# Artistic Style Classification with Convolutional Neural Networks , Transfer Learning and Explainability Analysis

Feyza Yildiz

## 1. Introduction

This project adresses the task of artistic style classification using deep learning approaches. The objective is to classify artwork images from the WikiArt dataset into six distinct artistic styles, comparing the performance of a simple convolutional neural network (CNN) trained from scratch against a pretrained ResNet-50 model using transfer learning. Additionally, a Vision Transformer (ViT) model is being trained to explore the effectiveness of attention-based architectures versus traditional CNNs.

The project systematically applies techniques from the course lectures, specifically:

- Unit 5 (Regularization): Data augmentation, dropout, and early stopping
- Unit 6 (Optimization): Adam optimizer, learning rate scheduling, and training strategies
- Unit 7 (Convolutional Neural Networks): CNN architectures, residual connections, transfer learning and visualization techniques
- Unit 9 (Transformers): Vision Transformer architecture and self-attention mechanisms

## 2. Lecture Contents

### 2.1 Regularization (Unit 5)

Regularization addresses the core trade-off in machine learning between model complexity and generalization. Models with excessive capacity relative to the available data tend to overfit by memorizing noise, while overly simple models may underfit and fail to capture meaningful structure. Regularization techniques control this balance by constraining the learning process.

Parameter-based methods add penalties to the loss function to discourage large weights, with L2 regularization promoting smooth solutions and L1 encouraging sparsity, corresponding to Gaussian and Laplacian priors in a MAP framework. Early stopping provides implicit regularization by limiting training time based on validation performance. Ensemble-inspired approaches further improve generalization: bagging reduces variance across models, dropout approximates this effect within a single network, and data augmentation increases the effective dataset size through label-preserving transformations. In practice, combining these techniques typically yields the most robust performance.

### 2.2 Optimization (Unit 6)

Training deep neural networks requires optimizing highly non-convex loss functions with saddle points and flat regions, making optimization strategy and learning rate selection critical. Stochastic gradient descent (SGD) provides efficient, noisy updates, while momentum accelerates convergence by leveraging gradient history. Adaptive optimizers such as Adam further improve training by adjusting per-parameter learning rates and applying bias correction, making them robust and easy to tune.

Learning rate schedules enhance convergence by enabling rapid initial learning followed by stable refinement. Additional techniques—including batch normalization, residual connections, and large-scale pretraining—further stabilize optimization and enable effective training of deep architectures.

### 2.3 Convolutional Neural Networks (Unit 7)

Convolutional Neural Networks (CNNs) efficiently process image data by exploiting spatial structure through local connectivity and weight sharing. Convolutional layers apply small filters to extract local patterns while preserving translation equivariance, and spatial resolution is reduced via pooling or strided convolutions to improve invariance and efficiency. Modern CNNs emphasize deep, hierarchical feature learning, with architectures such as VGG, GoogLeNet, and ResNet, where residual connections enable very deep networks by mitigating vanishing gradients.

Transfer learning is widely used, with models pretrained on large datasets like ImageNet providing robust visual features that can be fine-tuned for downstream tasks. To interpret CNN predictions, visualization methods such as saliency maps, deconvolution, and maximally activating patches are used. Among these, Grad-CAM is particularly effective, producing class-specific localization maps that highlight image regions most influential for a model's decision.

### 2.4 Transformers (Unit 9)

Attention mechanisms address limitations of recurrent models by enabling dynamic weighting of relevant input elements. Self-attention allows each element to attend to all others, while multi-head attention captures complementary relationships in parallel subspaces. These components form the core of the transformer architecture, together with feed-forward layers, residual connections, normalization, and positional encodings to handle order information.

Transformers extend naturally to vision through Vision Transformers (ViTs), which represent images as sequences of patch embeddings. When pretrained on large datasets, ViTs often outperform convolutional models with similar computational budgets by leveraging global receptive fields and strong scalability, albeit at higher memory cost.

## 3. Task and Dataset

### 3.1 Task Description

The goal of this project is a multi-class image classification task in which an input artwork image is assigned to one of six artistic style categories. The problem is formulated as supervised learning, where RGB images of varying original resolutions are resized to 224×224 pixels and mapped to one of six style labels (IDs: 12, 21, 23, 9, 20, 24). Model performance is evaluated by classification accuracy on a held-out test set.

