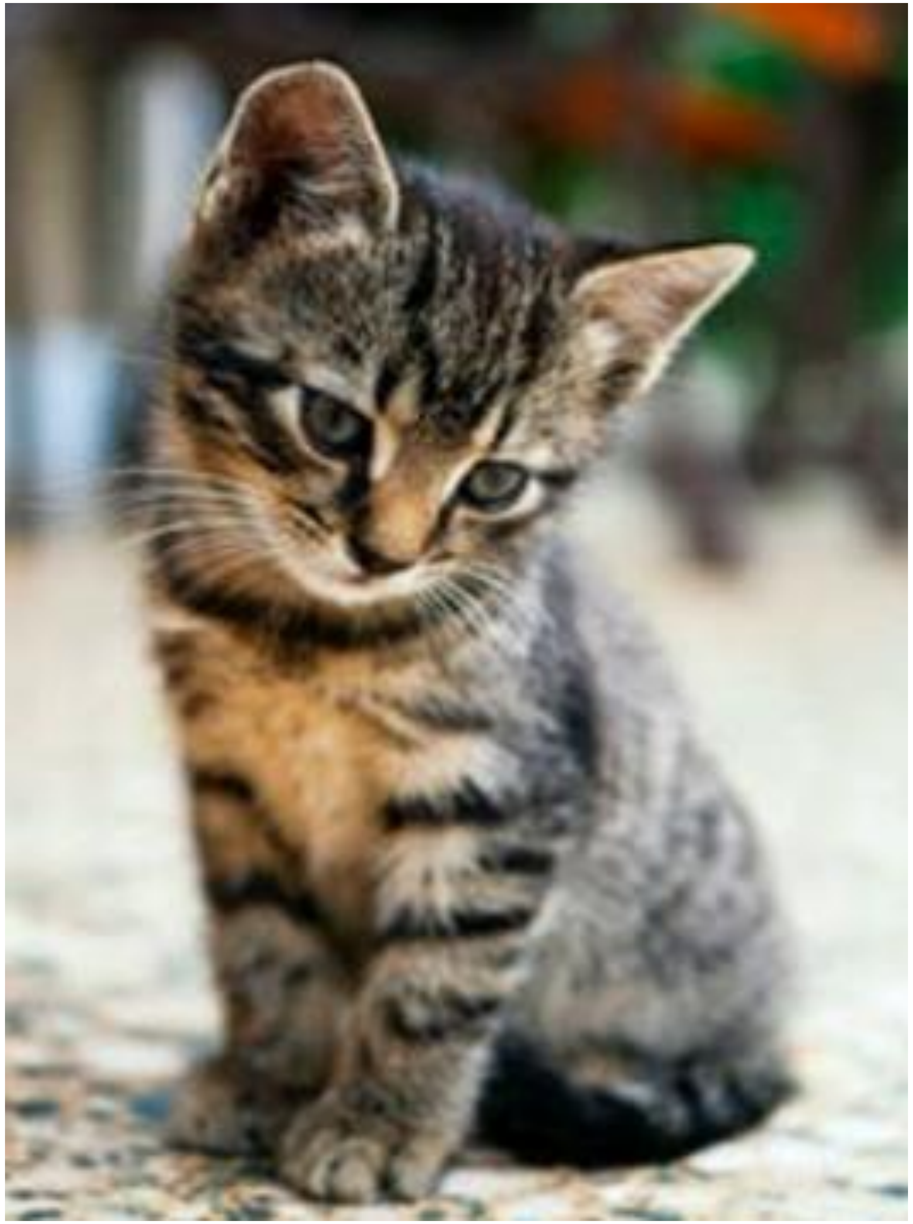
Art Style Classification using Computer Vision

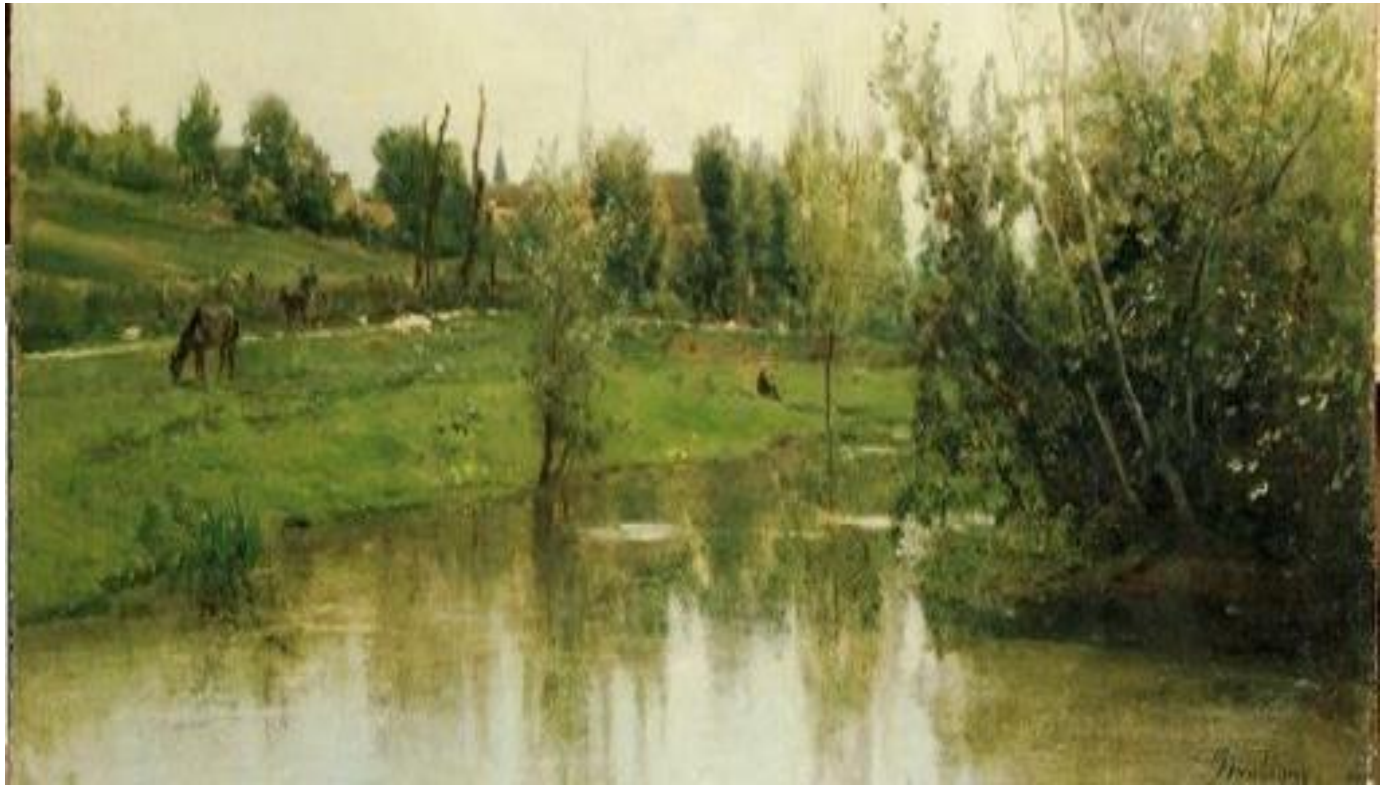
RSM8421 - Team 1:

Bokai Lai, Danish Siddiqui, Keqing Wang, Gabrielle Li , & Alex Griffith



Is this a Cat or a Dog?

