (0.333) and *water_lilly* (0.333), possibly due to local detail loss in the patch-based tokenization process.

While all models achieve high accuracy, ViT-B/16 stands out with superior generalization and category-level consistency. The fact that all three architectures perfectly classify the same 21 categories highlights that some classes are visually well-defined and separable, whereas others remain challenging even for modern deep models.

4.3. Confusion Matrix

Figure 5 (See Appendice) reveals the Top 10 most difficult classes for each model to correctly identify and classify. For ResNet18, classification performance remains inconsistent, with noticeable off-diagonal dispersion across semantically similar categories. Substantial confusion persists among visually correlated classes including *crayfish, crocodile,* and *water lily*. This indicates that the convolutional backbone of ResNet18 struggles to capture fine-grained inter-class variations, likely due to limited receptive field and reliance on local texture features.

In contrast, EfficientNet-B1 exhibits enhanced diagonal concentration, reflecting improved intra-class consistency and inter-class separability. Nevertheless, residual confusion among morphologically similar aquatic species (*crab, lobster, crayfish*) implies that even EfficientNet-B1 encounters limitations in distinguishing fine-grained visual attributes.

The ViT-B/16 model further reduces misclassification noise, yielding a more stable diagonal structure across categories. Its global self-attention mechanism enables more effective modeling of long-range dependencies, resulting in fewer off-diagonal mispredictions compared to convolution-based architectures. Misclassifications are primarily confined to the *crab* and *crocodile_head* classes.

5. Observations and Ablation Studies

5.1. Observations

The training and loss analyses of the three models reveal important insights into their performance dynamics, as illustrated in Figure 6**.** The initially higher training loss compared to validation loss—attributable to the use of data augmentation—indicates effective regularization and mitigation of early overfitting. As training progresses, all three models exhibit a steady decline in training loss, reflecting efficient optimization and progressive feature learning. However, the ViT-B/16 model shows one fluctuation in validation loss across several epochs, suggesting potential overfitting or instability when generalizing to unseen data.

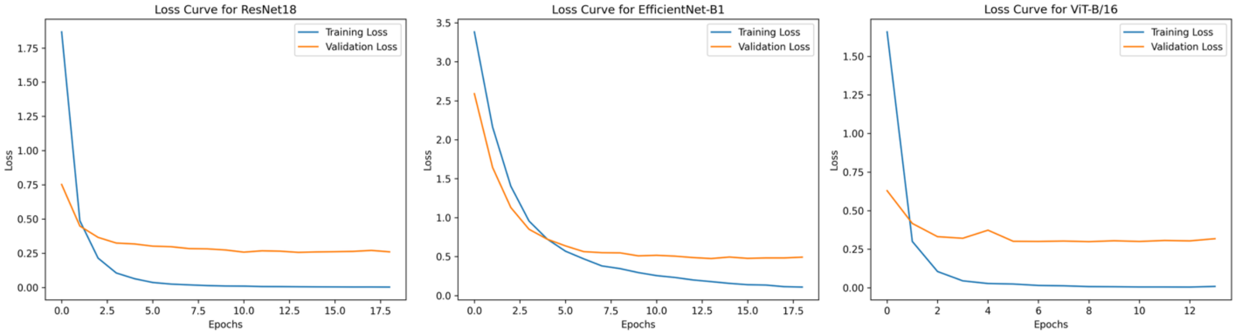


Fig 6 Loss curves for models

5.2. Ablation Study: Effect of Image Size and Optimizer

To investigate how image resolution and optimization strategy influence model performance, two ablation experiments were conducted on the ResNet18 architecture using different image sizes (64×64 vs. 128×128) and optimizers (SGD vs. Adam). As shown in Figure 7, models trained with the Adam optimizer generally converged faster and achieved lower training and validation loss compared to those trained with SGD, indicating more efficient gradient updates and better optimization stability. In contrast, models trained with smaller input images (64×64) exhibited higher loss values and lower final accuracy, suggesting that reduced spatial resolution limits the model's ability to capture fine-grained details necessary for object discrimination.