Artistic style classification is inherently challenging due to significant overlap between styles in terms of visual characteristics such as color usage and brushwork, as well as high variability within each style caused by differences across artists and time periods. Consequently, the task requires learning high-level semantic representations rather than relying solely on low-level visual cues.

### 3.2 Dataset Construction

The dataset is derived from the *huggan/wikiart* collection available on HuggingFace, which contains artwork images annotated with artistic style labels. An initial analysis of the dataset revealed a strongly imbalanced distribution of styles, with sample counts decreasing rapidly and several styles containing only a limited number of images. To ensure reliable training and fair evaluation, six styles with sufficient data availability were selected (IDs: 12, 21, 23, 9, 20, 24) which corresponds to impressionism, realism, romanticism, expressionism, post impressionism and symbolism respectively.

To maintain class balance and accommodate computational constraints, the number of images per style was capped at 2,000. This choice was particularly motivated by the use of Google Colab without GPU acceleration, which required limiting dataset size while preserving sufficient intra-class diversity. The resulting dataset contains approximately 12,000 images in total.

Images are streamed from HuggingFace, filtered by style label, and stored locally in JPEG format with associated JSON metadata. A fixed random seed is used to split the dataset into training (70%), validation (15%), and test (15%) sets, ensuring reproducibility and unbiased evaluation across all selected styles.

# 4. Data preprocessing

## 4.1 Image Preprocessing

All images were subjected to a standardized preprocessing pipeline to ensure compatibility with the chosen model architectures and stable training behavior. Specifically, images were resized to a fixed resolution of 224×224 pixels, which corresponds to the input size expected by widely used convolutional and transformer-based architectures such as VGG, ResNet, and Vision Transformers. This resizing ensures consistent spatial dimensions across the dataset and allows direct reuse of pretrained weights without architectural modifications.

In addition, pixel values were normalized using the ImageNet channel-wise mean and standard deviation. Since the pretrained ResNet-50 and ViT models were originally trained on ImageNet using these normalization statistics, applying the same normalization aligns the input distribution of the target dataset with that of the pretraining data. This step is crucial for effective transfer learning, as it prevents distribution shifts that could otherwise degrade convergence and performance during fine-tuning.

## 4.2 Data Augmentation

To improve generalization and reduce overfitting, data augmentation was applied exclusively during training, following the regularization principles discussed in Unit 5. In particular, random horizontal flipping with probability 0.5 was used to exploit the horizontal symmetry present in many artworks, encouraging the model to learn representations that are invariant to left–right orientation. In addition, color jittering was applied by randomly adjusting brightness, contrast, and saturation within a limited range (0.2), increasing visual diversity while preserving the semantic content of each image. As emphasized in the lectures, such geometric and appearance-based transformations effectively expand the training dataset without requiring additional labeled data and act as a strong form of regularization for image classification tasks.

During validation and testing, no data augmentation was applied. Images were only resized and normalized to ensure that performance evaluation remained unbiased and free from transformation-induced artifacts. This separation between training-time augmentation and evaluation-time preprocessing ensures that improvements in performance reflect genuine generalization rather than adaptation to artificial distortions.

# 5. Model Architectures

## 5.1. SimpleCNN: Baseline Architecture

As a baseline model, a simple convolutional neural network (SimpleCNN) was implemented to establish a reference point for evaluating the impact of architectural depth and pretraining in later experiments. The network takes a 224×224 RGB image as input and consists of three convolutional blocks followed by fully connected layers for classification.

Each convolutional block applies a 3×3 convolution with padding to preserve spatial dimensions, followed by a ReLU activation and a 2×2 max pooling operation. The number of channels increases progressively from 32 to 64 and then to 128, while the spatial resolution is reduced from 224×224 to 28×28. This design follows the principles discussed in Unit 7, where convolutional layers exploit weight sharing to efficiently capture local spatial patterns and maintain translation equivariance, while deeper layers learn increasingly abstract representations over larger receptive fields.