Is this a Cat or a Kitten?



What Art Style is this painting?

Introduction

Problem

- Huge digital art collections need consistent style labeling
- Manual labeling is slow, subjective, and inconsistent
- Even experts disagree on style boundaries

Goal

- Build deep-learning models that correctly classify artworks into art movements and evaluate their effectiveness

Why This Matters

- Organizing digital art collections
- Improving recommendation and search tools
- Supporting valuation, provenance, and restoration tasks

Background

Why Art Style Classification is Hard

- Visual differences between movements are subtle
- Style cues come from texture, palette, brushwork
- Objects don't define style, stylistic patterns do
- WikiArt has label noise and ambiguous example

Deep Learning Context

CNNs: capture local textures and edges.

Transfer Learning: leverages pretrained visual knowledge for small datasets.

Vision Transformers (ViT): model global relationships through attention.

Dataset Overview

Description

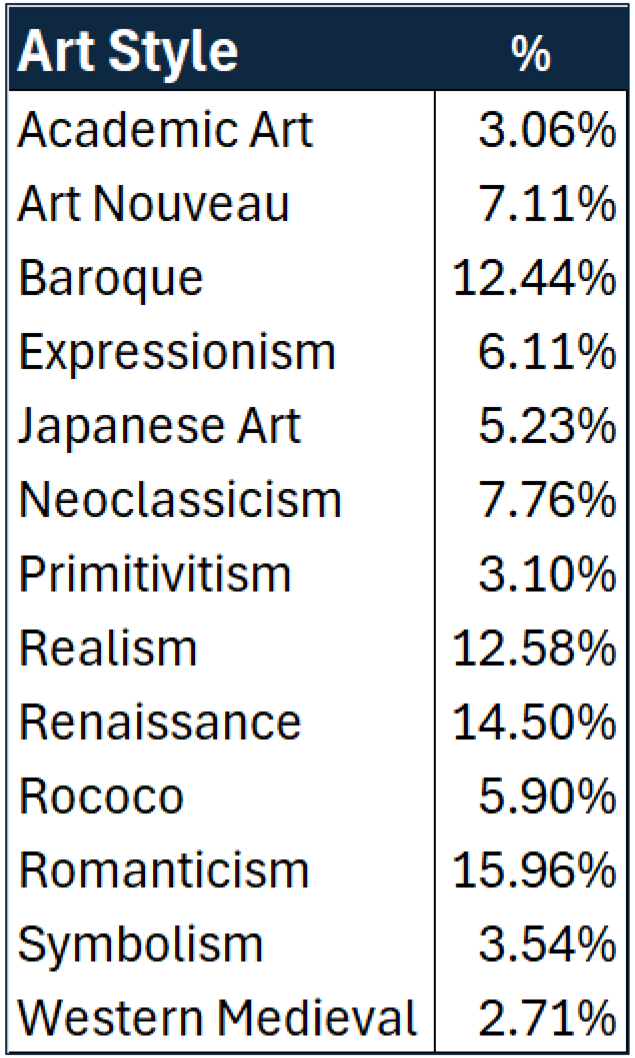
- **42.5k WikiArt images** across **13 art movements**, with some categories merged for coverage.
- Each Art movement contains at least **1150 images**
- **Noisy labels expected**, since movements overlap and some artworks can fit multiple styles.

Data Handling

- Normalized all images to 512x512
- Downsampled all images to 224 x 224
- Removed Duplicates

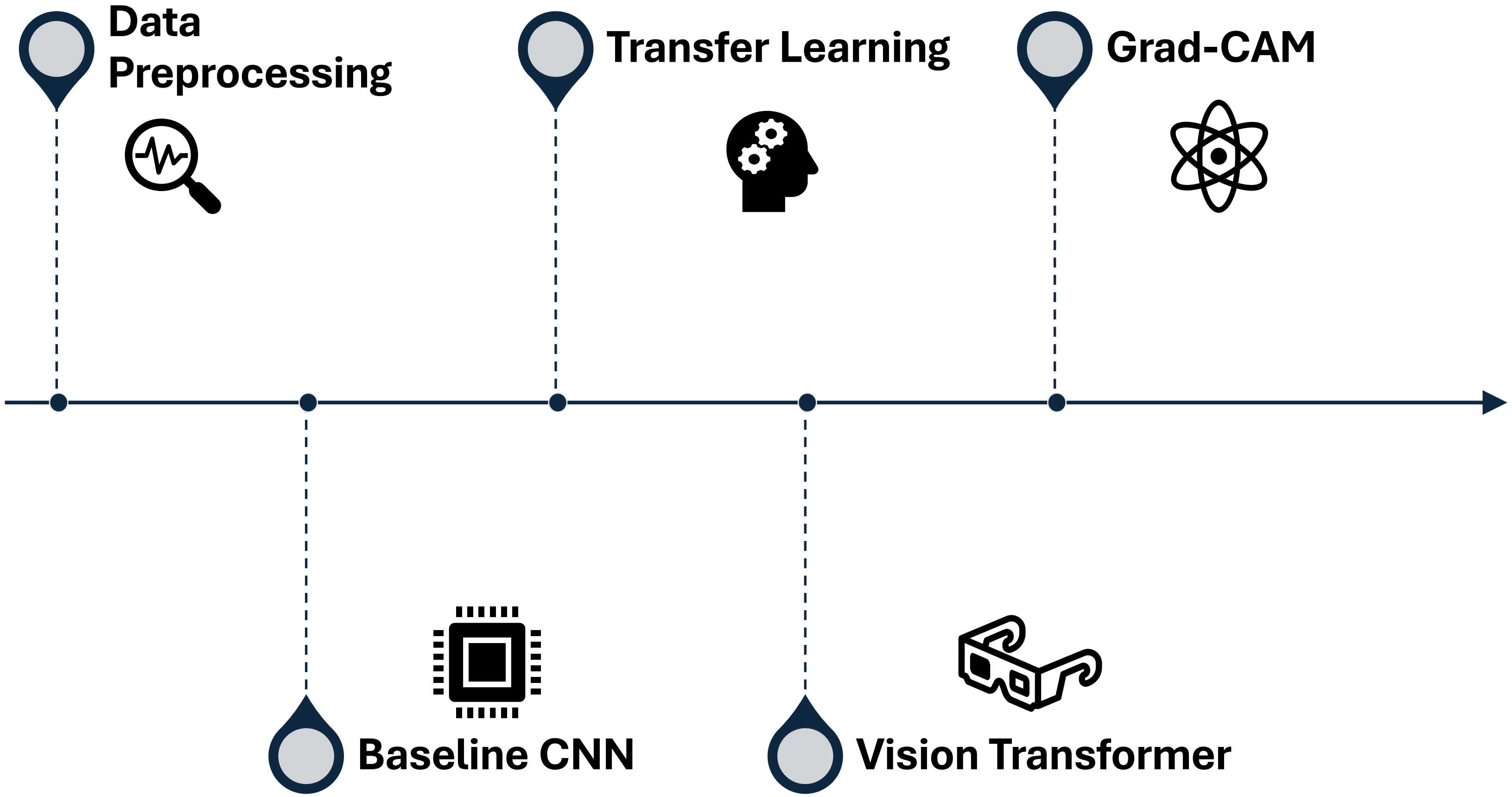
Test/Train Split

- 70% Training/15% Validation/15% Test



Art Style	%
Academic Art	3.06%
Art Nouveau	7.11%
Baroque	12.44%
Expressionism	6.11%
Japanese Art	5.23%
Neoclassicism	7.76%
Primitivitism	3.10%
Realism	12.58%
Renaissance	14.50%
Rococo	5.90%
Romanticism	15.96%
Symbolism	3.54%
Western Medieval	2.71%