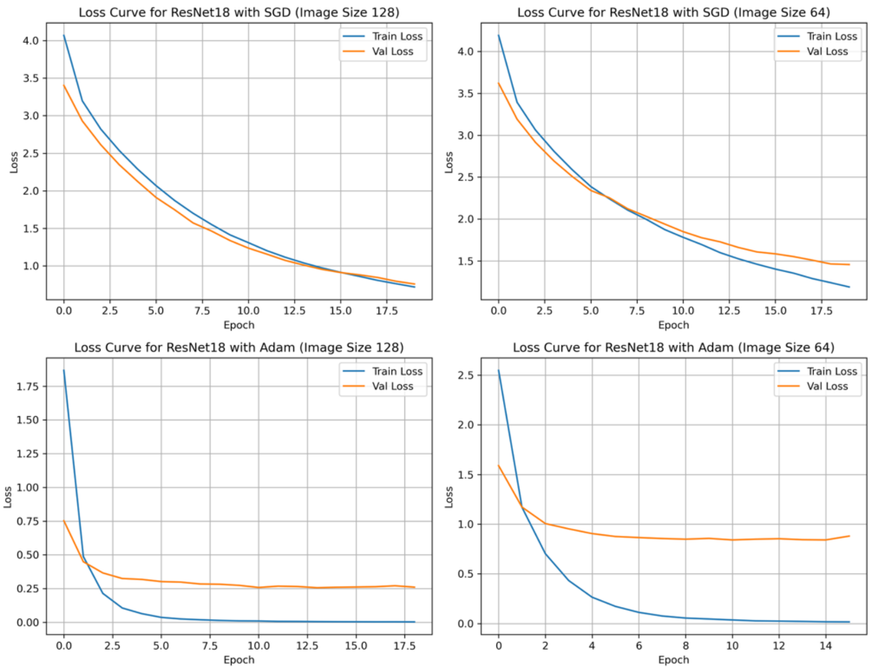


Figure 7. ResNet18 Loss Curve for different image sizes and optimizers

The quantitative results in Table 3 confirm these trends: the highest accuracy (0.929) was achieved using Adam with 128×128 images, while the lowest accuracy (0.675) occurred with SGD and 64×64 inputs. Adam performs better likely because of its adaptive learning rate mechanism, which adjusts step sizes for each parameter. Since dataset includes objects of diverse scales, textures, and shapes**,** this adaptability helps the model converge faster and handle variations more effectively than standard SGD. These findings highlight that both optimizer choice and image resolution substantially affect model convergence and final performance on the Caltech-101 dataset.

Table 3. ResNet18 Accuracy

	SGD	Adam
128x128	0.836	0.929
64x64	0.675	0.793

6. Interpretations and Lessons Learned

This project evaluated three representative deep learning architectures, ResNet18, EfficientNet-B1, and ViT-B/16, on the Caltech-101 dataset to investigate how different network designs and optimization strategies affect image classification performance. Through systematic experiments and ablation studies, several key insights emerged.

First, model architecture plays a crucial role in feature extraction and generalization. The transformer-based ViT-B/16 consistently achieved the highest overall accuracy (94.3%) and the most stable per-class performance, benefiting from its global attention mechanism that effectively models long-range dependencies. Convolutional models such as ResNet18 and EfficientNet-B1 also performed competitively (both 92.9% accuracy), confirming that CNNs remain highly effective for moderately sized image datasets. However, ViT demonstrated stronger robustness across diverse object scales and backgrounds, suggesting that attention-based models generalize better when object structure varies widely.

Second, the optimizer and input resolution significantly influence model convergence and stability. The ablation study comparing SGD and Adam optimizers across 64×64 and 128×128 image sizes revealed that Adam leads to faster convergence and lower final loss, particularly for higher-resolution inputs. This aligns with Adam's adaptive learning rate mechanism, which improves gradient updates for heterogeneous feature distributions, an important advantage for Caltech-101's visually diverse categories. Conversely, smaller input sizes (64×64) reduced accuracy across all models due to information loss, emphasizing the importance of maintaining sufficient spatial detail for fine-grained recognition.

