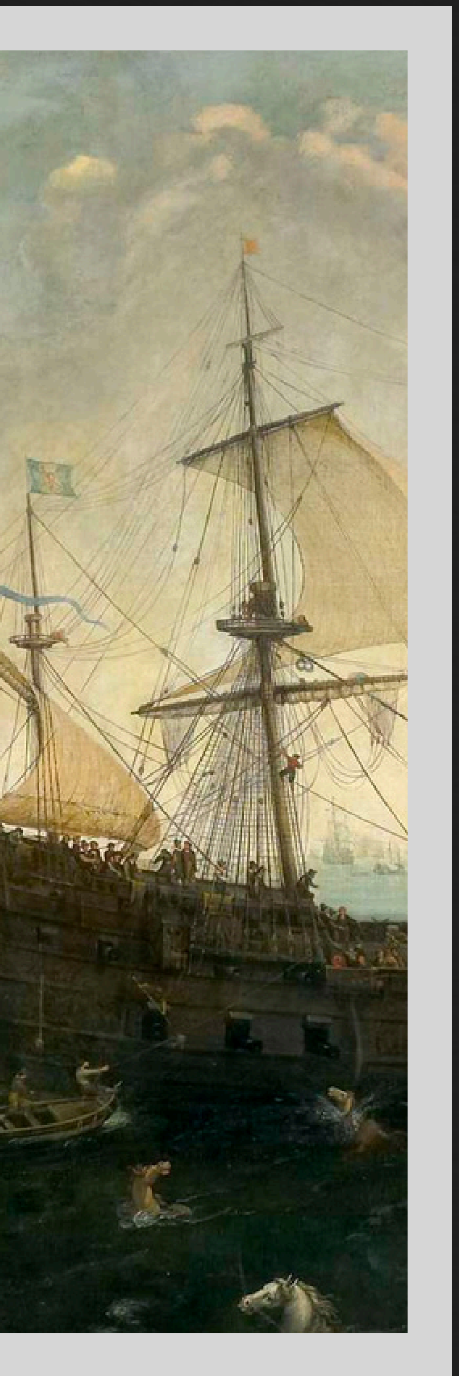
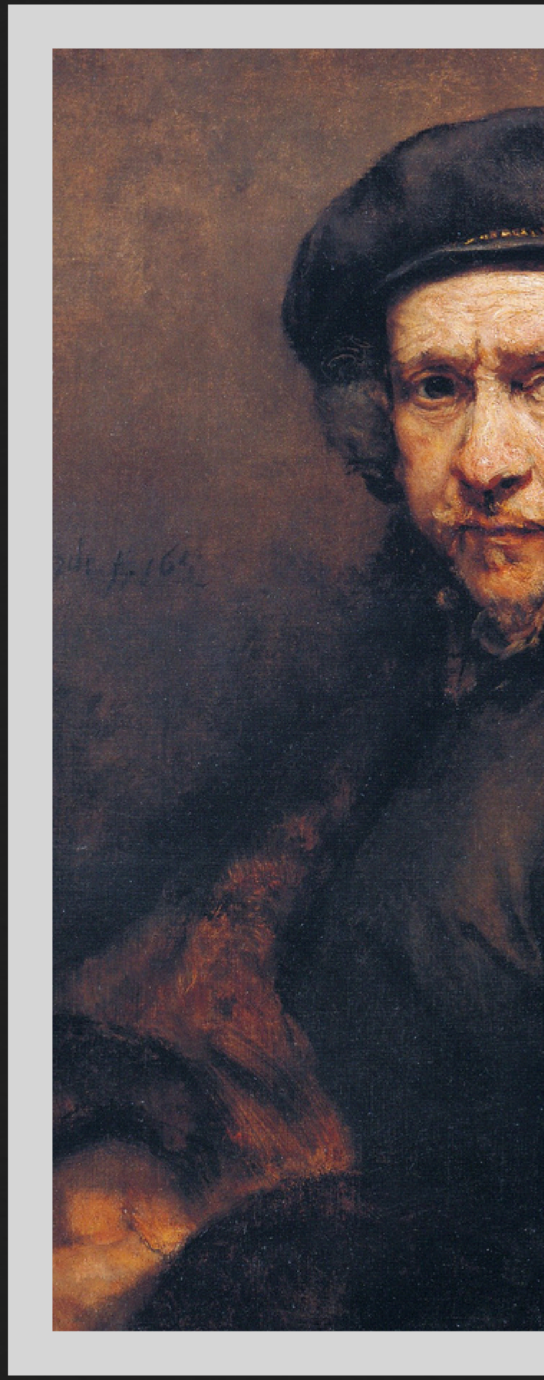


# CNN-BASED ART ANALYSIS

Anusha Bhat, Deepali Dagar, Nidhi Pareddy, Ritai Na, Sara Chaker

**ADSP 32023: Advanced Computer Vision with Deep Learning**

**May 29, 2025**



# Agenda

**01** Business Problem

**02** Exploratory Data Analysis

**03** Style Transfer

**04** Fake Art Detection

**05** Live Demo

# Business Problem

*Build an end–to–end, real–time image processing platform that:*

Stylizes images using iconic art forms like Cubism for creators and social media users seeking visually compelling content.

Identifies synthetic or manipulated content using deep learning to combat misinformation and ensure content authenticity.

Delivers a seamless and intuitive user experience through a browser-accessible platform that requires no installation.

# Datasets

Style Transfer

- WikiArt dataset
- 80,020 unique images from 1119 different artists in 27 styles
- Specifically used the Cubism genre with 2,235 images

Fake Art Detection

- 10,300 AI generated art images
- 8,288 real art images
- All images have a 512 x 512 resolution