Max pooling is used after each convolutional layer to reduce spatial resolution and introduce local translation invariance without adding additional parameters. As the network depth increases, the receptive field of individual neurons grows, enabling the later layers to capture mid-level artistic features relevant for style classification.

After the final convolutional block, the feature maps are flattened into a 102,400-dimensional vector and passed through a fully connected layer with 256 hidden units. A dropout layer with probability 0.3 is applied at this stage to reduce overfitting by randomly deactivating neurons during training, thereby discouraging co-adaptation of features. As discussed in Unit 5, dropout acts as an implicit regularizer and can be interpreted as approximating an ensemble of many sub-networks.

The final output layer maps the learned representation to six class logits corresponding to the artistic styles. The model contains approximately 26 million parameters, with the majority concentrated in the first fully connected layer. Despite its simplicity, this architecture provides a controlled baseline that captures essential CNN properties while remaining computationally feasible, making it suitable for isolating the effects of more advanced architectural choices explored in subsequent models.

### 5.2. ResNet-50: Transfer Learning Architecture

ResNet-50 was used as a transfer learning architecture to evaluate the impact of deep pretrained representations. The model consists of 50 layers and was pretrained on the ImageNet dataset. For this task, the original classification layer was replaced with a new linear layer mapping the 2,048-dimensional feature vector to six artistic style classes. Training was performed using a frozen-backbone strategy, where all pretrained layers were kept fixed and only the final classification head was trained. Although the full network contains approximately 23.5 million parameters, only about 12,300 parameters were trainable, which significantly stabilizes optimization and reduces the risk of overfitting given the moderate size of the WikiArt subset.

The strength of ResNet lies in its residual connections, which introduce skip connections of the form $H(x)=F(x)+xH(x) = F(x) + xH(x)=F(x)+x$ and enable effective gradient flow in deep networks. This design allows ResNet-50 to learn hierarchical feature representations without suffering from vanishing gradients, as discussed in Unit 7.

By leveraging ImageNet pretraining, the model reuses generic visual features learned from a large-scale dataset, while adapting only the final layer to the target task. Additionally, the extensive use of batch normalization in ResNet contributes to stable behavior and implicit regularization. Overall, this architecture provides a strong and computationally efficient baseline for style classification under limited computational resources.

### 5.3. Vision Transformer (ViT): Attention-Based Architecture

For this study, we adopt a ViT-Base model with 16×16 patches (vit_base_patch16_224). The 224×224 input image is divided into a 14×14 grid of non-overlapping 16×16 patches, yielding 196 tokens. Each patch is linearly projected to a 768-dimensional embedding, and positional encodings are added to preserve spatial information, as discussed in Unit 9's section on patch-based representations, which emphasizes that treating images as sequences of patches enables the model to capture relationships across the entire image, similar to words in NLP.

The transformer encoder consists of 12 layers of multi-head self-attention. Following the formulation in Unit 9, query, key, and value vectors are computed from the patch embeddings, and attention weights are obtained through scaled dot-product attention. This allows each patch to attend to all other patches, modeling long-range dependencies without the convolutional inductive biases found in CNNs. Multi-head attention, also highlighted in Unit 9, enables each head to learn complementary relationships among patches, with the concatenated outputs projected to enrich the representation. Compared to CNNs, as noted in Unit 9, transformers provide fully parallel computation, a global receptive field from the first layer, and flexibility in learning spatial relationships directly from the data.

For classification, the pretrained head is replaced with a linear layer mapping from 768 to 6 classes. The training strategy freezes the transformer encoder and updates only the classification head, leveraging the strong feature representations obtained from large-scale pretraining. Unit 9 emphasizes that such transfer learning is highly effective, allowing Vision Transformers to match or exceed the performance of CNNs like ResNet when pretrained on sufficiently large datasets.