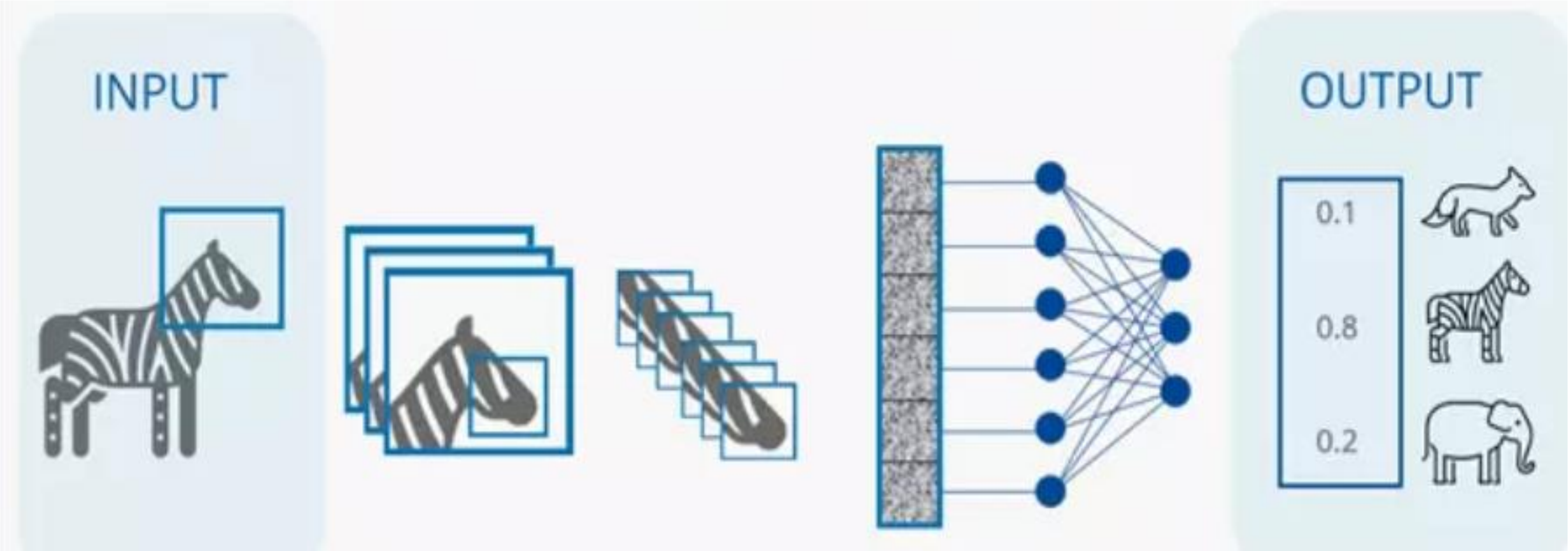
Project Workflow Overview



Convolutional Neural Networks(CNN)

How it works

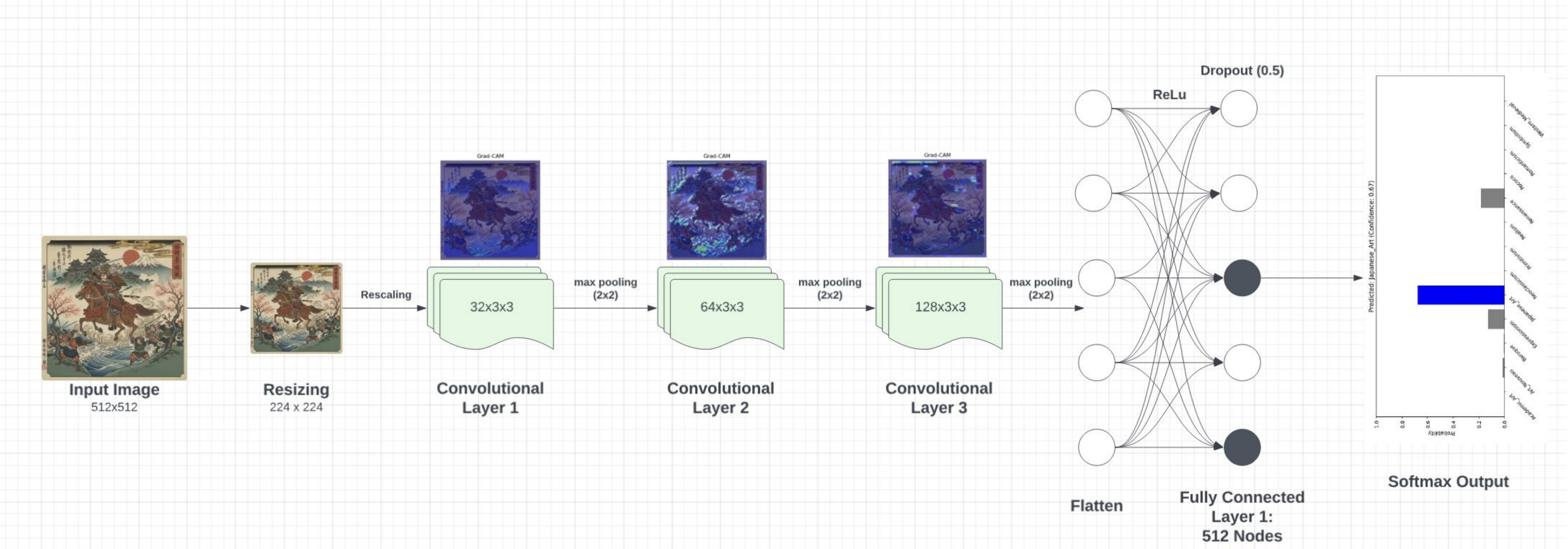
- The model trains on local features found in images and learns to classify them.
- Reuses learned visual features and adapts them to our art dataset



Why CNN?

Pros	Cons
• **Great at capturing local textures and brushwork**, which are key for style recognition.	**Limited global awareness**, so they may miss overall composition.
• **Computationally Efficient**, especially on moderate-sized datasets.	**Struggles with long-range relationships** beyond local texture patterns.
• **Reliable and well-validated**, with strong performance across image tasks.	**Performance can plateau** on complex tasks without deeper or more advanced architectures.