Third, per-class analysis highlighted that 21 categories were perfectly classified by all models. In contrast, lower-performing classes (e.g., *anchor, crocodile, water_lilly*) indicate that background clutter, small object size, or ambiguous boundaries remain key challenges. This suggests that future improvements could focus on data augmentation or attention mechanisms targeting local details.

Finally, this project demonstrates that fine-tuning pretrained models, rather than training from scratch, yields excellent performance even with limited data. By adjusting positional embeddings for ViT and selectively unfreezing layers in CNNs, the models were efficiently adapted to Caltech-101 with minimal overfitting.

In summary, the experiments show that while CNNs remain reliable and computationally efficient, vision transformers deliver superior flexibility and generalization. The combination of adaptive optimization (Adam), higher input resolution, and transformer-based architectures provides a powerful framework for tackling real-world image classification tasks with complex visual diversity.

References

[1] Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." arXiv preprint arXiv:2010.11929 (2020).

[2] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

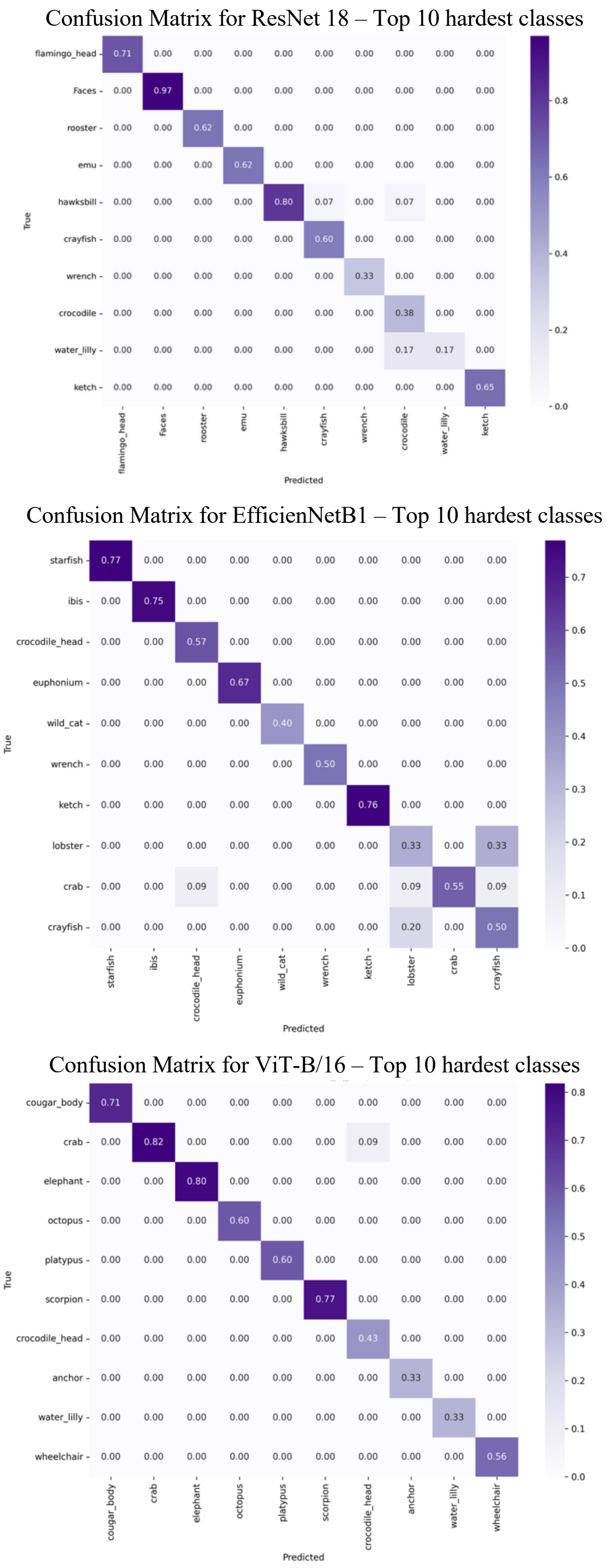
[3] Mao, Anqi, Mehryar Mohri, and Yutao Zhong. "Cross-entropy loss functions: Theoretical analysis and applications." *International conference on Machine learning*. pmlr, 2023.