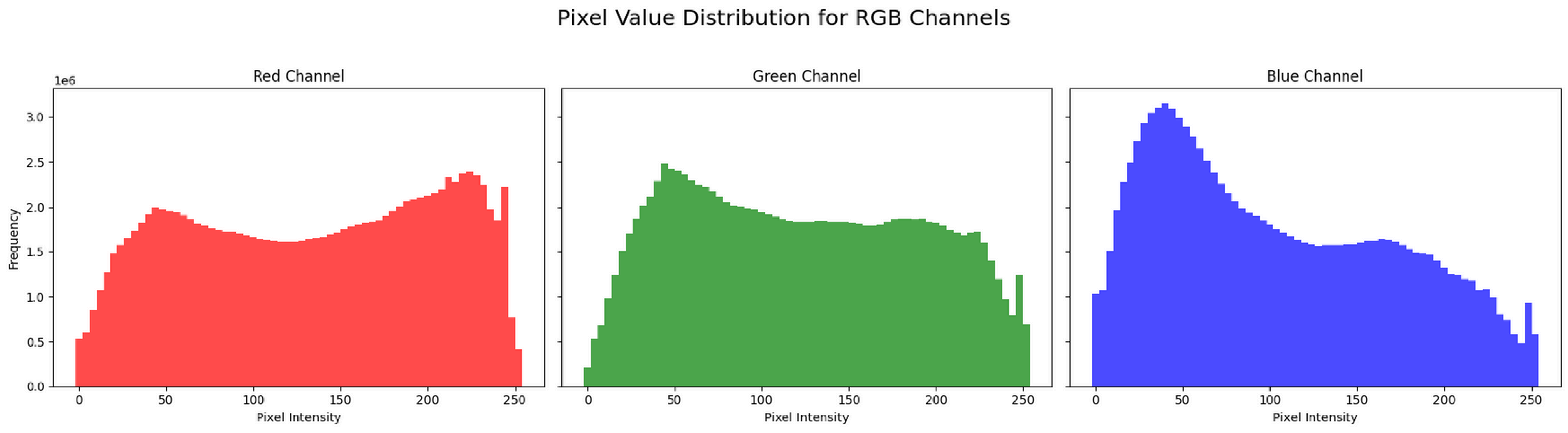
Cubism Sample

Human Art Sample

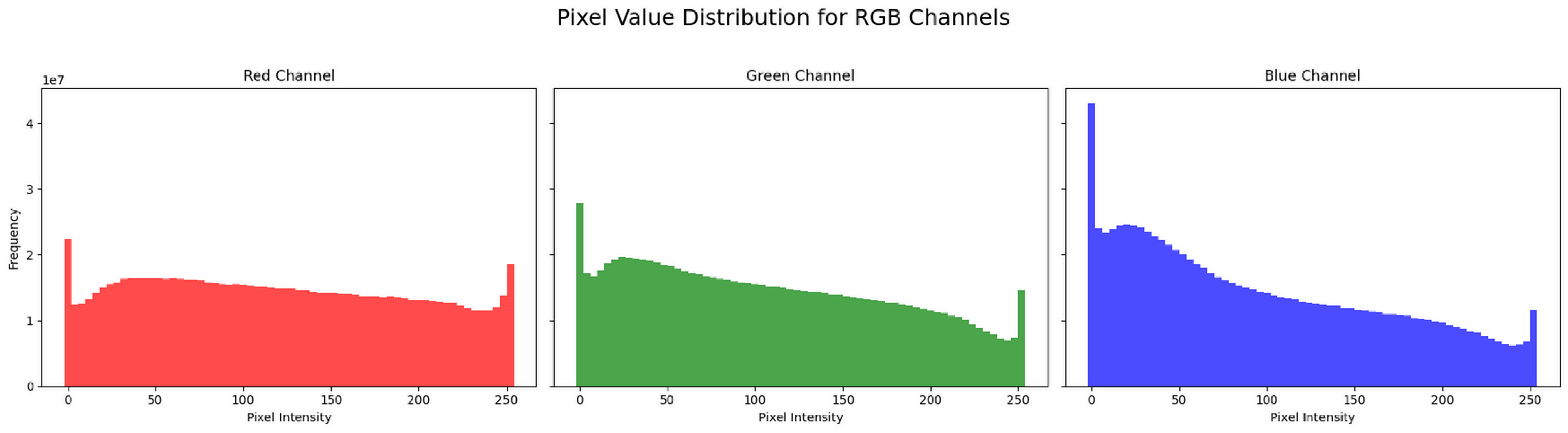
AI-Generated Art Sample

# Exploratory Data Analysis

Style Transfer




Fake Art Detection



# Models

## Style Transfer

**Champion**
VGG pretrained Neural style and Fast Style Transfer

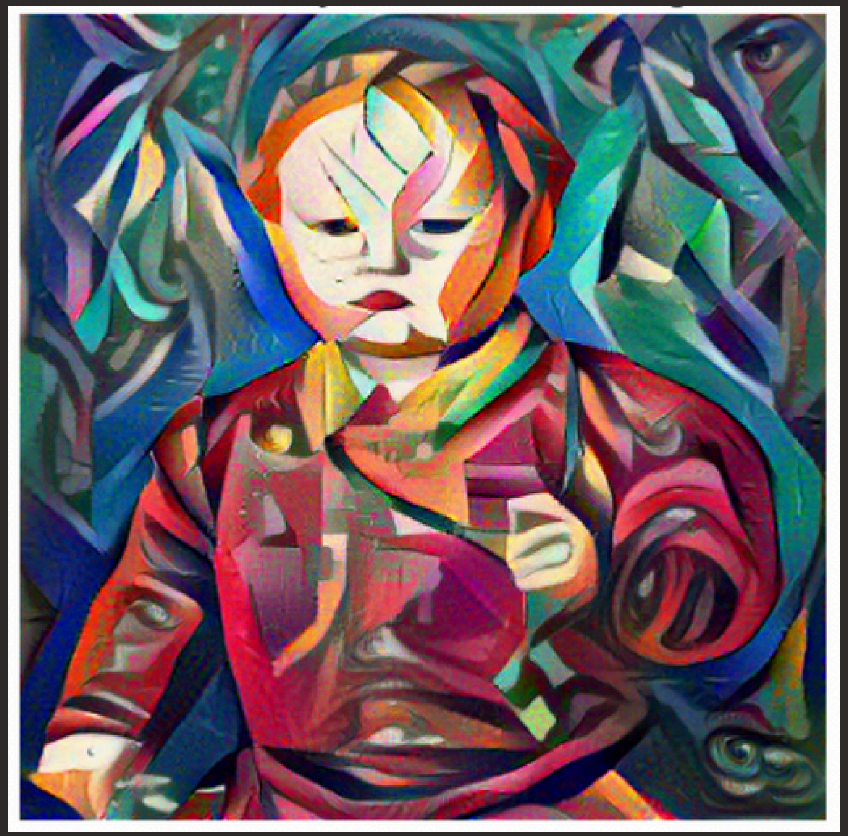
**Challenger**
NoiseInit-VGG Style Transfer and Custom CNN