Model Architecture: Baseline CNN



01 Design Choices Made for the Model

- The model initialized weights based on ImageNet.
- The input image was downscaled to 224x224: Computational hurdles and denoising images.
- "sparse_categorical_crossentropy" loss function is ideal for softmax classification

02 Model Analytics

- Total Params: **44,402,765** ! (Mostly due to final FC layer)
- Training time: **237 Minutes**
- Whilst model accuracy is better than a random guess (1/13 = 0.07), it is still worth seeing if the accuracy can be improved.

Baseline CNN: Experiments and Results

Baseline CNN			
Art Style	**Precision**	**Recall**	**F1-Score**
Academic Art	0.76	0.65	0.70
Art Nouveau	0.78	0.74	0.76
Baroque	0.75	0.80	0.78
Expressionism	0.73	0.73	0.73
Japanese Art	0.80	0.85	0.82
Neoclassicism	0.81	0.78	0.79
Primitivitism	0.86	0.77	0.81
Realism	0.77	0.75	0.76
Renaissance	0.81	0.80	0.80
Rococo	0.84	0.74	0.78
Romanticism	0.73	0.83	0.79
Symbolism	0.77	0.63	0.69
Western Medieval	0.87	0.78	0.82
Accuracy			0.78
Macro Avg	0.79	0.76	0.77
Weighted Avg	0.78	0.78	0.78

Model Summary

- The CNN model performs relatively well with a test accuracy of **78%.** The model performed best with Japanese Art and Western Medieval Art and worst with Academic and Symbolism art
- Primivitism and Rococo had the highest precision, while Japanese Art and Romanticism had the highest recall.

Limitations to the Model

- maxpooling() after every layer was shrinking the feature map significantly (to 26x26), which shows up as lower apparent activation in the final convolutional layer.
- The model was disproportionately relying on the fully connected layer due to the high number of nodes.

Takeaways

- For image classification, it is better to use models that have more layers, as opposed to fewer layers. Also, shrinking the feature map significantly would lead to the network losing vital information.

Transfer Learning

How it works?

Training a deep network like InceptionV3 from scratch would require millions of images.
It benefits us by reusing the pretrained filters (such as textures, curvature) on a pretrained deep network built from scratch and on millions of images.

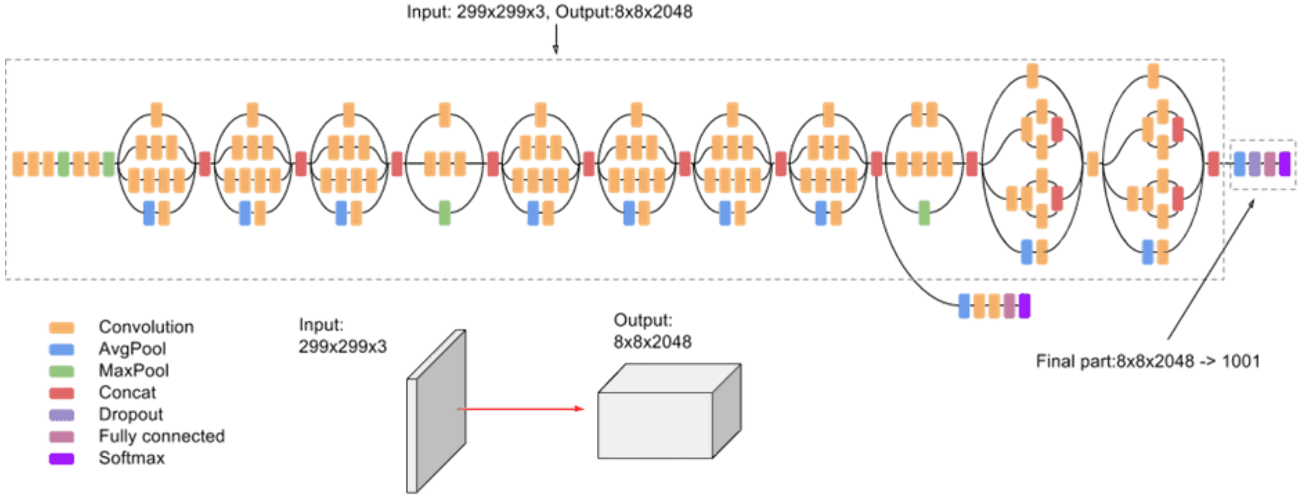
Why Transfer Learning?

Pros

- Increase accuracy with small art dataset and label noises, without huge data required
- Forster the learning pattern faster and training process more reliable

Cons

- Pretrained features may not match the target domain, which can limit performance
- Unfreezing layers may trigger memorization on the WikiArt images, and overfitting risk
- Hard for distinguishing the direction on the movement as pretrained model purpose differs



Model Architecture: Transfer Learning

01 Backbone: InceptionV3 (ImageNet)

- InceptionV3(weights="imagenet", include_top=False) used as the feature extractor
- 224 x 224 x 3 inputs size with 200 frozen layer and all BatchNormalization layer frozen
- Training Set up - using sparse_categorical_crossentropy, epochs=20, valid-accuracy 0.65

02 Model Limitation and Improvement Direction

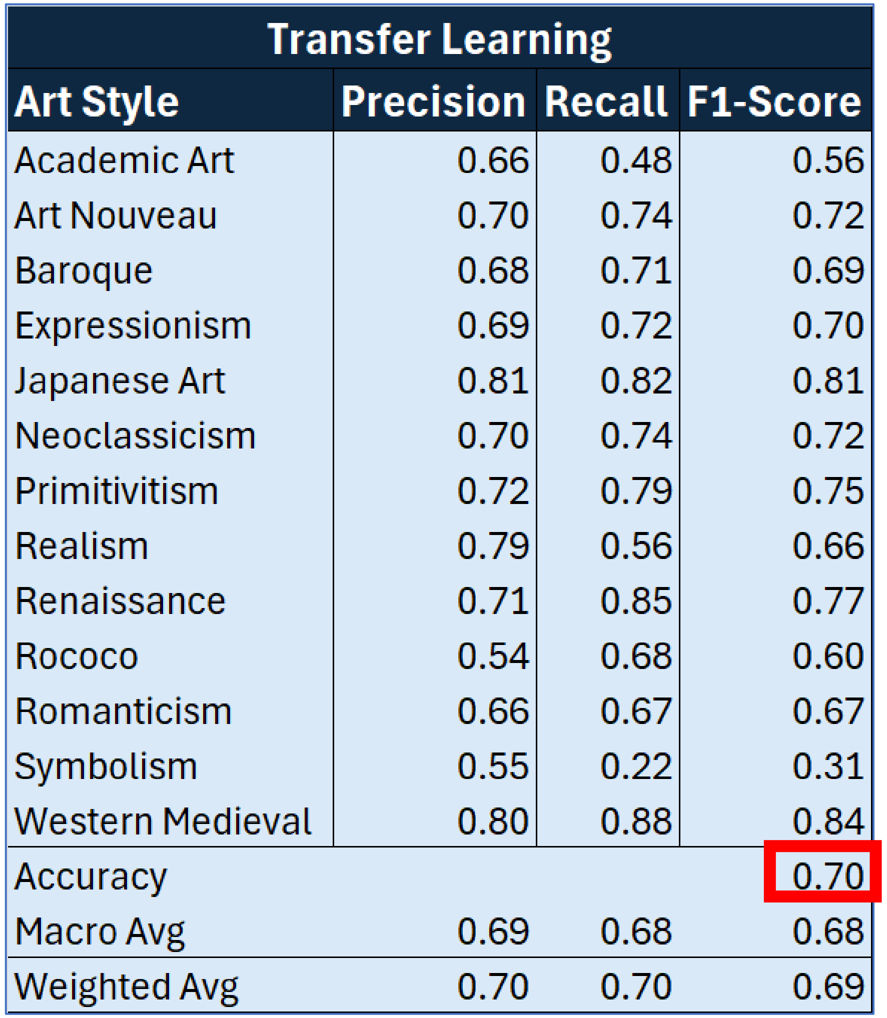
- **Explore alternative backbones:** Try DenseNet121, VGG19, or lighter architectures to adjust parameter count and complexity.
- **Pooling Refinement:** Replace **GlobalAveragePooling** with a Flatten layer when compute allows, increasing feature detail and improving classification accuracy.