## 6. Loss Function and Training Configuration

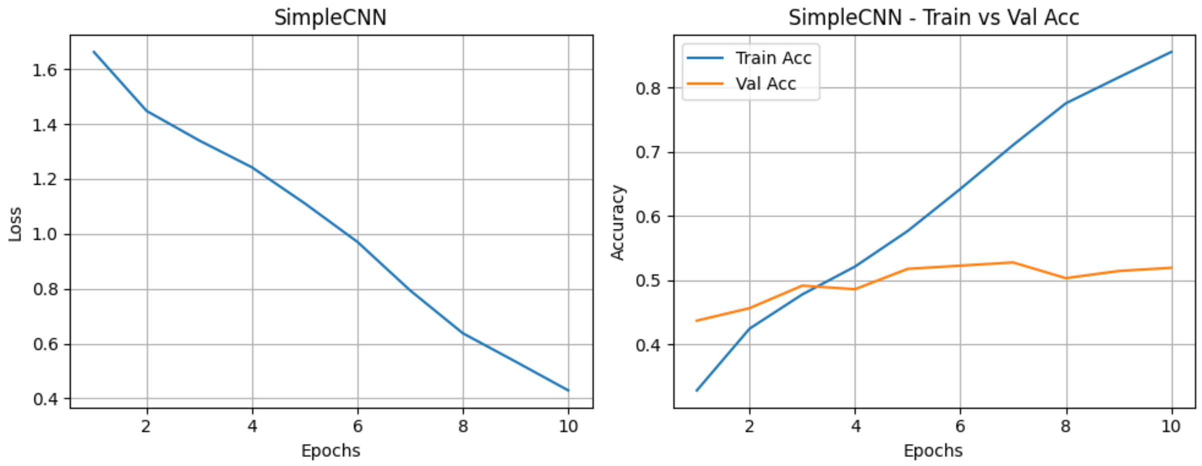
For the multi-class classification task, we use the standard cross-entropy loss, which compares the predicted class probability distribution with the true one-hot encoded labels. Cross-entropy can be interpreted as measuring the Kullback–Leibler divergence between the true and predicted distributions, making it a well-established and theoretically grounded objective for classification problems (Unit 6, Section 6.1).

Model optimization is performed using the Adam optimizer with an initial learning rate of 0.001. As discussed in Unit 6, Section 6.2, Adam combines momentum-based updates with adaptive per-parameter learning rates by maintaining running estimates of both the first and second moments of the gradients. Bias correction is applied during early training to improve stability. These properties make Adam particularly robust in practice, especially when gradients are noisy, sparse, or unevenly scaled, and reduce the need for extensive hyperparameter tuning.

Learning rate scheduling is applied differently depending on model size and training dynamics. For the smaller SimpleCNN, we use a ReduceLROnPlateau strategy that monitors validation accuracy and reduces the learning rate when performance stagnates. This approach aligns with Unit 6, Section 6.3, which highlights the importance of adaptive learning rate schedules for escaping plateaus and enabling finer-grained updates in later training stages. For larger pretrained models such as ResNet-50, we employ a cosine annealing schedule that smoothly decreases the learning rate over training epochs. Unit 6 notes that cosine annealing is particularly effective for pretrained networks, as it provides a stable and deterministic decay without abrupt changes that could disrupt convergence.

To mitigate overfitting, early stopping is applied with a patience of five epochs, restoring the model parameters corresponding to the best validation performance. As described in Unit 5, Section 5.2, early stopping acts as an implicit form of regularization by limiting effective model complexity through training duration. This technique is especially appropriate when validation data is available, training time is constrained, and overfitting risk is high, all of which apply in this project.