[4] Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." *International conference on machine learning*. PMLR, 2019.

Appendices

Table 2. Per-class accuracy

	ResNet18	EfficientNet-B1	ViT-B/16
Faces	0.970	0.985	1
Faces_easy	1	0.970	0.985
Leopards	1	1	1
Motorbikes	1	1	1
accordion	1	1	1
airplanes	0.992	1	1
anchor	0.667	0.833	0.333
ant	0.667	0.833	1
barrel	1	1	0.857
bass	0.750	0.875	1
beaver	0.714	0.857	0.857
binocular	1	1	1
bonsai	1	1	1
brain	1	0.933	1
brontosaurus	0.667	0.667	1
buddha	1	1	1
butterfly	1	0.857	0.857
camera	1	0.857	1
cannon	1	0.833	0.833
car_side	1	1	1
ceiling_fan	0.857	1	1
cellphone	1	0.889	1
chair	0.800	0.800	0.900
chandelier	0.938	1	0.938
cougar_body	0.857	0.714	0.714
cougar_face	1	1	0.900
crab	0.909	0.545	0.818
crayfish	0.600	0.500	0.800
crocodile	0.375	0.875	0.750
crocodile_head	0.857	0.571	0.429
cup	1	0.875	0.875
dalmatian	1	1	1
dollar_bill	1	1	1
dolphin	1	0.900	0.900
dragonfly	0.900	1	1
electric_guitar	0.818	0.909	0.909
elephant	0.900	0.900	0.800
emu	0.625	1	1
euphonium	1	0.667	0.778
ewer	0.923	0.923	1
ferry	1	1	1
flamingo	1	0.900	1
flamingo_head	0.714	1	1
garfield	1	1	1
gerenuk	1	0.600	1
gramophone	1	0.750	0.750
grand_piano	0.867	0.933	1
hawksbill	0.800	0.933	1
headphone	0.857	1	1
hedgehog	0.875	1	1
helicopter	0.923	0.846	0.846
ibis	0.833	0.750	0.917
inline_skate	1	1	1
joshua_tree	0.889	1	1
kangaroo	0.846	1	1
ketch	0.647	0.765	0.941
lamp	0.889	1	1
laptop	1	1	1
llama	0.818	1	0.818
lobster	0.833	0.333	1
lotus	1	0.900	0.800
mandolin	0.833	1	0.833
mayfly	1	0.833	0.833
menorah	1	0.846	1
metronome	1	1	1
minaret	1	1	1
nautilus	0.889	1	1
octopus	1	0.600	0.600
okapi	1	1	0.833
pagoda	0.857	1	1
panda	1	1	0.833
pigeon	1	0.833	0.833
pizza	1	0.875	0.875
platypus	0.800	1	0.600
pyramid	0.875	1	1
revolver	1	0.846	0.923
rhino	1	1	1
rooster	0.625	0.750	1
saxophone	0.833	1	1
schooner	0.889	0.778	0.778
scissors	0.667	0.833	1
scorpion	0.923	0.846	0.769
sea_horse	1	1	1
snoopy	0.8	1	1
soccer_ball	1	1	1
stapler	0.714	1	1
starfish	1	0.769	0.923
stegosaurus	0.889	0.778	0.889
stop_sign	1	1	1
strawberry	1	1	1
sunflower	1	1	1
tick	1	0.857	0.857
trilobite	1	1	1
umbrella	1	1	0.818
watch	1	1	0.972
water_lilly	0.167	0.833	0.333
wheelchair	0.889	0.889	0.556
wild_cat	0.600	0.400	0.800
windsor_chair	0.889	1	1
wrench	0.333	0.500	0.833
yin_yang	1	1	1



Confusion Matrix for ResNet 18 – Top 10 hardest classes

Confusion Matrix for EfficienNetB1 – Top 10 hardest classes

Confusion Matrix for ViT-B/16 – Top 10 hardest classes

Figure 5. Confusion Matrix for Top 10 hardest classes for model to correctly identify