03 Why InceptionV3?

Why InceptionV3? (Design Choice)

- **Strong, production-tested CNN backbone** with high ImageNet accuracy
- **Rich pretrained filters** specialized in detecting edges, colors, textures, and curvature
- **Multi-scale Inception modules** extract meaningful patterns from complex artworks, capturing fine brushstrokes and composition
- **Deep, expressive visual representation**, giving the model a strong starting point for art-style differentiation
- Original fully connected layers and softmax head are **replaced with a custom classifier**, enabling transfer learning on our 13-style dataset
- **Improved** our style-classification accuracy from **~50% (scratch CNN)** to **~65%**, demonstrating effective feature transfer

Experiments & Results: Transfer Learning



Transfer Learning			
Art Style	**Precision**	**Recall**	**F1-Score**
Academic Art	0.66	0.48	0.56
Art Nouveau	0.70	0.74	0.72
Baroque	0.68	0.71	0.69
Expressionism	0.69	0.72	0.70
Japanese Art	0.81	0.82	0.81
Neoclassicism	0.70	0.74	0.72
Primitivitism	0.72	0.79	0.75
Realism	0.79	0.56	0.66
Renaissance	0.71	0.85	0.77
Rococo	0.54	0.68	0.60
Romanticism	0.66	0.67	0.67
Symbolism	0.55	0.22	0.31
Western Medieval	0.80	0.88	0.84
Accuracy			0.70
Macro Avg	0.69	0.68	0.68
Weighted Avg	0.70	0.70	0.69

Key Observations

- Train of 90%; Test Accuracy averaging 70%
- Most major art movements achieve f1 scores in the 0.65 to 0.80 range, indicating solid generalization
- Model performance is capped by the pretrained ImageNet filter rather than WikiArt specific field

Why it Underperforms

- Backbone frozen → model cannot fully adapt ImageNet features to art-specific textures, brushstrokes, or color patterns
- Class imbalance + label noise WikiArt makes the frozen model struggle to adjust to overlapping art movements
- Risk of underfitting because pretrained layers cannot learn new stylistic directions when frozen

Takeaways

- Partial fine-tuning of deep inception helps to improve the adoption of the artistic textures
- Unfreezing control altering could improve the model performance and rich textures diagnosis

Vision Transformer (ViT)

What is ViT?

A Vision Transformer applies the Transformer architecture—originally designed for NLP—to images. Instead of convolutions, ViT processes images as sequences of fixed-size patches.

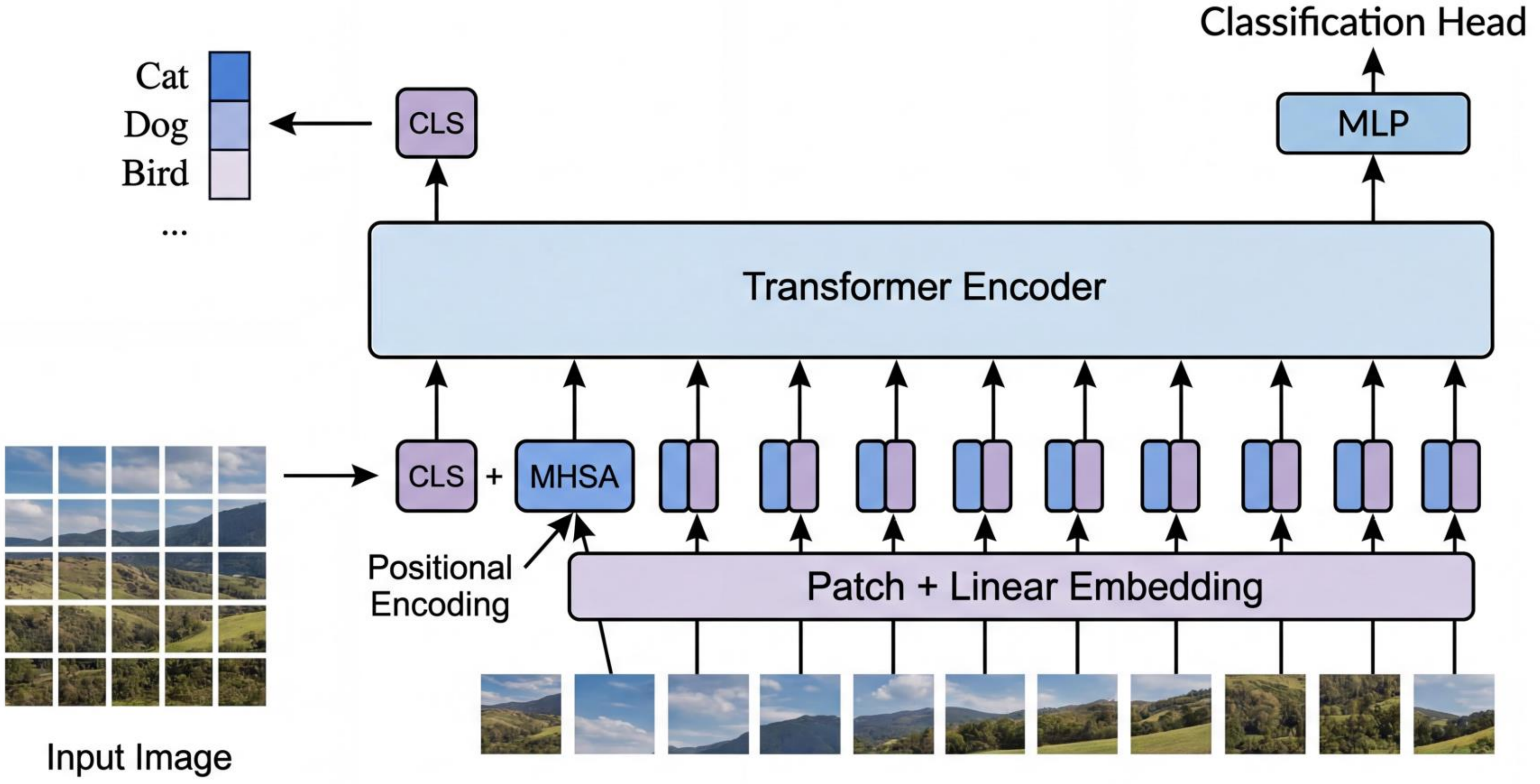
Why ViT? (Pros & Cons)

Pros

- Captures long-range dependencies better than CNNs
- Scales extremely well with large datasets
- Less inductive bias → more flexible feature learning

Cons

- Needs more training data than CNNs
- More computationally expensive
- Less local spatial bias → may underperform on smaller datasets without augmentation



Our ViT Architecture

ViT-Base pretrained on ImageNet, with a frozen backbone and a task-specific classification head for 13 art styles.