## Fake Art Detection

**Champion**
Pre-trained DINO model with a classification head

**Challenger**
EfficientNet

# Cubism Style Transfer – Champion



Using VGG Neural  Style Transfer

## VGG- Neural Style Transfer:

**Transfer Engine**

- Uses VGG19 **with freezing parameters** to extract content and style features

**Two-Phase Learning**

- Content: Preserves image structure
- Style: Learns artistic textures via Gram matrices

**Optimization**

- Uses LBFGS optimizer for stable and high-quality convergence

## Fast Style Transfer:

**Transfer Engine**

- Pretrained TF Hub model designed for real-time stylization

**One-Shot Inference**

- No training or optimization needed
- Model directly maps content and style to stylized output

**Deployment-Ready**

- Fast, lightweight, & easily converted to TFLite for mobile/web apps

# NoiseInit–VGG and Self Based CNN Withought VGG

NoiseInit-VGG Style Transfer

## NoiseInit-VGG Style Transfer:

**Transfer Engine**

- Uses VGG19 features up to conv4_1, **without freezing parameters**.

**Two-Phase Learning**

- Content: Starts from random noise.
- Style: Learns style using fewer layers, with equal weights

**Optimization**

- Applies gradient descent with Adam optimizer and high learning rate

## Scratch CNN:

**Transfer Engine**

- Uses a custom-built CNN for feature extraction and image transformation.
- Does not rely on any pretrained models like VGG.

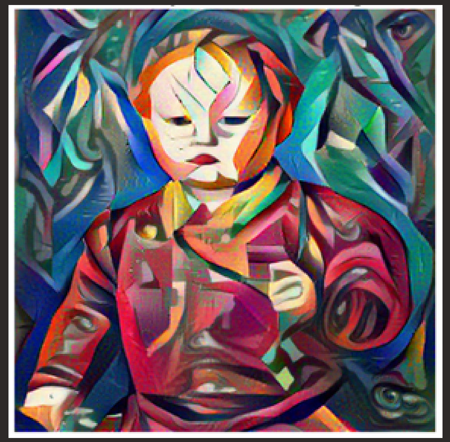
**Two-Phase Learning**

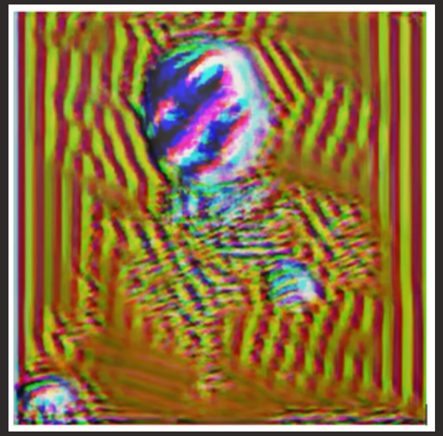
- Content: Preserves image structure using deep encoder features from the self-built network
- Style: Matches multi-layer Gram matrices of intermediate features to capture artistic texture

**Optimization**

- Applies Adam optimizer to iteratively update the image, combining style loss, content loss, and total variation loss
- Trains from scratch with randomly initialized CNN weights

# Comparison – Style Transfer Methods

| FEATURE | VGG- Neural Style Transfer | Fast Style Transfer: pre - trained | NoiseInit-VGG Style Transfer | Scratch_CNN |
|---|---|---|---|---|
| **LPIPS Score (lower is better)** | 0.80 | 0.66 | 0.82 | 0.88 |
| **Speed** | **Slow** | **Fast** | **Fast** | **Fast** |
| **Customization** | Yes | No | No | Yes |
| **Deployment Ready** | Moderate (with conversion) | Yes (.tflite) | No | Yes (TorchScript / ONNX) |

# Fake AI Art Detection

## ***Challenger***

### Model Training & Evaluation

- **EfficientNet convolutional network** that “sees” and summarizes each artwork into a compact fingerprint
- **Two-Phase Learning**
  - General Feature Learning: The model first learns broad visual patterns (color, textures, shape)
  - Fine-Tuning: re-trained on specific examples to specialize
  - Randomly flip, rotate, and zoom to teach the model to focus on content

## ***Champion***

### Model Training & Evaluation

- **DINO (Self-Distillation with No Labels)** using a pre-trained Vision Transformer (ViT).
  - Learns rich image representations without supervision by aligning augmented views via a student-teacher setup.
- **Linear Support Vector Machine (SVM)** classifier head trained on frozen DINO embeddings.
  - Robust in high-dimensional feature space and able to find a maximum-margin decision boundary.

# Fake AI Art Detection – *Results*

## **Champion** – DINO

Accuracy: 91.03%

Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Human-made | 0.92 | 0.88 | 0.90 | 1658 |
| AI-generated | 0.91 | 0.94 | 0.92 | 2066 |
| accuracy | | | 0.91 | 3724 |
| macro avg | 0.91 | 0.91 | 0.91 | 3724 |
| weighted avg | 0.91 | 0.91 | 0.91 | 3724 |

Confusion Matrix:

```
[[1456  202]
  132 1934]]
```

## **Challenger** – EfficientNet

Accuracy: 89.42%

Classification Report:

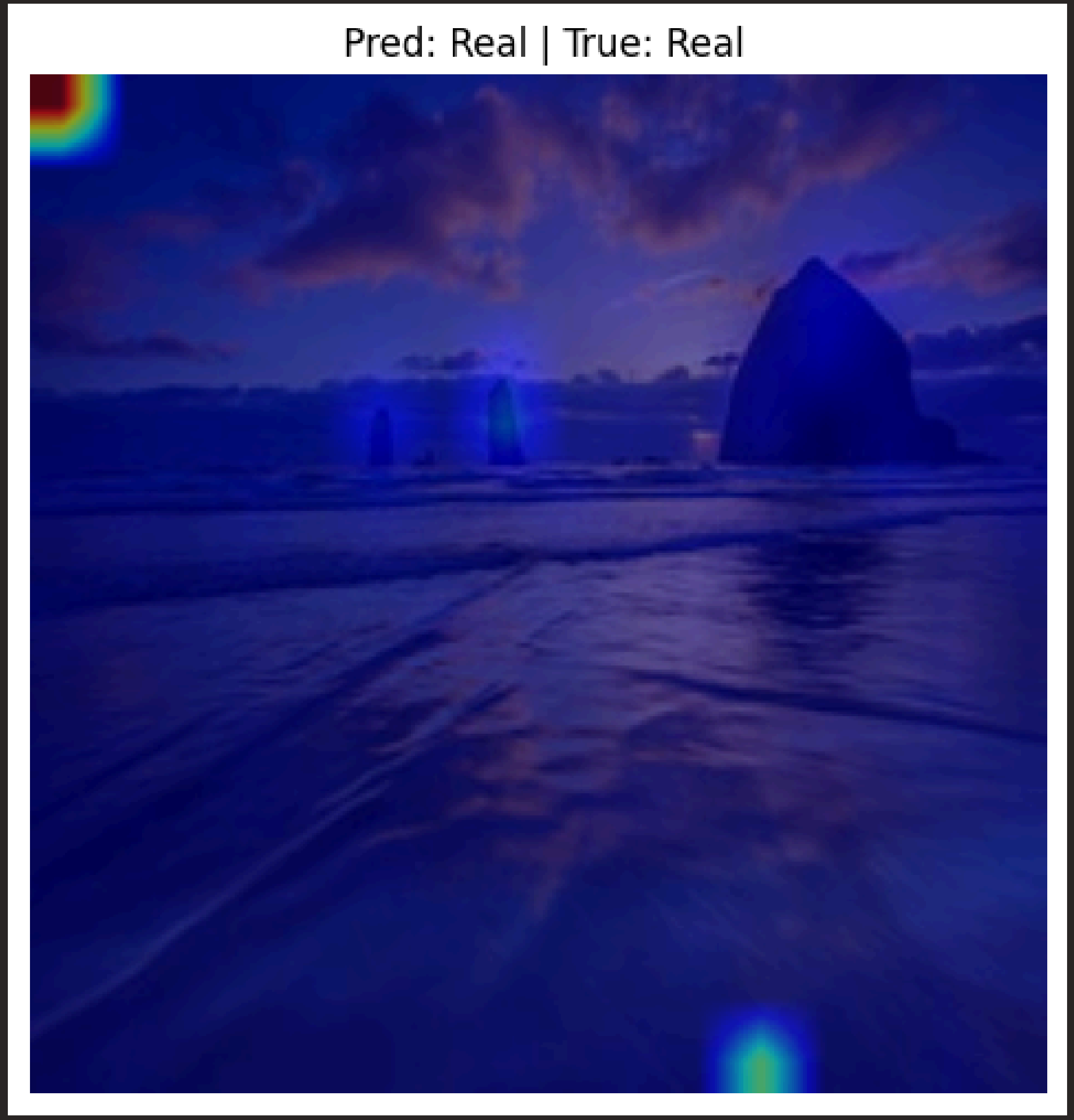
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| AI_GENERATED | 0.88 | 0.93 | 0.91 | 2066 |
| NON_AI_GENERATED | 0.91 | 0.85 | 0.88 | 1658 |
| accuracy | | | 0.89 | 3724 |
| macro avg | 0.90 | 0.89 | 0.89 | 3724 |
| weighted avg | 0.89 | 0.89 | 0.89 | 3724 |

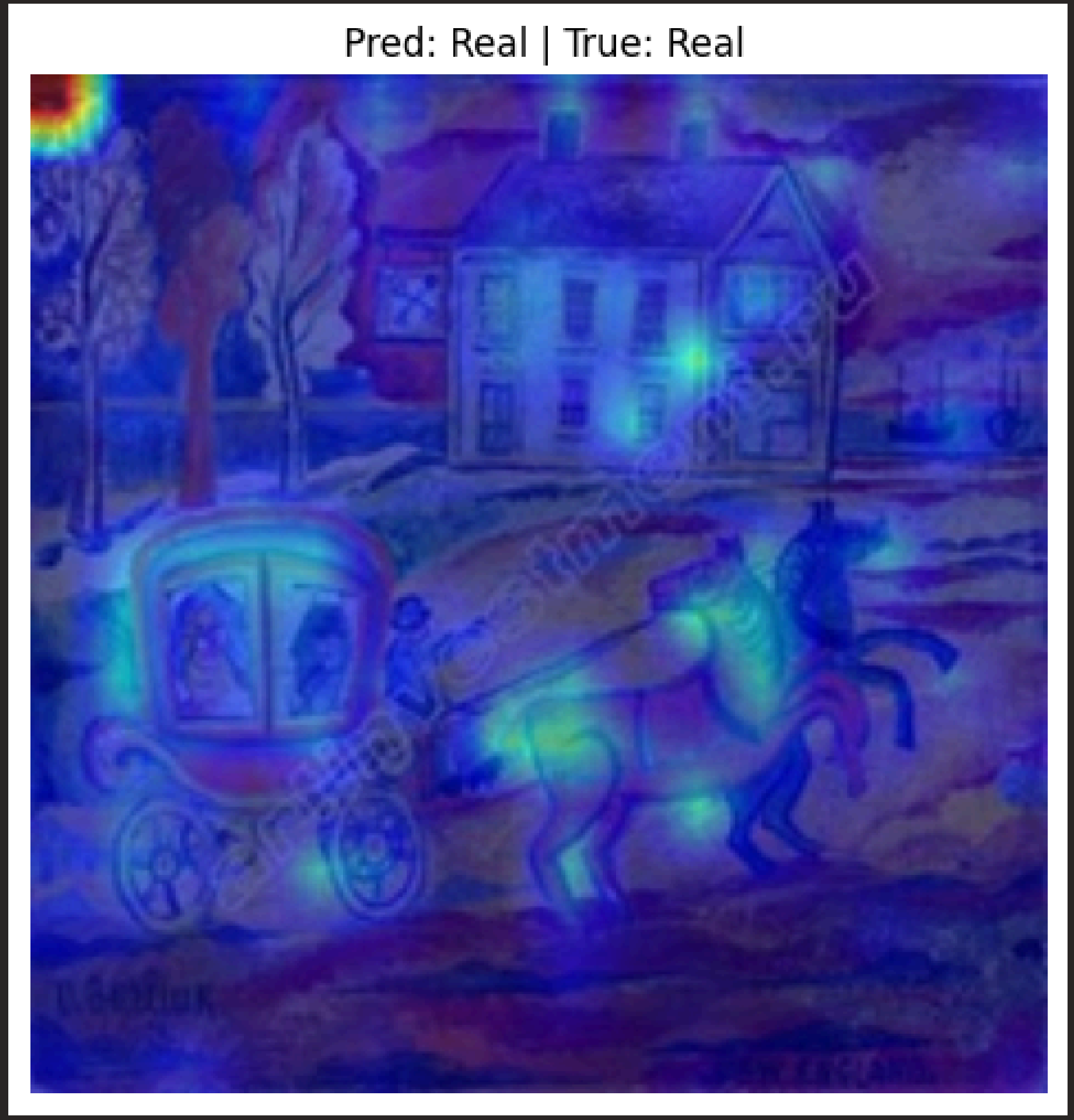
Confusion Matrix:

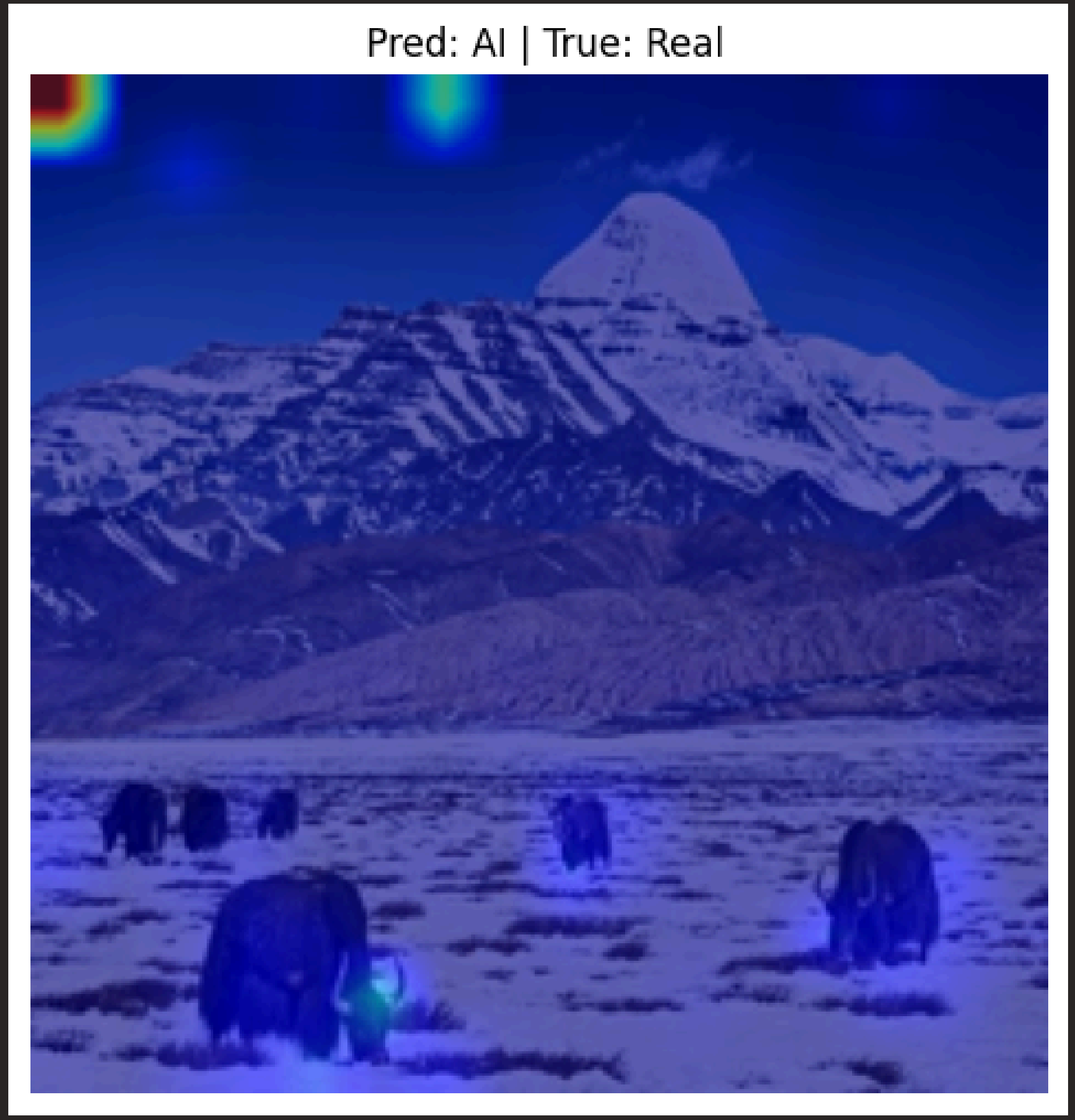
```
[[1926  140]
 [ 254 1404]]
```

# Attention Maps from DINO Predictions

*Visualizing Model Focus in Real vs. AI-Generated Art Classification*



# Fake Art Detection: *Model Pros & Cons*

## Challenger: EfficientNet

✓ **Pros:**

- **Generalizes well**: The two-phased approach lets the model adapt to new art trends ot tweaks.
- **Lightweight Inference:** Once trained, it's able to scan and score images quickly on standard GPU's

✗ **Cons:**

- **Heavy Upfront Training:** Needs poweful GPU and several hours of compute time
- **Threshold Tuning:** Human calibration required to balance false alarms vs. misses