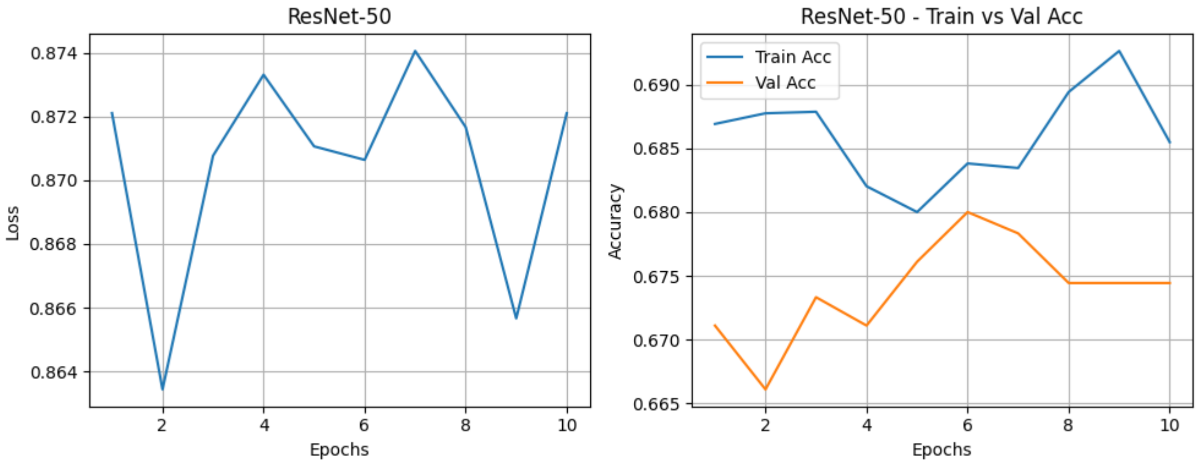
### 6.1 SimpleCNN Training Dynamics



During training, the loss decreased steadily from approximately 1.6 to 0.4 over 10 epochs, while the training accuracy rose from around 0.30 to 0.85. Validation accuracy, however, improved more slowly, increasing from 0.43 to 0.53 before plateauing around epoch 5. This large discrepancy between training and validation performance indicates overfitting, suggesting that the model effectively memorizes patterns in the training data that do not generalize to unseen examples, despite the use of regularization techniques such as dropout and data augmentation.

Unit 5 explains that overfitting commonly arises when model capacity exceeds the information content of the training data. In this case, the small dataset of roughly 8,400 training images is insufficient relative to the model's complexity (~26 million parameters). Additionally, the network's limited architectural depth, with only three convolutional layers, may restrict its ability to learn robust high-level feature hierarchies necessary for distinguishing subtle artistic styles, which often share overlapping visual characteristics. As discussed in Unit 5, effective regularization—such as early stopping, weight penalties, or data augmentation—can mitigate overfitting, but the observed validation plateau indicates that the model still relies heavily on memorized features rather than generalized representations.

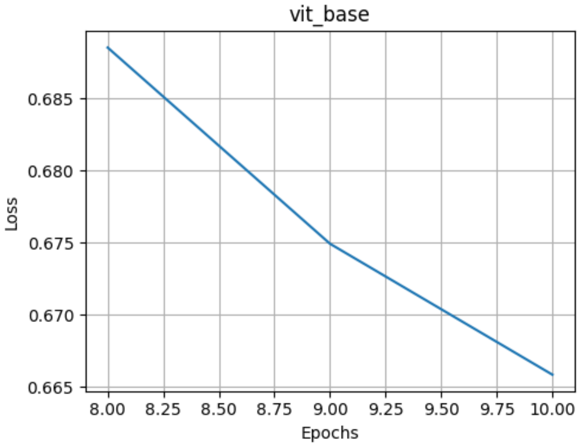
### 6.2 ResNet-50 Training Dynamics

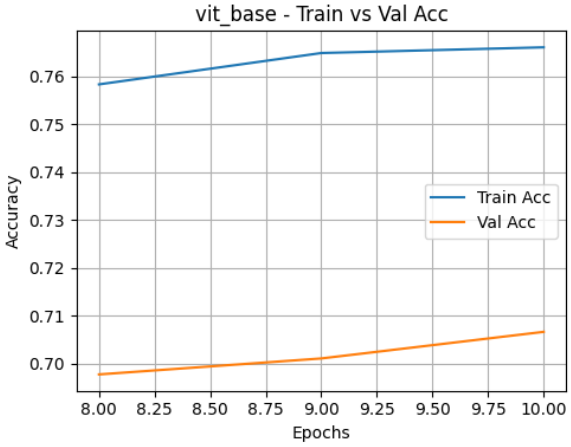


During training, the loss remained relatively stable around 0.87, while both training and validation accuracy plateaued at approximately 0.68–0.69. The near-identical training and validation metrics indicate minimal overfitting, reflecting several factors. First, the pretrained ViT backbone provides powerful generic feature representations from ImageNet, allowing the model to extract meaningful visual features without extensive task-specific learning. Second, the trainable capacity is intentionally limited, with only 12,000 parameters in the final classification head, which constrains the model's ability to memorize training data. Third, freezing the transformer encoder acts as a form of structural regularization, further reducing the risk of overfitting.

The model reaches this performance plateau quickly because the backbone already encodes robust visual features, and the 6-class classifier head is relatively simple to optimize. Any further improvements would likely require unfreezing deeper layers of the transformer to allow fine-tuning, which could increase accuracy but also introduce potential overfitting—making it a subject for future work. Overall, the training curves suggest that the chosen strategy successfully leverages pretrained features while maintaining generalization.

### 6.3 Vision Transformer (ViT) Training Dynamics





Over 10 training epochs, the Vision Transformer exhibited stable and smooth progression. The training loss decreased modestly from 0.688 to 0.666, while training accuracy improved from 0.76 to 0.77. Validation accuracy followed a similar trajectory, rising from 0.70 to 0.71, resulting in a minimal gap between training and validation metrics. This indicates that overfitting was effectively controlled despite the model's large overall capacity of 86 million parameters.

Several factors contributed to this stable behavior. The global self-attention mechanism of the transformer allows each patch to attend to all others, capturing long-range dependencies between visual styles, which is particularly important for distinguishing subtle differences in artistic content. Despite the model's large size, only 18.4K parameters in the trainable classification head were updated, and the backbone remained frozen, acting as structural regularization and constraining the model from memorizing training data. This setup mirrors the principles of effective transfer learning, where pretrained features from ImageNet provide strong inductive bias, enabling rapid convergence and robust generalization.

The Vision Transformer slightly outperformed the ResNet-50 baseline, validating the effectiveness of attention-based architectures for vision tasks. This observation aligns with Unit 9, which notes that "ViT generally outperforms ResNets with the same computational budget," highlighting the transformer's ability to leverage global feature interactions efficiently even when only a small fraction of the parameters is fine-tuned.

### 6.5 Test Set Performance

After training, the best models (selected by validation accuracy) are evaluated on the held-out test set:

| **Model** | **Test Accuracy** | **Test F1** | **Test Precision** | **Test Recall** | **Δ vs SimpleCNN** |
|---|---|---|---|---|---|
| SimpleCNN | **50.44%** | 50.32% | 50.58% | 50.44% | — |
| ResNet-50 | **66.89%** | 66.95% | 67.61% | 66.95% | **+16.45%** |
| ViT | **69.44%** | 69.39% | 69.56% | 69.50% | **+19.00%** |

The evaluation results demonstrate that the Vision Transformer achieves the highest overall performance, reaching 69.44% accuracy and surpassing ResNet-50 by 2.55 percentage points. Both pretrained models significantly outperform the SimpleCNN baseline, with ResNet-50 and ViT improving over SimpleCNN by 16.45% and 19.00%, respectively. The SimpleCNN model barely exceeds the random baseline of 16.67% for six classes, confirming that its limited depth and capacity are insufficient to capture the complex visual patterns required for artistic style classification.

Across all models, performance metrics such as F1 score, precision, and recall remain closely aligned, indicating consistent predictive quality across classes without systematic bias toward either high- or low-frequency categories. The modest yet meaningful 2.55% improvement of ViT over ResNet-50 suggests that the self-attention mechanism more effectively captures long-range dependencies and subtle patterns inherent in artistic styles than the local convolutional inductive bias of ResNets. Overall, these results highlight the advantage of attention-based architectures for tasks where global relational information is critical and confirm the value of leveraging pretrained features combined with minimal task-specific fine-tuning.

## 7. Grad-CAM Visualization Analysis

### 7.1 Grad-CAM Implementation

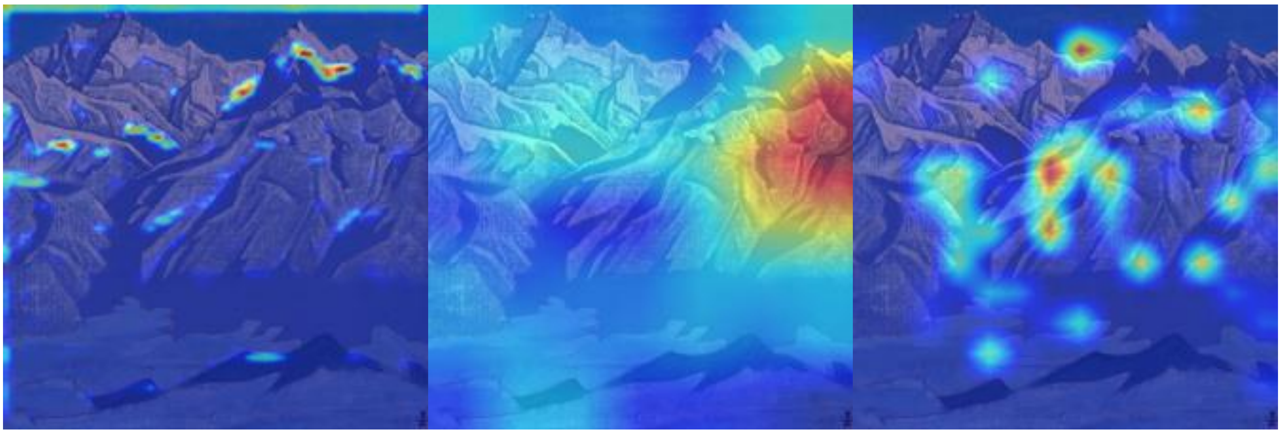
Grad-CAM provides class-specific visual explanations by highlighting the regions of an input image that most influence the model's prediction. The method works by first performing a forward pass through the network to obtain the predicted score for the target class. Then, a backward pass computes the gradients of this score with respect to the feature maps of a selected convolutional layer. These gradients are aggregated to determine the relative importance of each feature map for the target class. The final Grad-CAM heatmap is obtained by combining the weighted feature maps and applying a rectified linear operation to emphasize positive contributions. This heatmap is then resized to match the input image resolution and overlaid on the original image, visually highlighting the regions that were most influential for the model's decision.

```
target_layers = [model.conv3]  # For SimpleCNN: last conv layer

target_layers = [model.layer4]  # For ResNet-50: last residual block

target_layers = [model.blocks[-1].norm1]  # For ViT: Last transformer block
```

### 7.2 Grad-CAM Results

Experimental setup: 10 random test images; 3 representative cases shown below.

**Case 1: All models Correct (Style: Symbolism)**



(a) SimpleCNN (left): Attention scatters across multiple isolated regions - mountain peaks, color patches, and high-contrast edges. The fragmented heatmap with discrete circular hotspots indicates reliance on local texture features and color discontinuities without compositional understanding. This aligns with Unit 7's observation that shallow CNNs learn "simple gradient/Gabor filters."

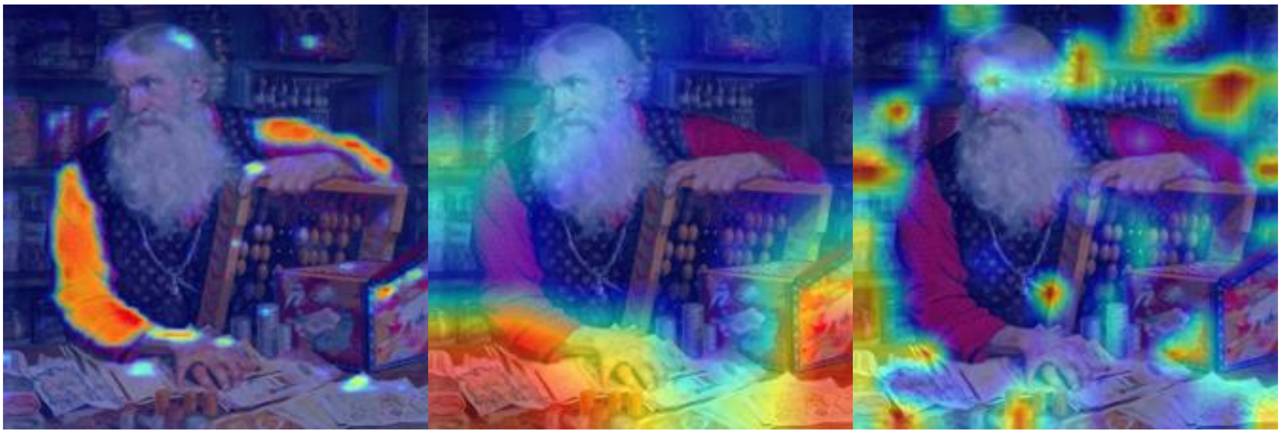
(b) ResNet-50 (center): Attention concentrates along coherent spatial structures - the diagonal mountain ridge, horizon line, and depth transitions. The spatially continuous heatmap demonstrates that ResNet-50 recognizes mid-level semantic features (objects, spatial relationships) rather than isolated textures. Residual connections enable learning these meaningful compositional elements.

(c) ViT (right): Attention distributes globally across the entire canvas, with smooth gradients rather than discrete hotspots. This holistic pattern reflects ViT's self-attention mechanism capturing long-range dependencies, color harmony, and overall compositional balance - the defining characteristics of artistic style. ViT understands the scene as an integrated whole rather than a collection of local features.

While all models achieve correct classification, their decision-making processes differ architecturally. SimpleCNN relies on low-level cues, ResNet-50 on structural patterns, and ViT on global composition. This validates the lecture's claim that deeper networks learn hierarchical abstractions, with transformers achieving the most holistic understanding.

**Case 2: ViT Outperforms CNNs (Style: Realism)**

SimpleCNN Prediction: Style 9 | ResNet-50 Prediction: Style 20 | ViT Prediction: Style 21



(a) SimpleCNN (left): Attention scatters across isolated color regions - the red book, bright clothing patches, and random high-saturation areas. The model exhibits no semantic understanding of the portrait subject or artistic composition. By fixating on local color textures resembling Style 9's training examples, SimpleCNN misclassifies the artwork. This validates that shallow CNNs remain trapped at low-level feature detection (Unit 7: "First layer learns simple gradient/Gabor filters").

(b) ResNet-50 (center): Attention concentrates on the red book/papers in the lower-left, with moderate focus on the figure's clothing. While ResNet-50 successfully recognizes semantic objects (book, person, garments), it over-weights the prominent foreground object at the expense of the overall composition. The heavy focus on the book's color and texture leads to confusion with Style 20, which likely shares similar object-level features. ResNet-50 captures what is depicted but misses how Style 21 characteristically renders scholarly portraits.

(c) ViT (right): Attention distributes globally across the figure, integrating the arm position, book placement, facial region, and background into a unified compositional understanding. The smooth, connected heatmap demonstrates ViT's ability to model long-range dependencies: the relationship between the subject's pose, the reading gesture, color harmony (blue garment + red book), and spatial layout. This holistic attention captures the artistic style's defining characteristics - composition, lighting, brushwork, and thematic treatment - enabling correct classification.

## 8. Conclusion and Limitations

This project shows that architecture design and transfer learning matter far more than raw parameter count. Despite having fewer trainable parameters, both ResNet-50 and ViT significantly outperform the SimpleCNN, confirming the lecture principle that depth, residual connections, and pretrained representations dominate performance. The Vision Transformer achieves the best result (69.44%), outperforming ResNet-50 (66.89%) through global self-attention, which captures long-range dependencies crucial for artistic style recognition.

Grad-CAM visualizations reinforce these findings: SimpleCNN focuses on low-level textures, ResNet-50 attends to semantic structures, and ViT integrates attention across entire compositions. These differences show that performance gains are rooted in fundamental architectural reasoning, not just statistical variance. Overall, techniques from Units 5–7—data augmentation, early stopping, learning rate scheduling, and transfer learning—were essential for achieving competitive results under limited computational resources.

The project faced significant computational constraints, limiting training to small batch sizes, a maximum of 10 epochs, and minimal hyperparameter tuning. As a result, SimpleCNN likely did not fully converge, and more extensive training could reduce its performance gap. Similarly, freezing the backbones of ResNet-50 and ViT prevented domain-specific fine-tuning, which would likely yield further gains if deeper layers were unfrozen with lower learning rates.

Dataset size was another limiting factor: only 8,400 training images were available, which is insufficient for training a 26M-parameter CNN from scratch. Pretrained models mitigate this issue, highlighting the importance of transfer learning in small-data regimes. Grad-CAM also provides only coarse interpretability, especially for ViT, where direct attention visualization would offer clearer insights