01 Backbone: Inception (ImageNet)

- ViTImageClassifier.from_preset("vit_base_patch16_224_imagenet")
- Input size 224 × 224, patch size 16 × 16
- 12 Transformer encoder blocks, 12 attention heads768-dim patch embeddings, 3072-dim MLP layers
- Uses a CLS token + final dense layer to output 13 classes

02 Data Pipeline & Preprocessing

- tf.data pipeline: load → resize → batch → prefetch
- Custom preprocessing: x = tf.cast(x, tf.float32) / 255.0
- Data augmentation applied in prepare_dataset():
    - RandomFlip
    - RandomRotation
    - RandomZoom
- Set preprocessor=None in ViTImageClassifier (avoids double preprocessing since we already normalize & augment)

03 Why keras_hub ViT (Design Choice)

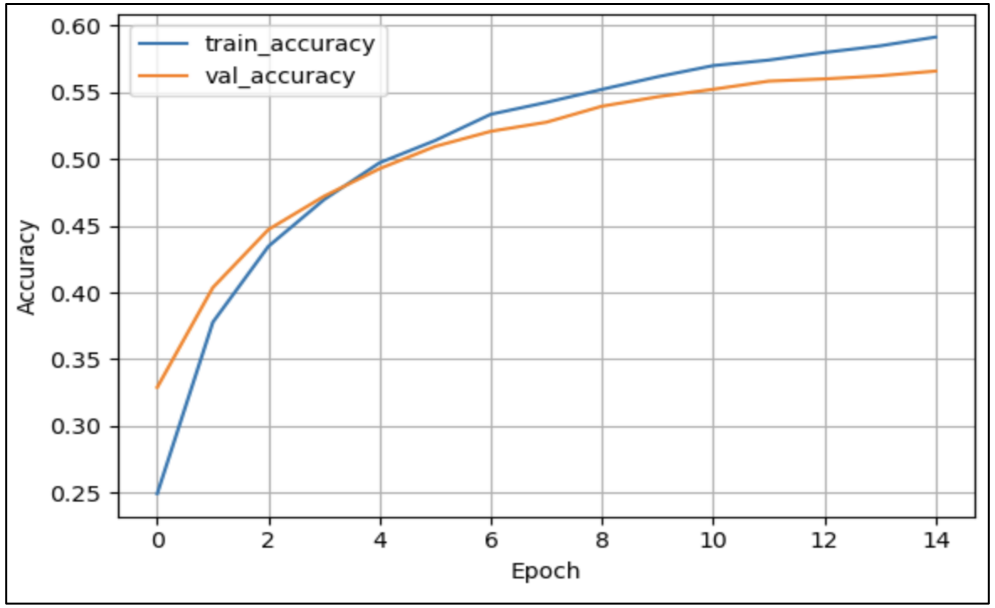
Why use keras_hub.ViTImageClassifier?

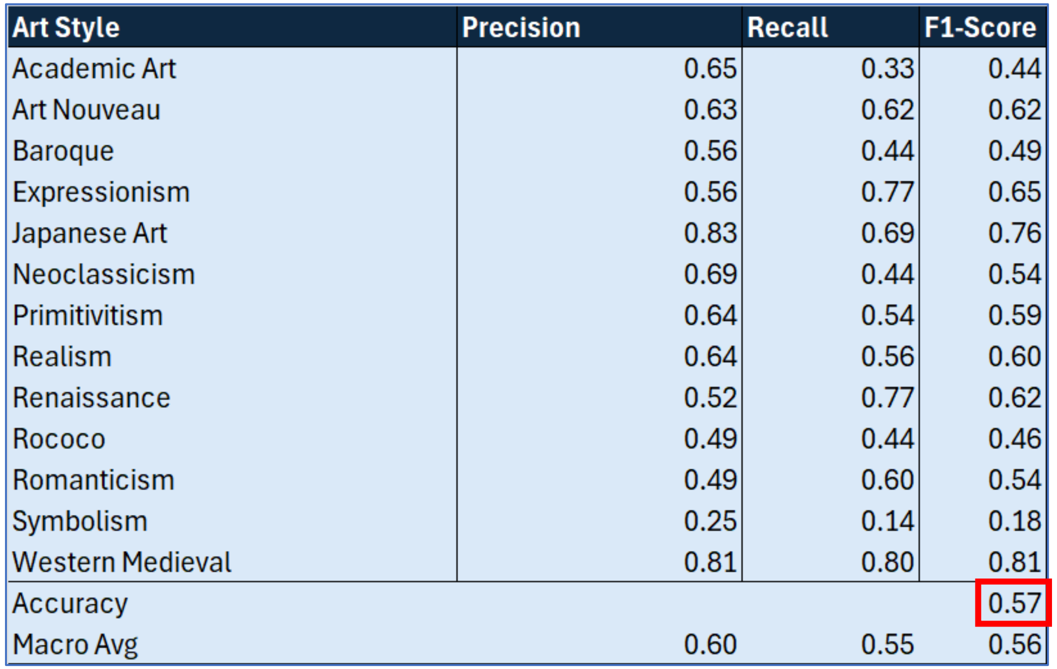
- Provides a tested, production-grade implementation of ViT-Base
- Automatically loads pretrained ImageNet weights
- Avoids re-implementing multi-head attention / positional encodings by hand
- Easy to configure with presets ("vit_base_patch16_224_imagenet") and num_classes=NUM_CLASSES

Why set backbone.trainable = False?

- Compute constraints:
    - ViT-Base has tens of millions of parameters;
    - Freezing the backbone makes training much lighter on GPU/CPU.
- Data vs model size:
    - With ~42k images, fully fine-tuning the backbone risks overfitting.
    - Freezing reuses robust general visual features learned from ImageNet.