## Champion: DINO

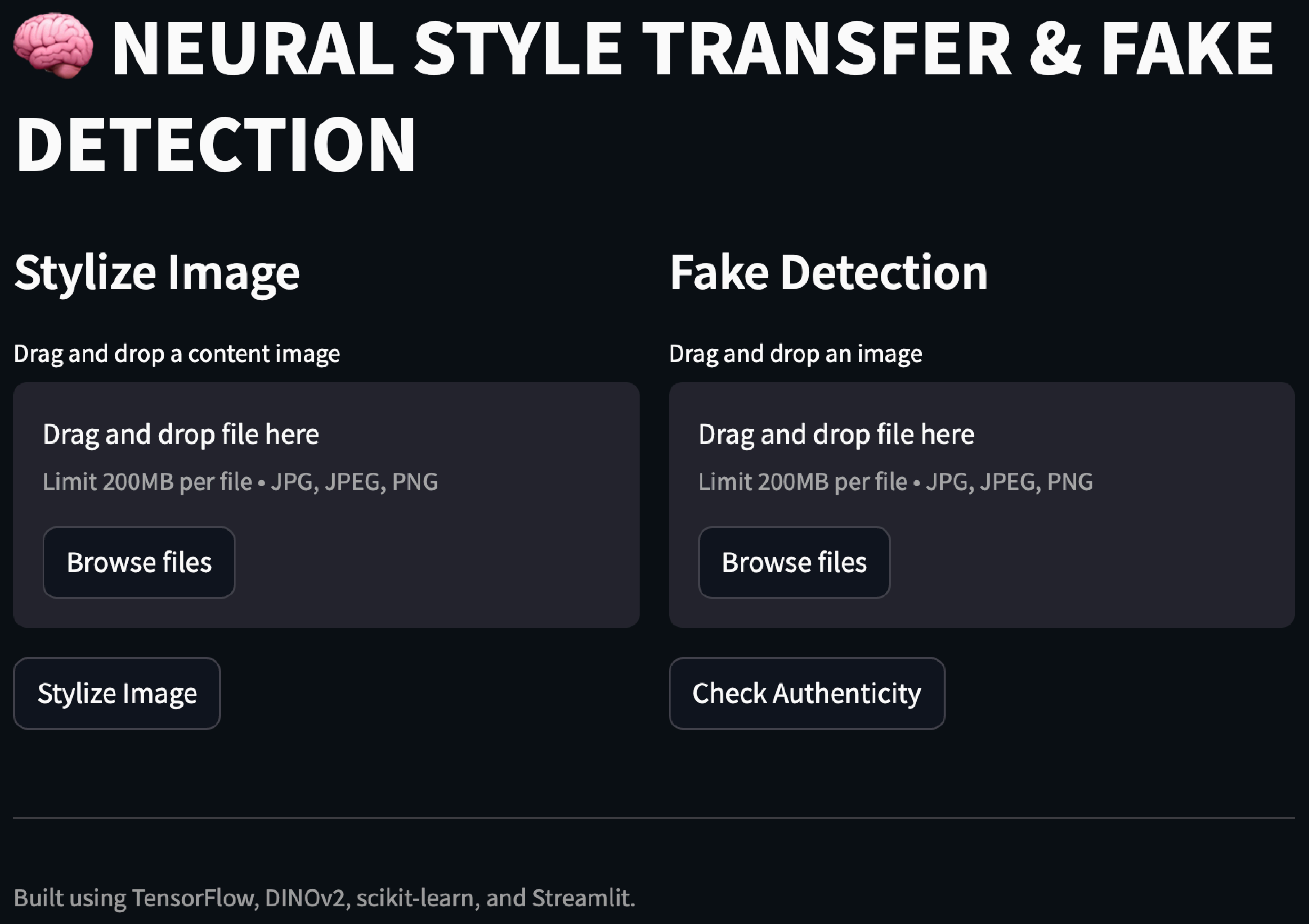
✓ **Pros**

- DINO extracts semantic features even without labels, which is ideal for abstract concepts like "artistic authenticity."
- SVM is effective for small labeled datasets and avoids overfitting due to regularization.

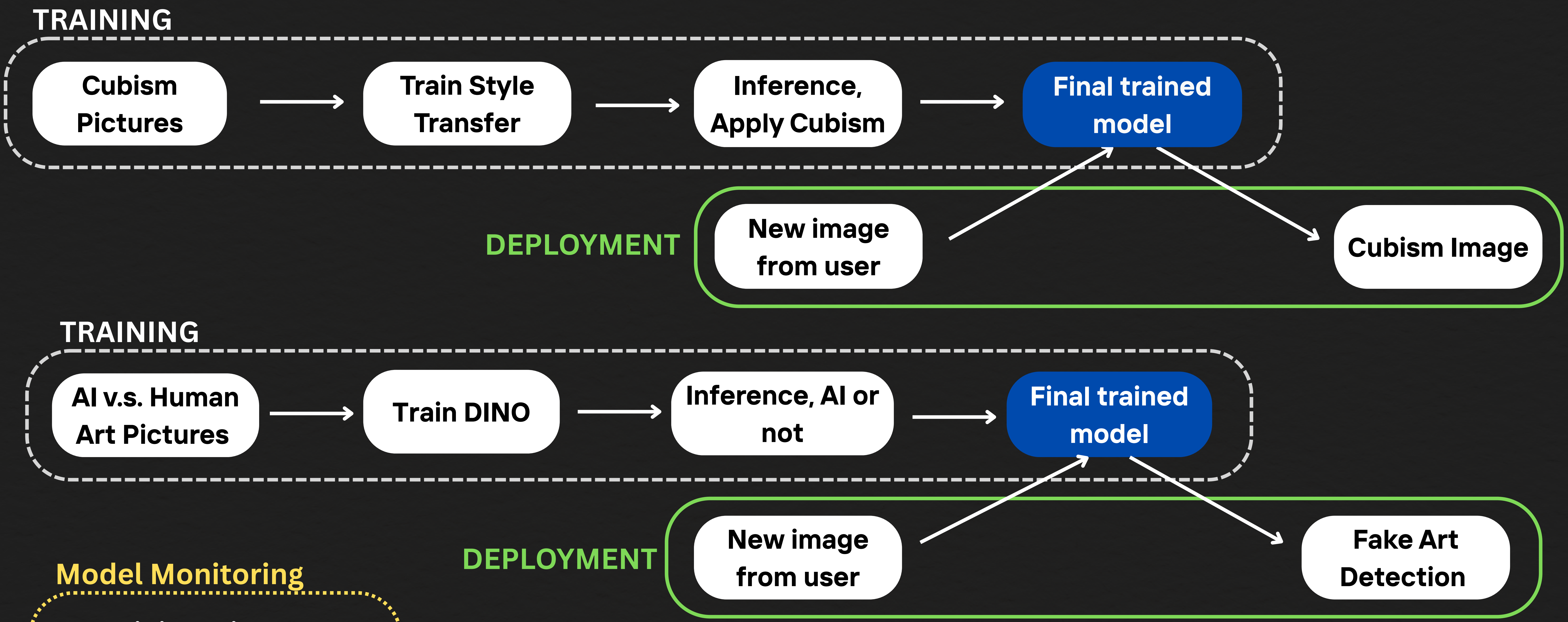
✗ **Cons**

- DINO is computationally heavy to train.
- SVM lacks probabilistic outputs unless calibrated.

# Live Demo


NEURAL STYLE TRANSFER & FAKE DETECTION
Stylize Image
Drag and drop a content image
Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG
Browse files
Stylize Image
Fake Detection
Drag and drop an image
Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG
Browse files
Check Authenticity
Built using TensorFlow, DINOv2, scikit-learn, and Streamlit.