Experiments & Results of ViT





Art Style	Precision	Recall	F1-Score
Academic Art	0.65	0.33	0.44
Art Nouveau	0.63	0.62	0.62
Baroque	0.56	0.44	0.49
Expressionism	0.56	0.77	0.65
Japanese Art	0.83	0.69	0.76
Neoclassicism	0.69	0.44	0.54
Primitivitism	0.64	0.54	0.59
Realism	0.64	0.56	0.60
Renaissance	0.52	0.77	0.62
Rococo	0.49	0.44	0.46
Romanticism	0.49	0.60	0.54
Symbolism	0.25	0.14	0.18
Western Medieval	0.81	0.80	0.81
Accuracy			0.57
Macro Avg	0.60	0.55	0.56

Key Observations

- Train/val accuracy plateaus around **57–60%**
- Several art styles show low recall/F1 despite decent precision
- Large gap vs CNN baseline → ViT underfits in current setup
- Symbolism, Academic_Art, Rococo consistently misclassified

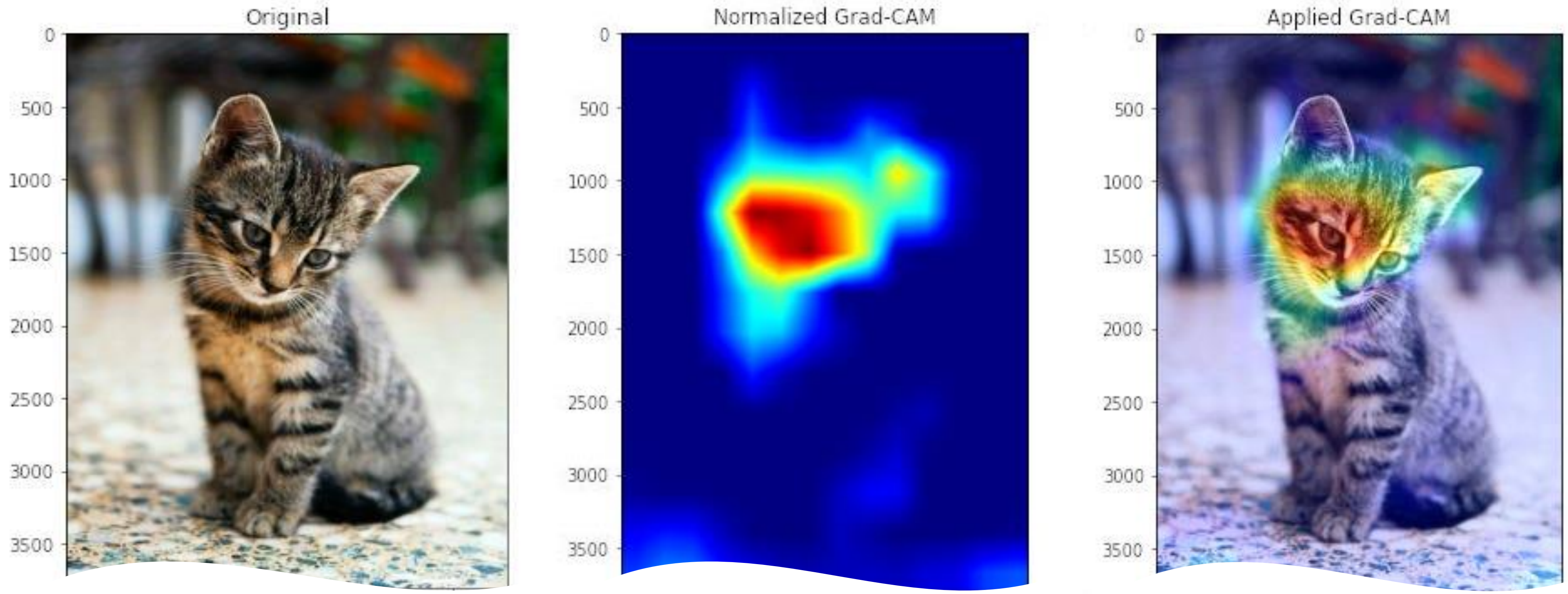
Why it Underperforms

- Backbone frozen → ViT cannot adapt ImageNet features to art-style textures
- Lower resolution (224) loses fine-grain patterns your CNN sees at 512
- Shallow training (only head trained, 15 epochs) → model underfits complex styles

Takeaways

- Current ViT acts like a fixed feature extractor → not enough for style classification
- Needs fine-tuning and higher resolution to close gap with CNN

Grad-CAM



What is Grad-CAM?

Grad-CAM is a method that shows where the model is “looking” when it makes a prediction.

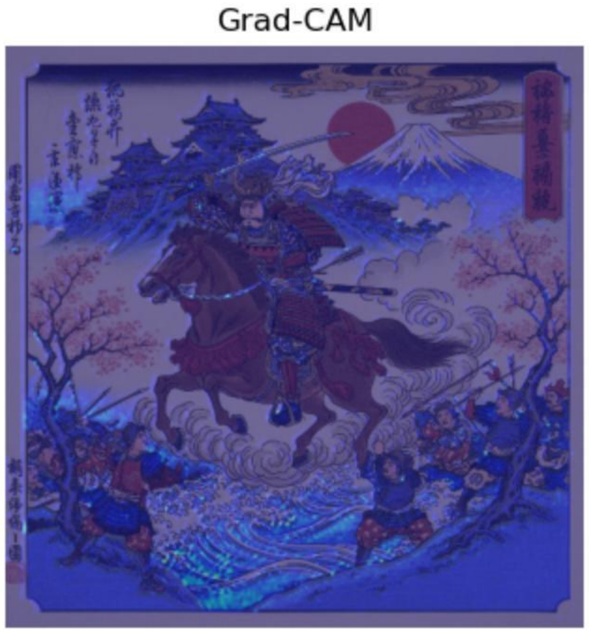
How it works?

- Takes the model’s predicted class.
- Computes which regions of the last convolutional layer matter most.
- Produces a heatmap that overlays on the painting.

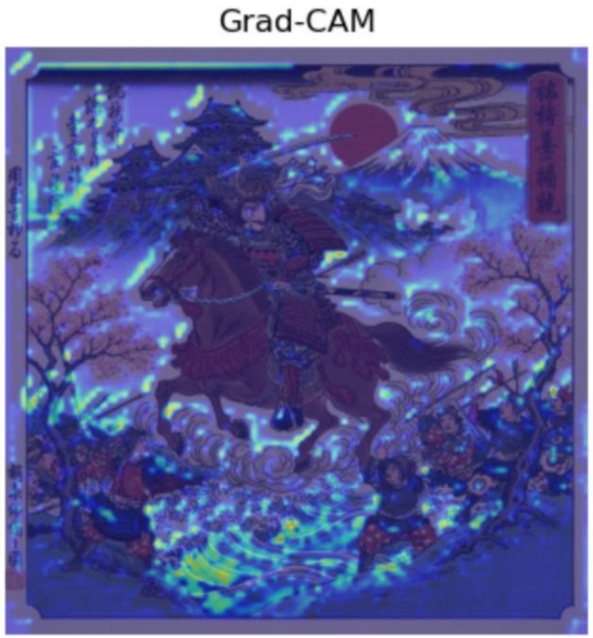
Why is it Helpful?

Increases interpretability by illustrating why the model made its prediction.