# Model Operations




**Model Monitoring**

Retraining trigger:

- When accuracy drops <0.85

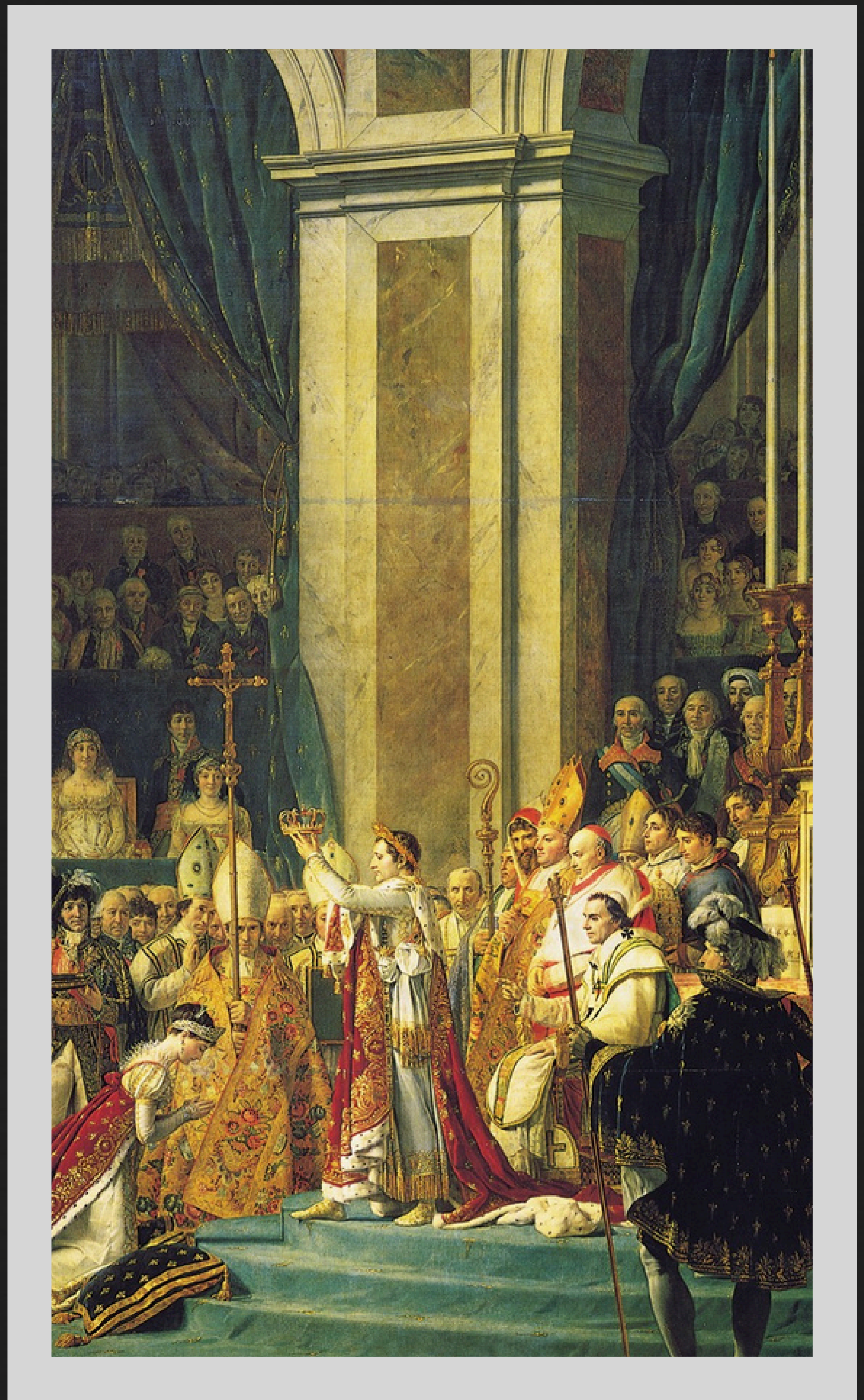
# Future Work

Expand Style Coverage: Broaden artistic genres beyond Cubism.

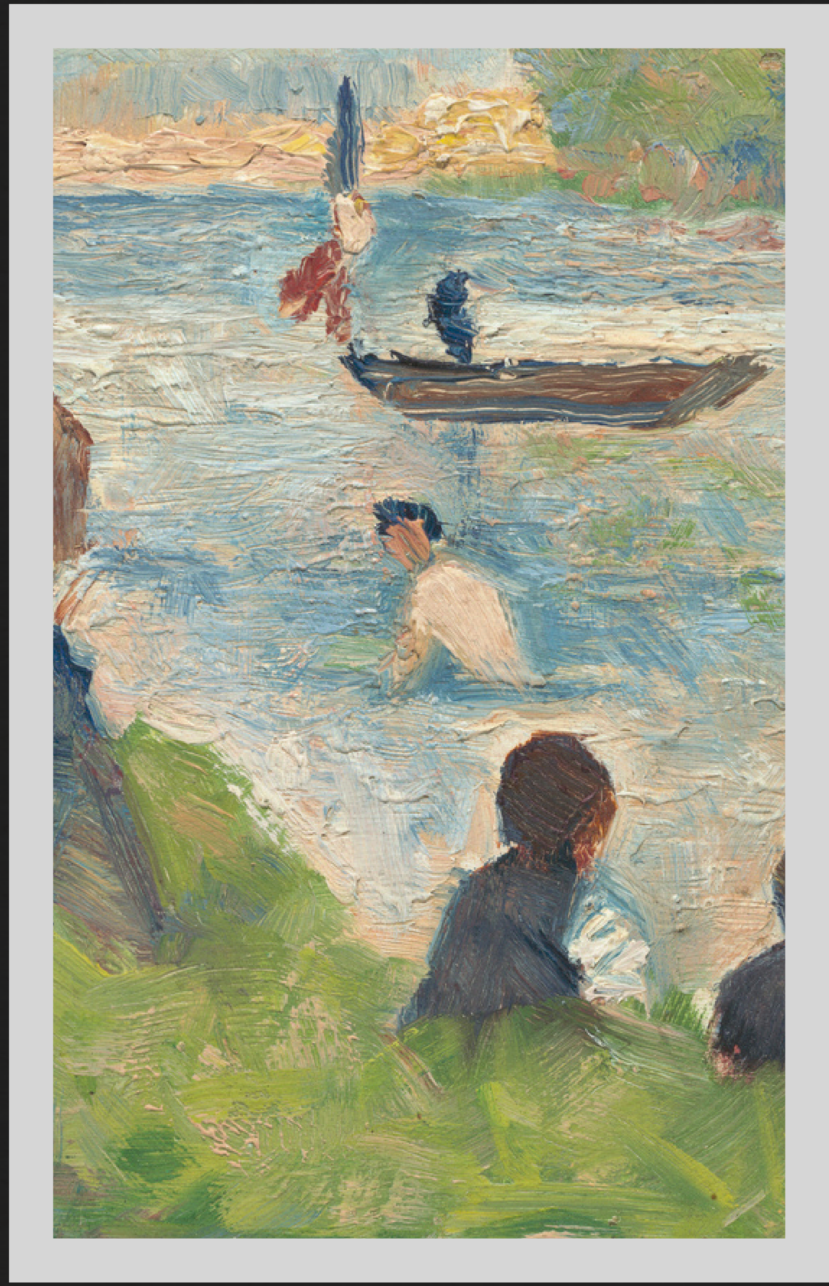
Interactive Style Control: Use sliders or presets for personalized style and content adjustments.