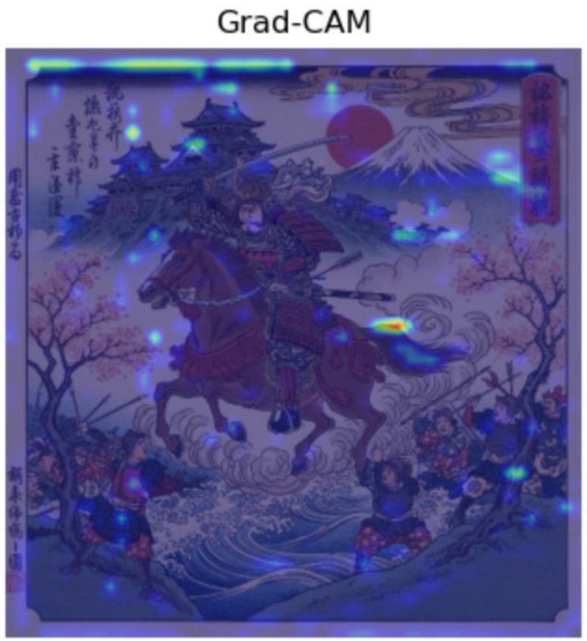
Grad-CAM Results – Baseline CNN



Output from Convolutional Layer 1



Output from Convolutional Layer 2



Output from Convolutional Layer 3

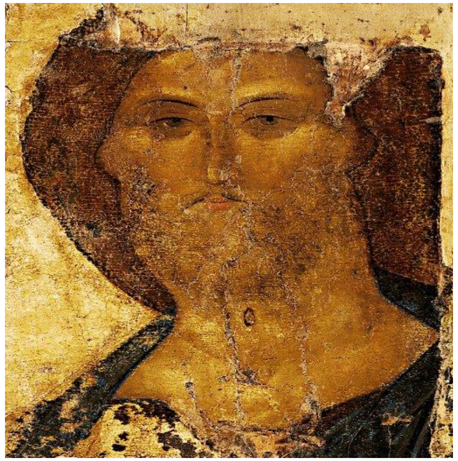
Conclusion & Key Insights

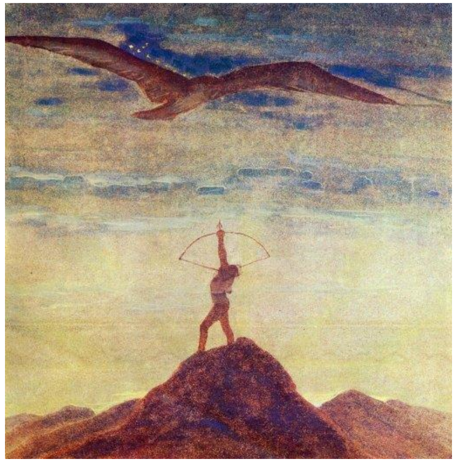
The Baseline CNN performs the best overall at about **0.77** accuracy, outperforming both Transfer Learning (about **0.68**) and the ViT model (about **0.57**).

All three models seem to correctly classify distinct art forms (Like Japanese and Western Medieval art) with high precision and recall.

The models struggle more with art forms that are more abstract in nature, like Symbolism art. They also struggle with art categories with significant overlap in techniques and time periods, like Renaissance, Baroque, Rococo, Neoclassicism Art.

F1-Score			
Art Style	CNN	TL	ViT
Academic Art	0.70	-0.14	-0.26
Art Nouveau	0.76	-0.04	-0.14
Baroque	0.78	-0.09	-0.29
Expressionism	0.73	-0.03	-0.08
Japanese Art	0.82	-0.01	-0.06
Neoclassicism	0.79	-0.07	-0.25
Primitivitism	0.81	-0.06	-0.22
Renaissance	0.76	-0.10	-0.16
Rococo	0.80	-0.03	-0.18
Romanticism	0.78	-0.18	-0.32
Realism	0.79	-0.12	-0.25
Symbolism	0.69	-0.38	-0.51
Western Medieval	0.82	0.02	-0.01
AVG Difference		-0.09	-0.21
Macro Avg Acc.	0.77	0.68	0.57





Limitations & Future Improvements

Limitations

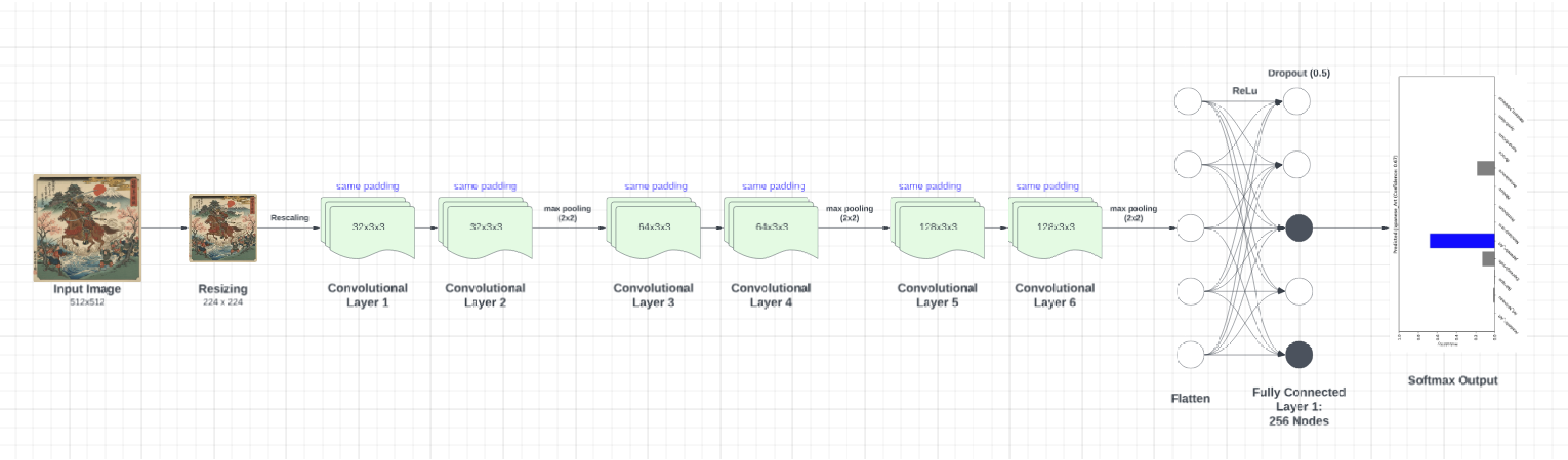
- Dataset Collection Method
    - Labeled by the artist, not individual artworks.
- Computing Power
- Label Noise
    - Sharing visual elements
    - Blended or transitional artworks?
    - Subjective

Future Improvements

- Expand Dataset
    - More Art Movements
    - Labeled by Artworks
- Model Enhancements
    - Self-supervised Learning
- Data Augmentation
    - Brightness, Color, Shear, Cropping
- Improved Labeling
    - Multi-label Classification
    - Manually Labeled

Thank You!!

Appendix 1: Future Improvements to CNN Model



01 Design Improvements Made to the Baseline CNN Model

- Double the number of Convolutional layers from 3 to 6
- Reduce the Fully Connected Layer's dependence on the model by reducing the number of nodes (512 -> 256)
- Same padding in every convolutional layer
- Max Pooling after every two layers