Model Feedback & Monitoring: Integrate feedback mechanisms for automatic retraining and improvement.

Improved Interpretability for Fake Detection: Utilize explainability techniques to help users understand why artwork is flagged as fake, fostering trust and adoption.



# Thank You!



# Appendix

**For access to the full code:**

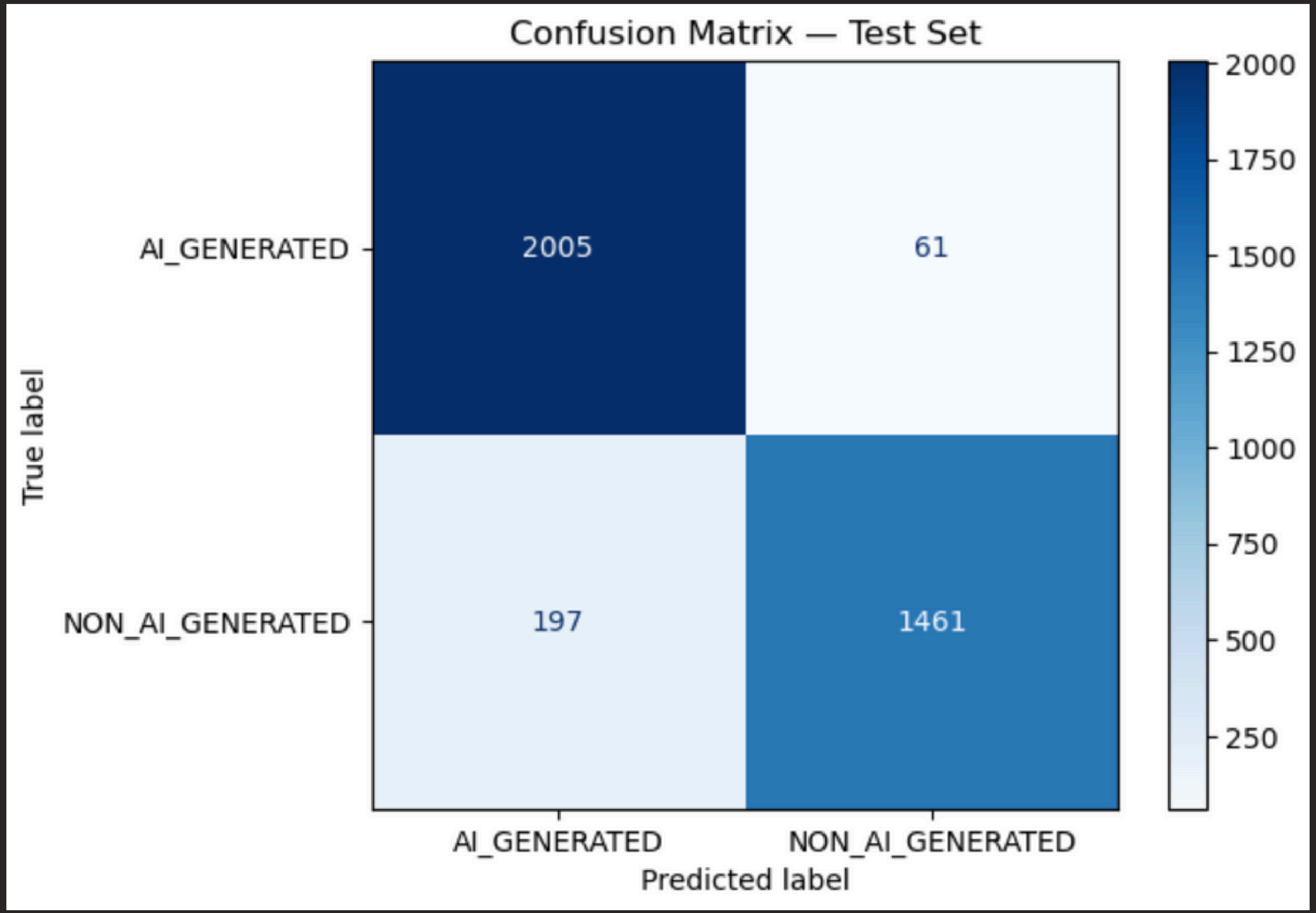
https://github.com/abhat09/CV_Final_Project

# Fake AI Art Detection – *Challenger*

## *Results*

Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| AI_GENERATED | 0.91 | 0.97 | 0.94 | 2066 |
| NON_AI_GENERATED | 0.96 | 0.88 | 0.92 | 1658 |
| accuracy | | | 0.93 | 3724 |
| macro avg | 0.94 | 0.93 | 0.93 | 3724 |
| weighted avg | 0.93 | 0.93 | 0.93 | 3724 |



# Fake AI Art Detection – *Champion*

## *Results with Logistic Regression Classifier Head*

**DINOv2 with Logistic Regression Classifier**

Accuracy: 90.87%

Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| Human-made | 0.91 | 0.88 | 0.90 | 1658 |
| AI-generated | 0.91 | 0.93 | 0.92 | 2066 |
| accuracy | | | 0.91 | 3724 |
| macro avg | 0.91 | 0.91 | 0.91 | 3724 |
| weighted avg | 0.91 | 0.91 | 0.91 | 3724 |

Confusion Matrix:

```
[[1465  193]
 [ 147 1919]]
```

# Cubism Image used for training: