AI-GENERATED ART DETECTION

Presented by
Zakarya BOUDRAF
Mohamed Aziz KHITMI
Melissa OULD BRAHAM

Supervised by Prof. Carmen BISOGNI
and Prof. Fabio NARDUCCI

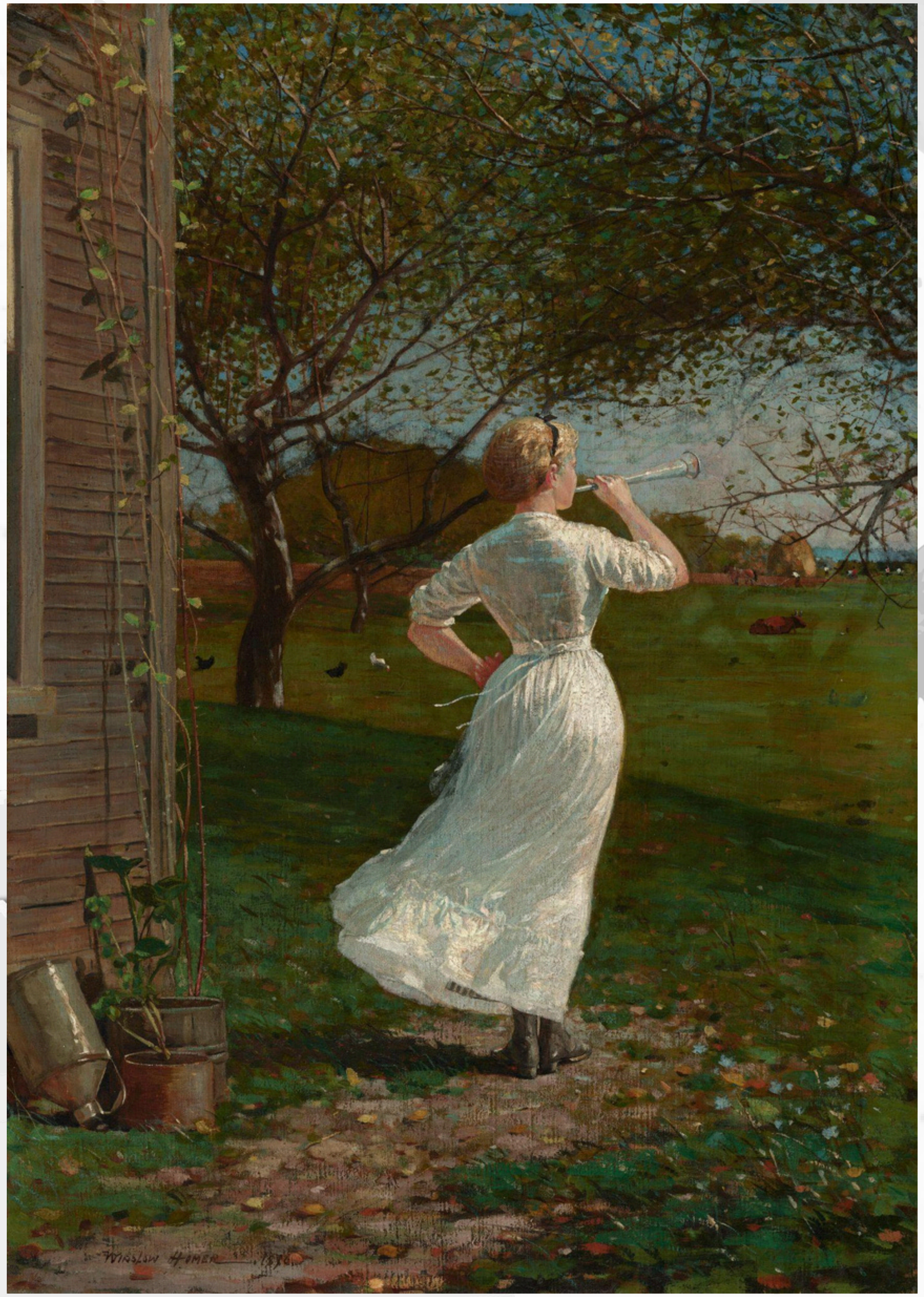
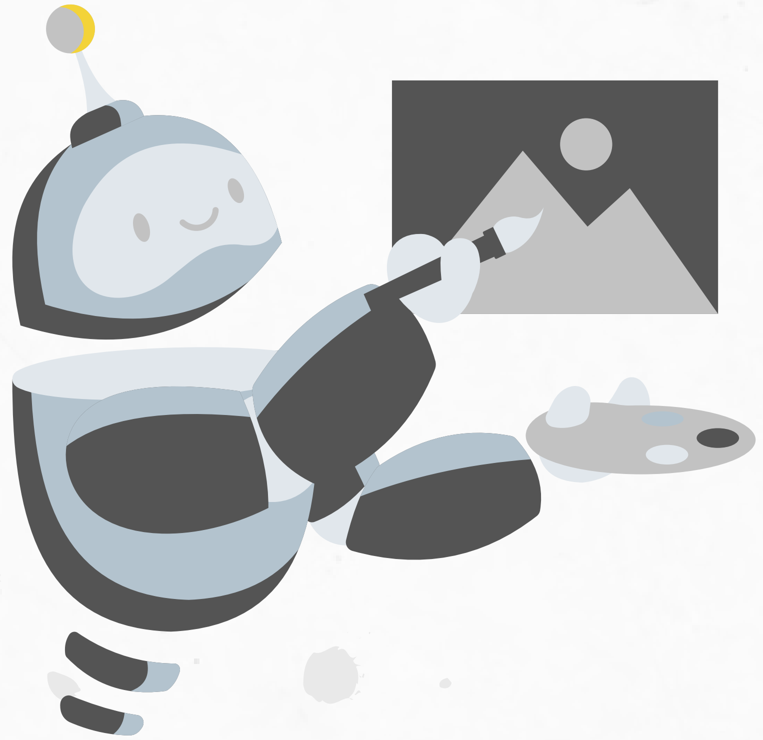


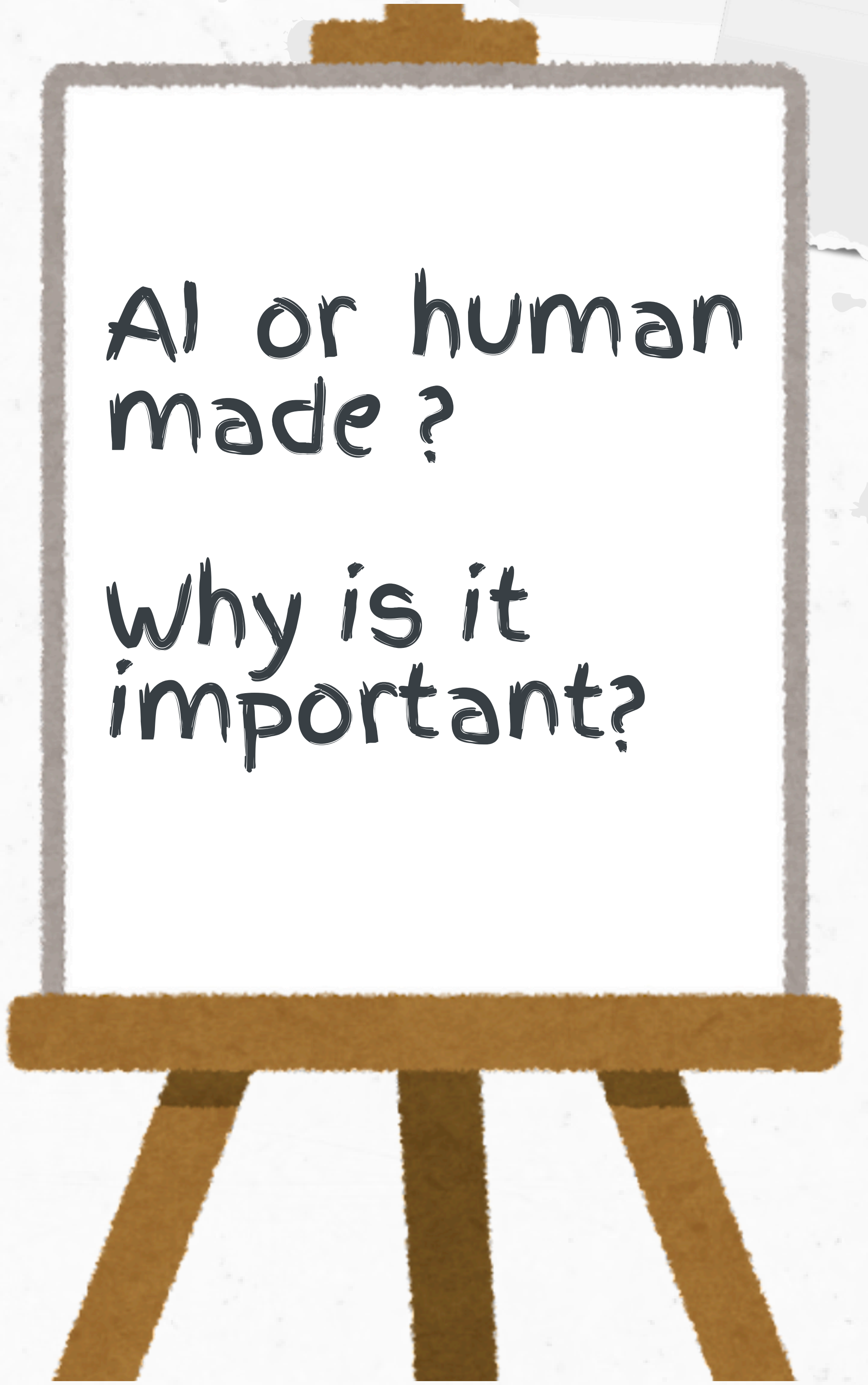
Table of contents

Introduction

Introduction

The world of art is rapidly changing. Artificial intelligence (AI) is now generating its own creative works, blurring the lines between human and machine.





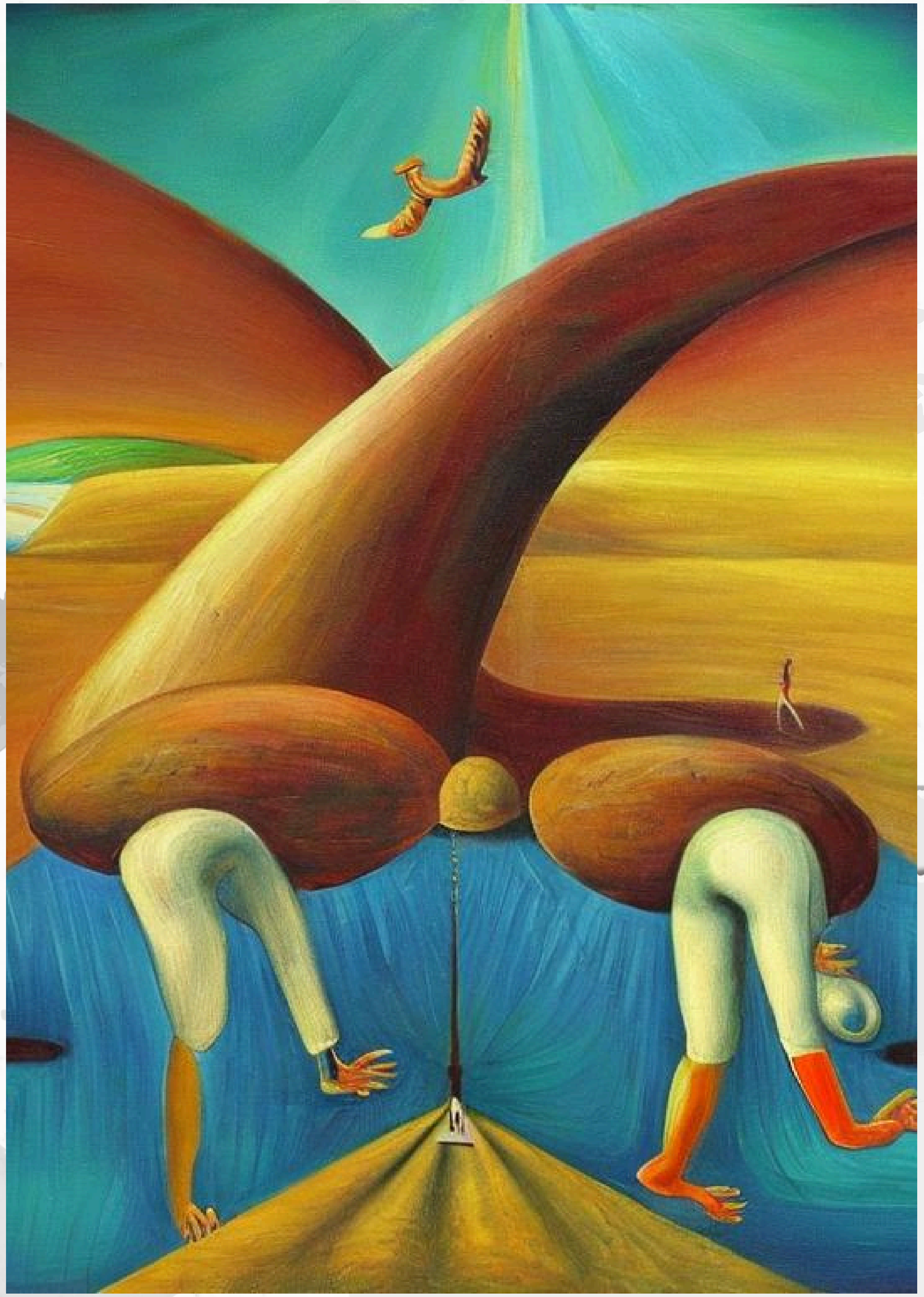
AI-Generated Art Classification

AI-Generated Art Classification

- **Problem**: Distinguishing between real art and AI-generated art.

- **Solution**: Binary classification using machine learning.

- **Impact**: Protects artists' rights, fosters transparency in the art market, aids art research.



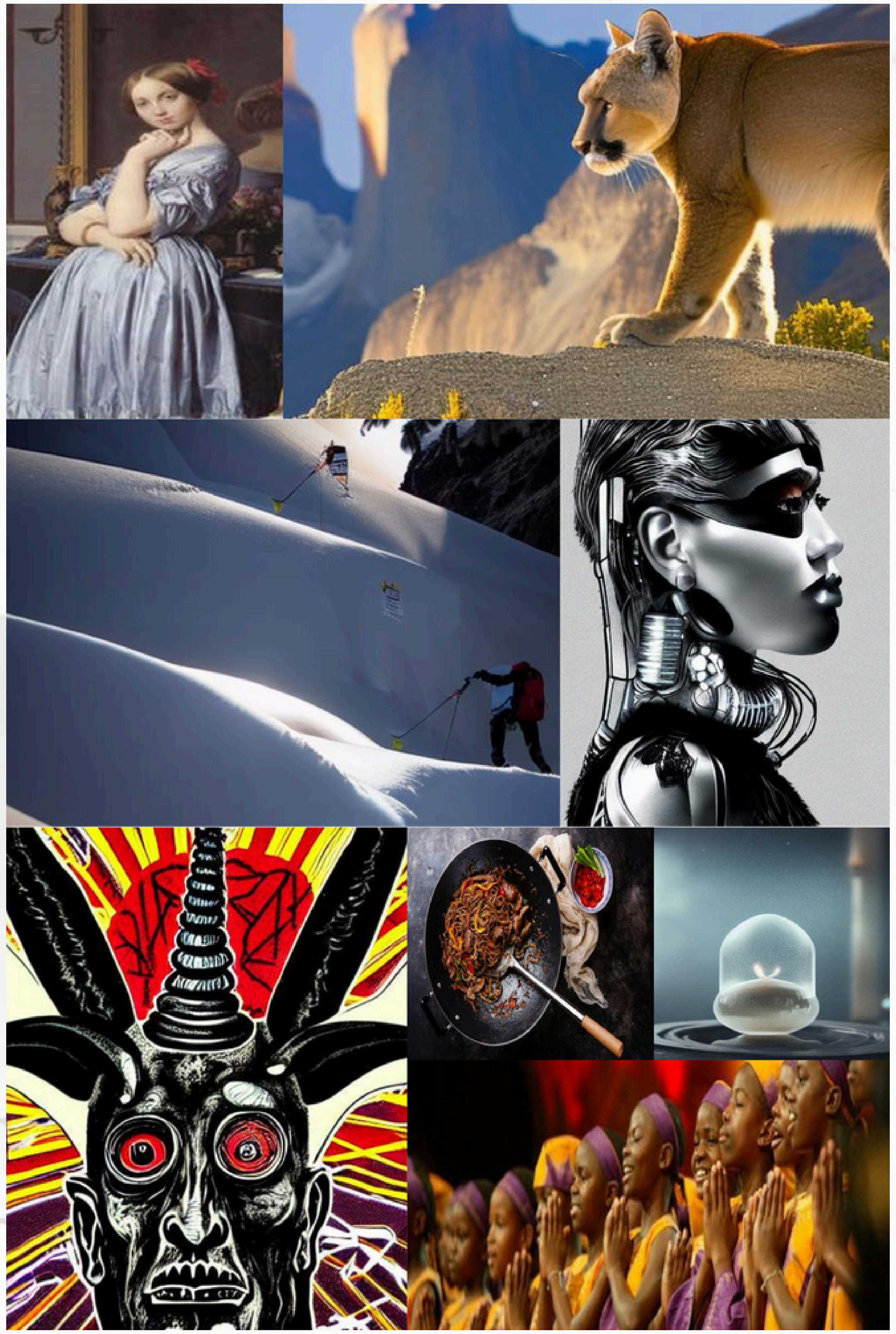
Datasets Used

AI-ArtBench

- **Source:** Publicly available on Kaggle
- **Size**: 180,000+ images 60,000 Human-drawn Art (256x256 resolution)
- **Remainder**: AI-Generated Art (Latent Diffusion & Standard Diffusion) respectively 256x256 resolution 768x768 resolution
- **Key Strengths**:

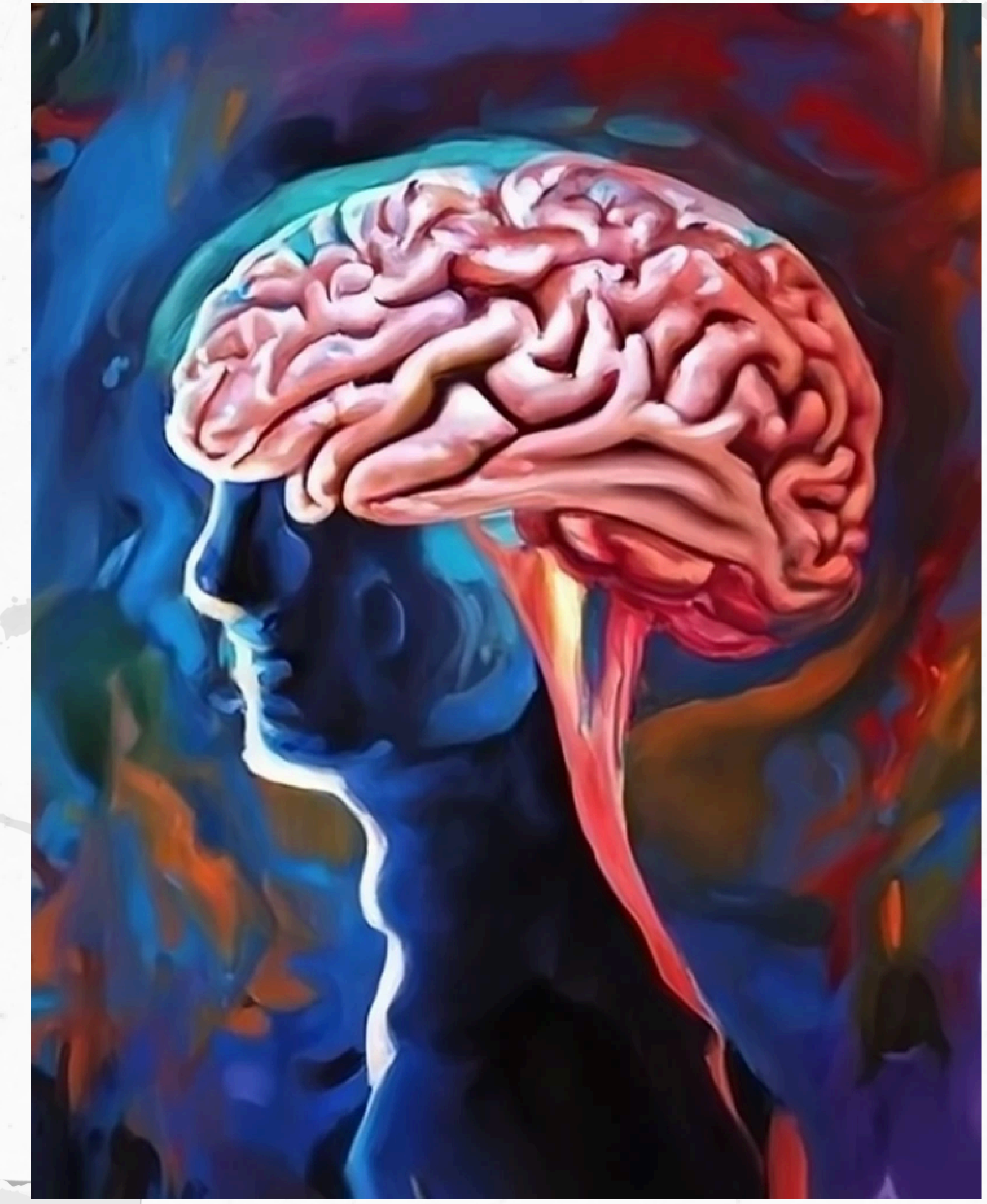
1. **Class-Balanced**: Ensures all artistic styles (10) are represented equally.
2. **High-Quality**: Curated for clarity and suitability for machine learning.
3. **Cleanly Annotated**: Images labeled with the correct artistic style.
4. **Standardized**: Easy integration with various machine learning frameworks.

AI and Human Art Classification



- **Source:** Publicly available on Kaggle
- **Size:** AI-generated art has 8k+ images and 10k+ images in Non-AI generated Images
- **Strengths:** Balanced Classes (equal human and AI art)
- **Limitations:**

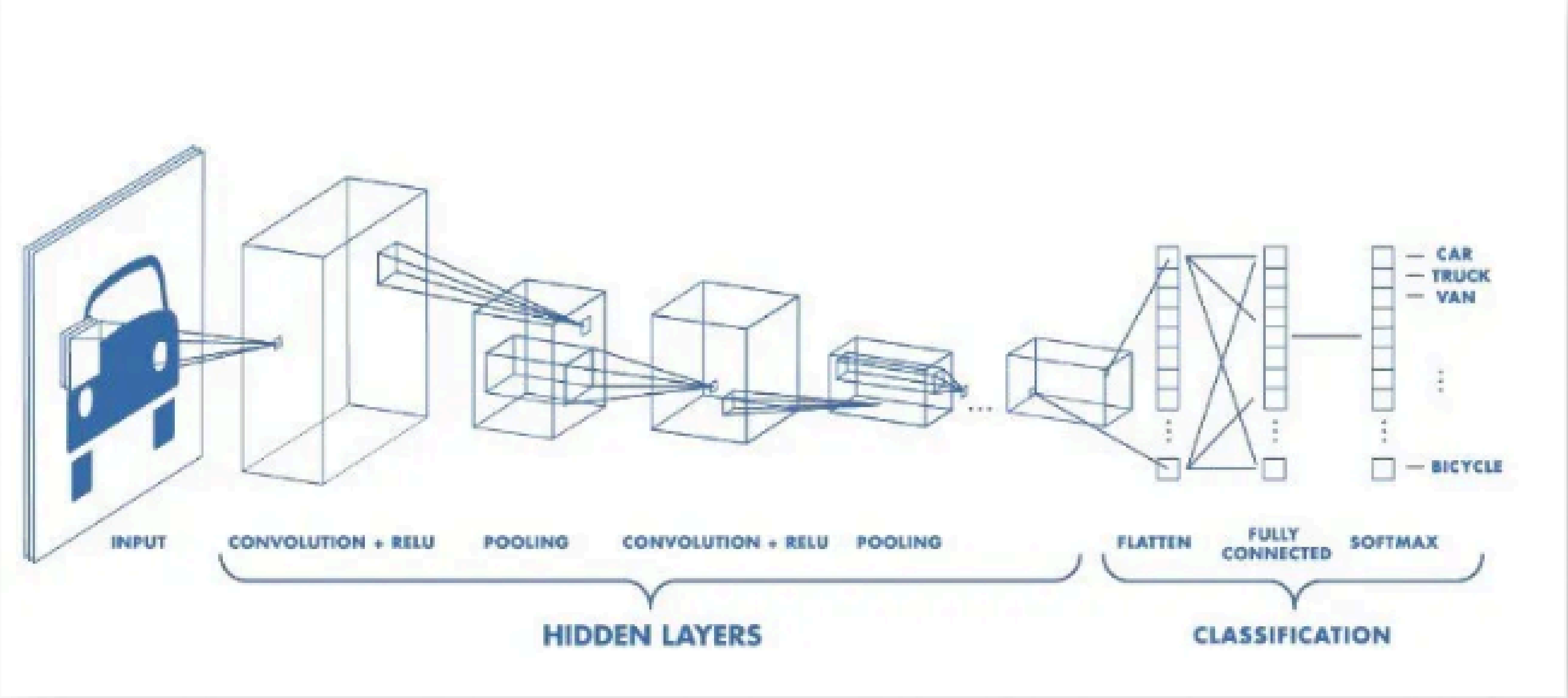
1. Smaller size compared to AI-ArtBench
2. Limited information on artistic styles/AI models used for generation
3. Broader artistic variety compared to AI-ArtBench (potentially less focused).



Architectures Used

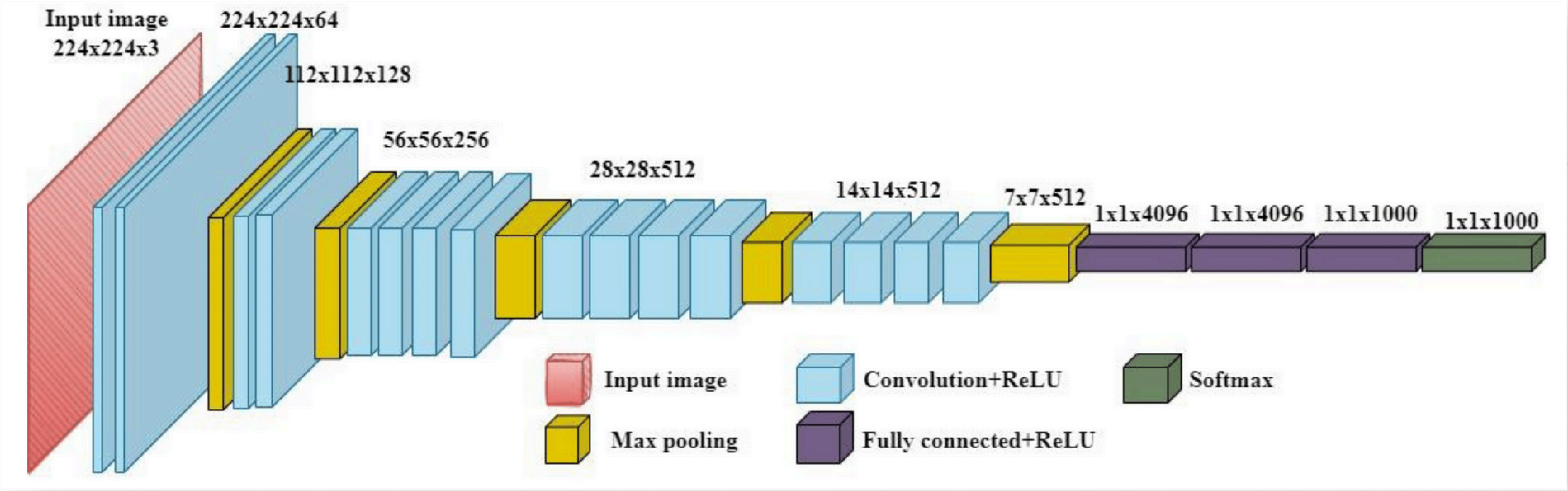
Convolutional Neural Networks (CNN)

- Convolutional Neural Networks, or **CNNs**, are a class of **deep learning** models particularly well-suited for **image classification** tasks.
- CNNs consist of several key components: **convolutional layers** that apply filters to capture spatial hierarchies in images, **pooling layers** that reduce dimensionality, and **fully connected layers** that perform the final classification.



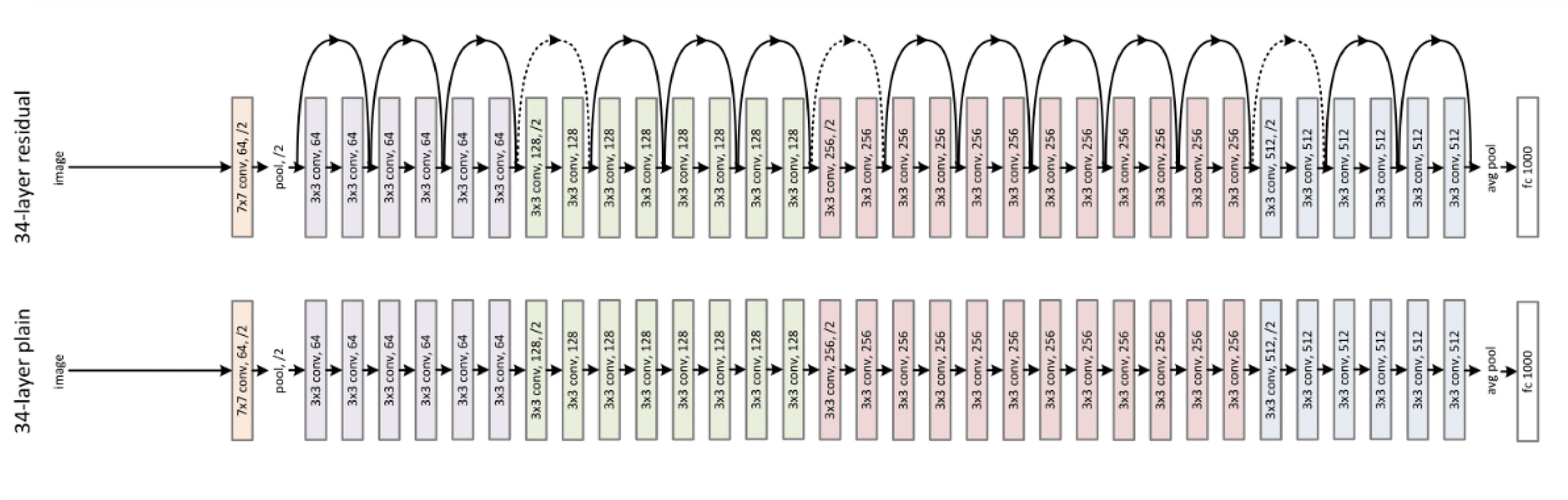
VGGNet

- **VGGNet** is a well-known deep learning model developed by the **Visual Geometry Group** at Oxford.
- Its architecture is characterized by its depth, typically **16 to 19** layers, and its use of small **3x3 convolution filters**. VGGNet's simplicity and uniform architecture make it **easy to implement** while providing **powerful** feature extraction capabilities.



ResNet

- **ResNet**, short for Residual Networks, introduced a groundbreaking innovation in deep learning: **residual blocks.**
- These connections allow the gradient to bypass certain layers, **effectively mitigating the vanishing gradient problem** that often faces deep networks.



DenseNet

- In **DenseNet**, each layer obtains **additional inputs** from **all preceding layers** and passes on its own feature-maps to **all subsequent layers**.
- **Concatenation** is used.

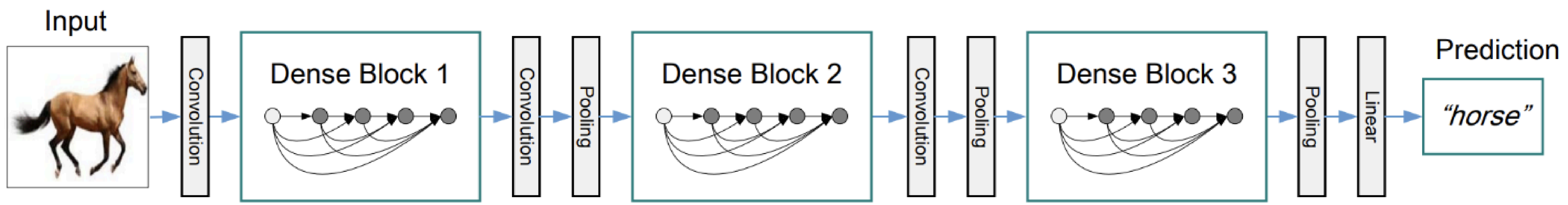


Figure 2: A deep DenseNet with three dense blocks. The layers between two adjacent blocks are referred to as transition layers and change feature-map sizes via convolution and pooling.

Data Preprocessing

Preprocessing Steps

Image Normalization

Rescaling pixel values to a common range (0 to 1) for faster training convergence.

Resizing Images

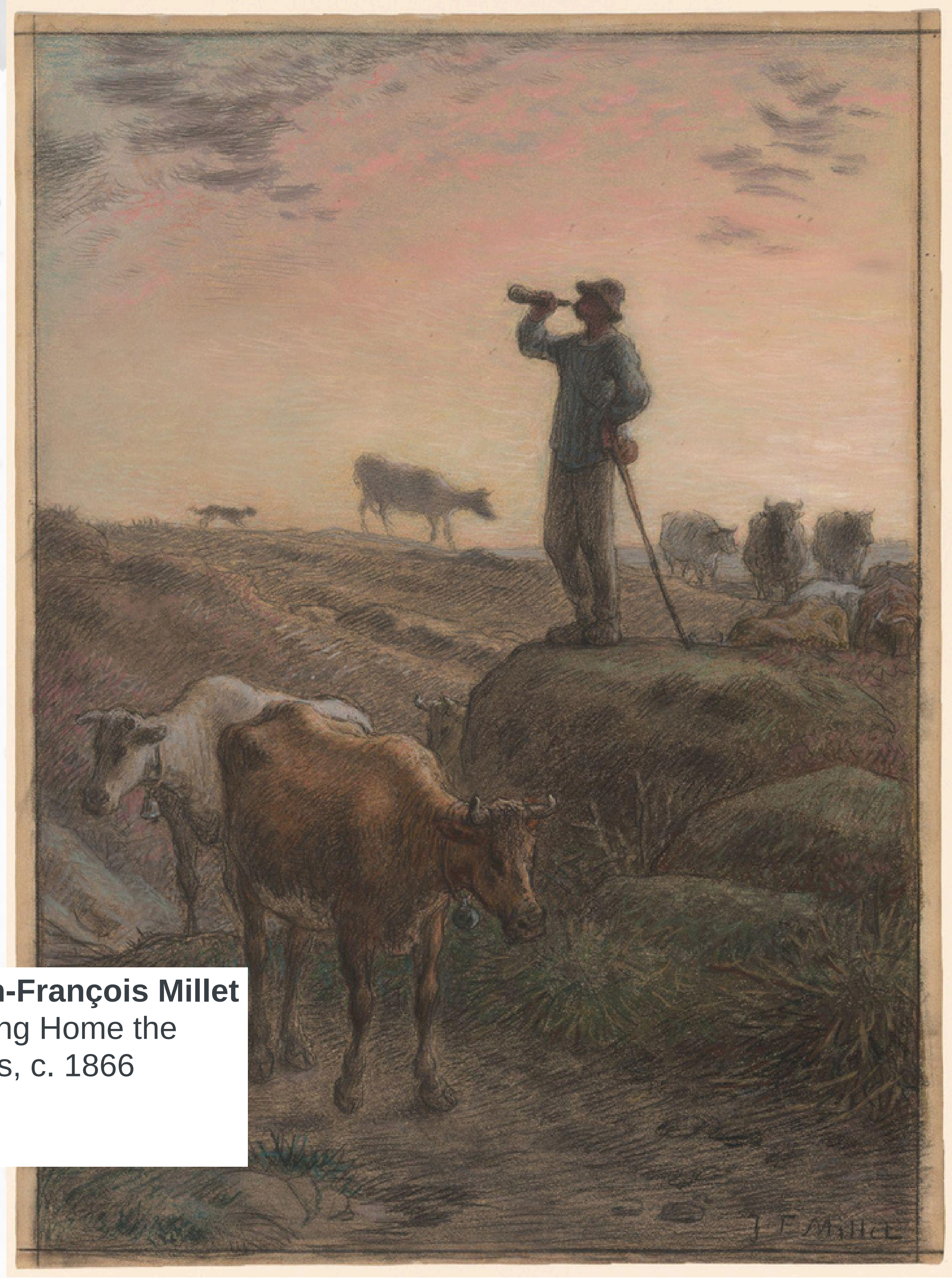
Resizing all images to 32x32 pixels.

Training Dataset Creation

Defining and configuring the training data using TensorFlow's Keras API.

Methodology Overview

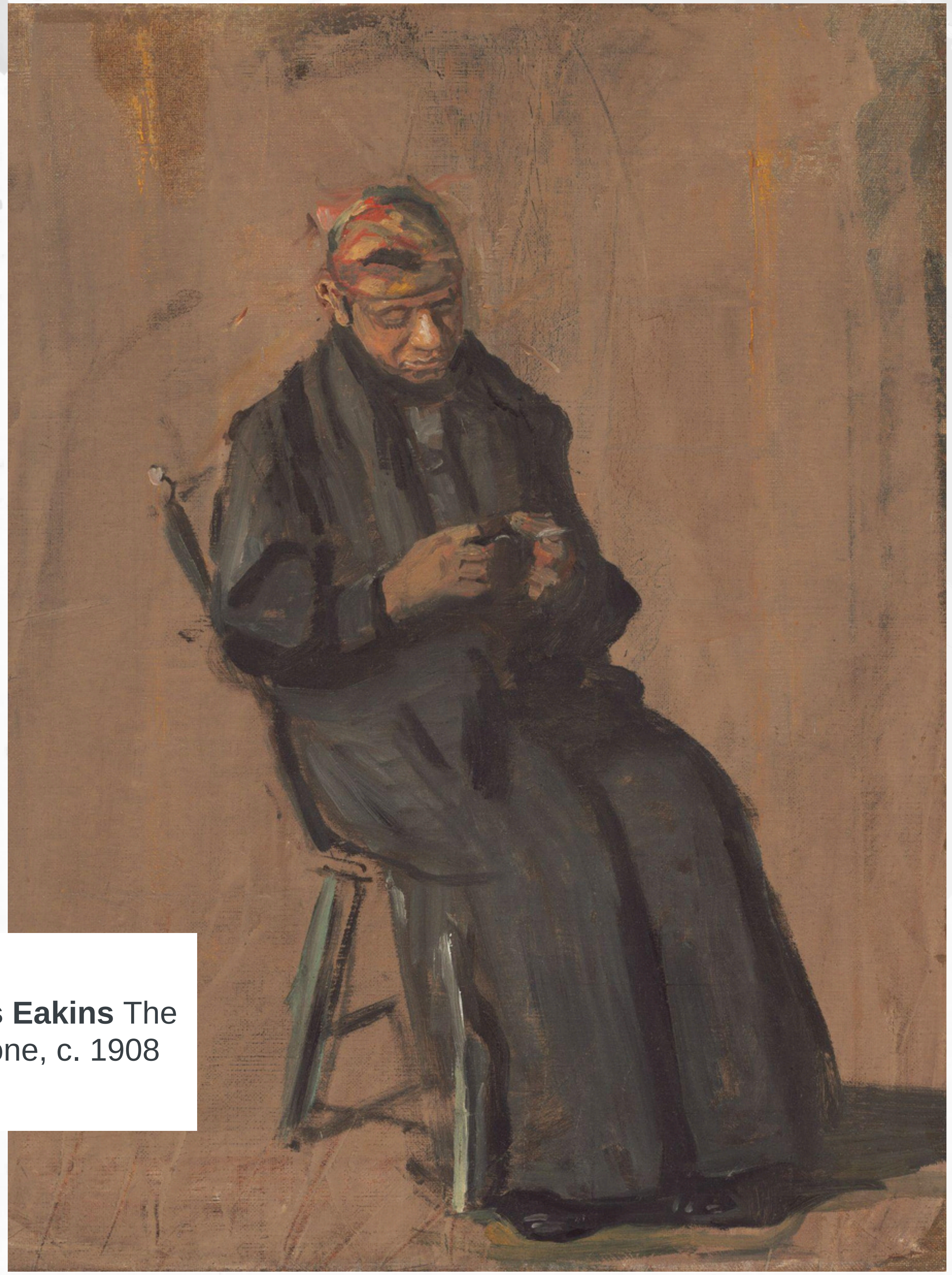
- First Training and Testing Phase
- Second Training and Testing Phase
- Evaluation and Results



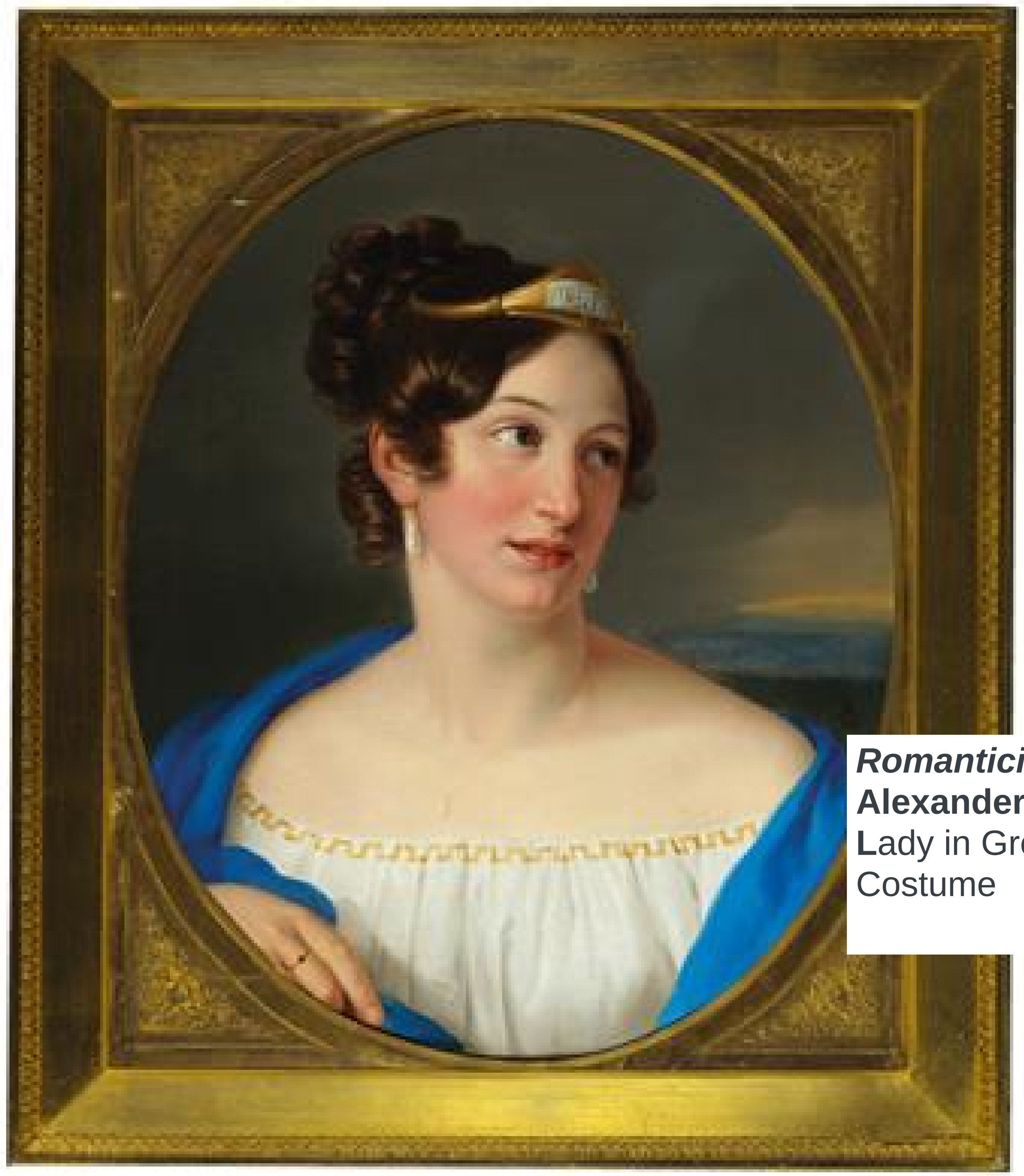
Jean-François Millet
Calling Home the Cows, c. 1866

Overview of Training Setup

- 30 Epochs
- Checkpointing
- Saving the best model



Thomas Eakins The Chaperone, c. 1908

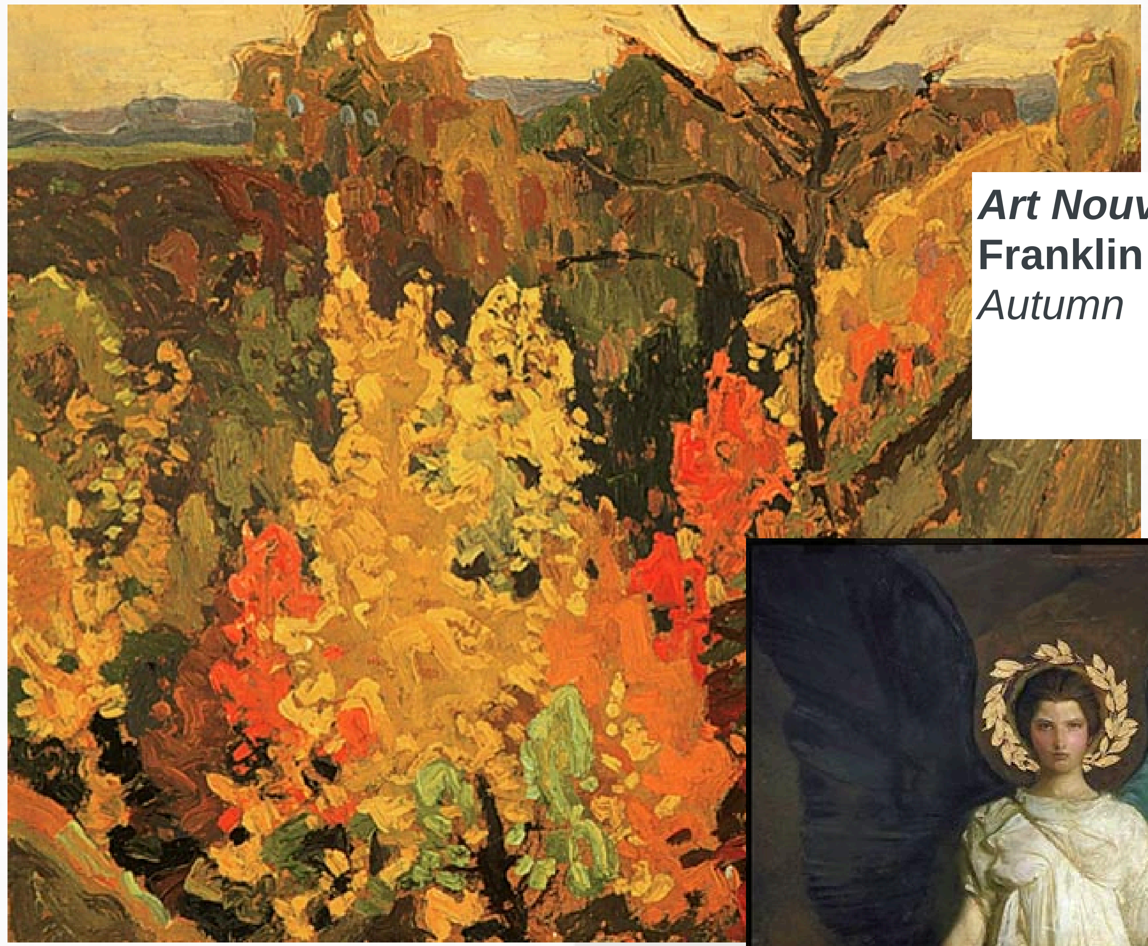


Romanticism
Alexander Clarot
Lady in Greek Costume

First Training & Testing phase

Training each style separately

Ranking Styles
Expectations



Art Nouveau
Franklin Carmichael
Autumn

Realism
Abbott Handerson Thayer
A Winged Figure

1. Realism
2. Renaissance
3. Ukiyo-e
4. Baroque
5. Post-Impressionism
6. Impressionism
7. Romanticism
8. Expressionism
9. Surrealism
10. Art Nouveau

First Training Phase

CNN architecture

64 filters 3x3 size ReLU	32 and 16 units ReLU	2 units Softmax	113,042
Convolutional Layer	Fully connected Layer	Output Layer	Parameters

Compilation: Adam optimizer, binary cross-entropy loss

First Training Phase

CNN architecture

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 30, 30, 64)	1,792
max_pooling2d (MaxPooling2D)	(None, 15, 15, 64)	0
conv2d_1 (Conv2D)	(None, 13, 13, 64)	36,928
max_pooling2d_1 (MaxPooling2D)	(None, 6, 6, 64)	0
flatten (Flatten)	(None, 2304)	0
dense (Dense)	(None, 32)	73,760
dense_1 (Dense)	(None, 16)	528
dense_2 (Dense)	(None, 2)	34

Table 5.1: Summary of the CNN model architecture

First Testing Phase

Model Performance

Accuracy

Precision

Recall

F1-Score

Table 5.2: Classification Reports for Different Models

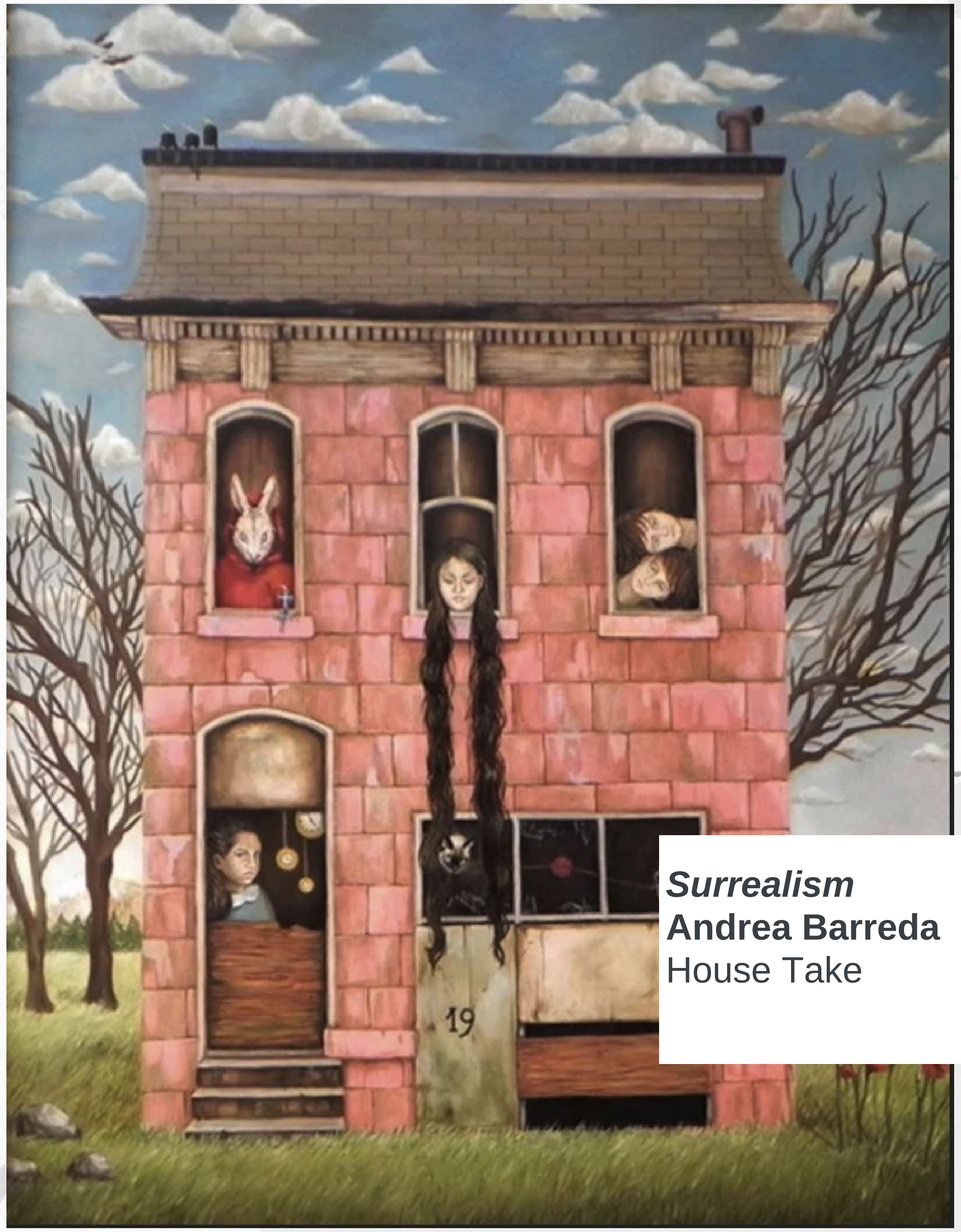
	Fake			Real			Overall
Art Style	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Acc.**
Ukiyo-e	**0.99**	**0.99**	**0.99**	**0.97**	**0.98**	**0.98**	**0.98**
Art Nouveau	**0.99**	0.98	0.98	0.96	0.97	0.97	**0.98**
Impressionism	0.98	**0.99**	0.98	**0.97**	0.95	0.96	**0.98**
Baroque	0.97	0.98	0.97	0.96	0.93	0.95	0.97
Expressionism	0.97	0.96	0.97	0.93	0.94	0.94	0.96
Post-Impressionism	0.97	0.97	0.97	0.94	0.94	0.94	0.96
Renaissance	0.95	0.98	0.96	0.95	0.89	0.92	0.95
Realism	0.94	0.96	0.95	0.92	0.88	0.90	0.93
Romanticism	0.93	0.97	0.95	0.93	0.85	0.89	0.93
Surrealism	0.94	0.93	0.93	0.86	0.88	0.87	0.91

Ranking Styles

Expectations vs Results

1. Realism
2. Renaissance
3. Ukiyo-e
4. Baroque
5. Post-Impressionism
6. Impressionism
7. Romanticism
8. Expressionism
9. Surrealism
10. Art Nouveau

1. Ukiyo-e
2. Art Nouveau
3. Impressionism
4. Baroque
5. Expressionism
6. Post-Impressionism
7. Renaissance
8. Realism
9. Romanticism
10. Surrealism

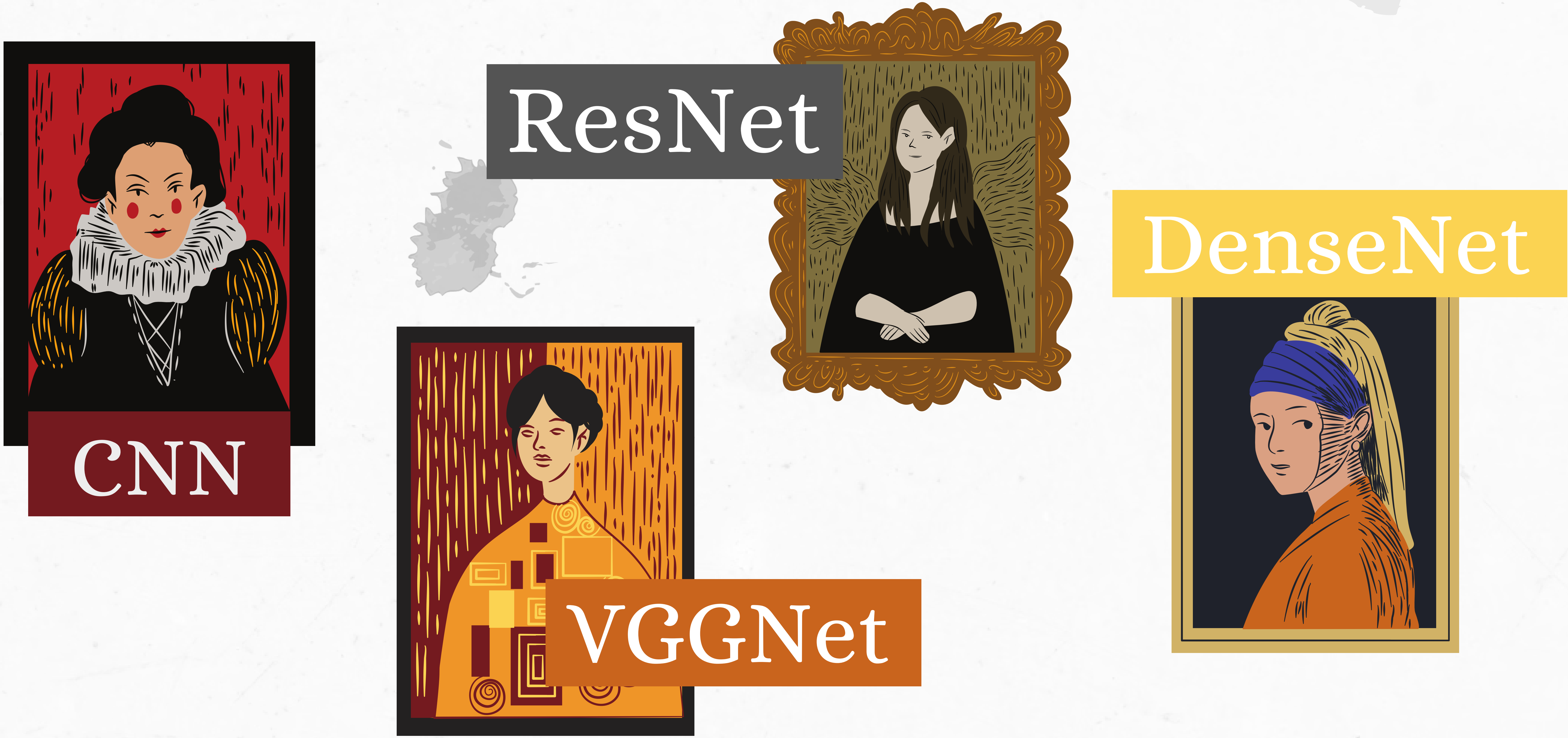


Surrealism
Andrea Barreda
House Take

Second Training & Testing phase

Using the whole Dataset

Model architectures



Second Training Phase

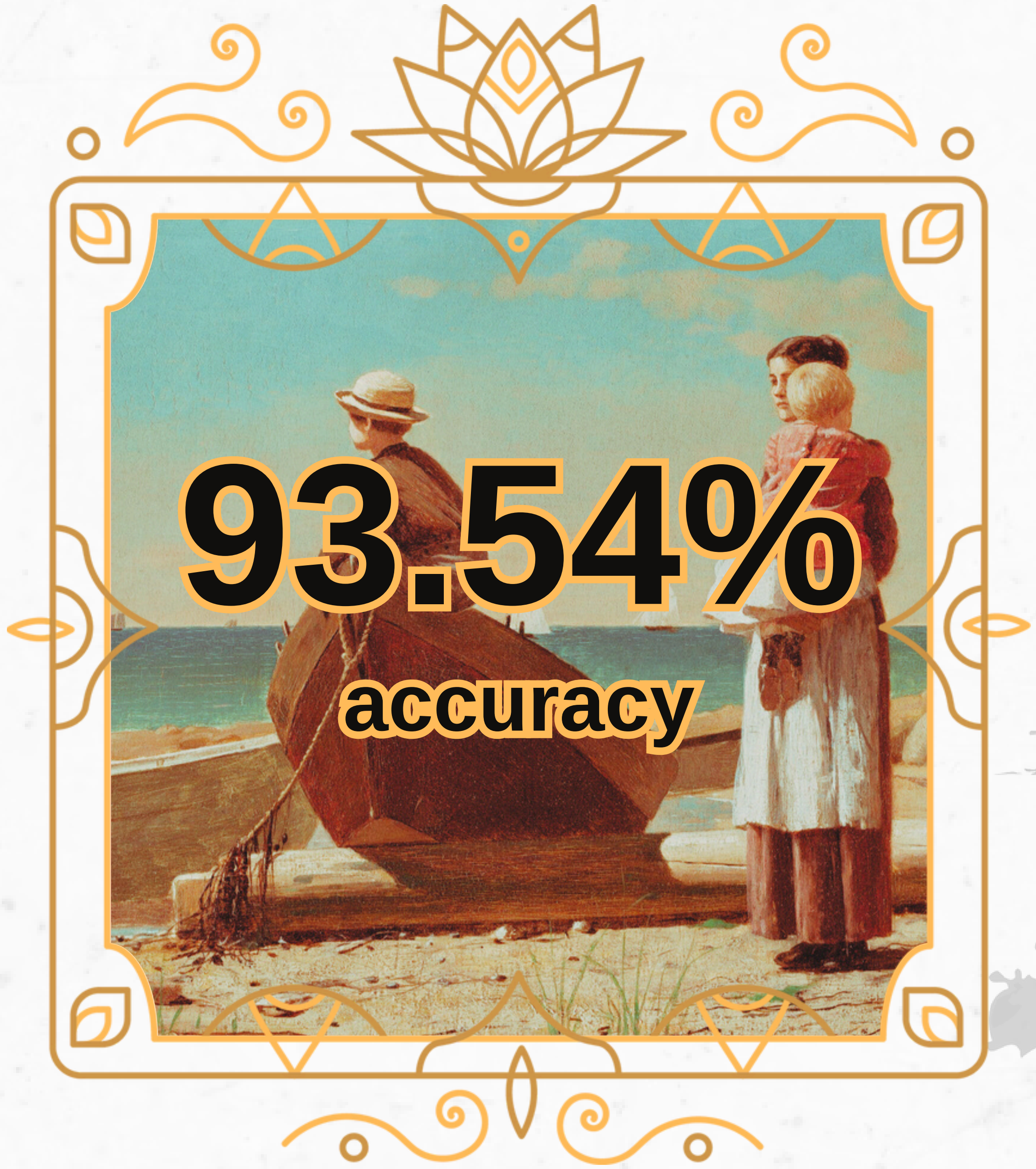
Second CNN architecture

Layer (type)	Output Shape	Param #
conv2d_2 (Conv2D)	(None, 30, 30, 512)	14,336
max_pooling2d_2 (MaxPooling2D)	(None, 15, 15, 512)	0
conv2d_3 (Conv2D)	(None, 13, 13, 128)	589,952
max_pooling2d_3 (MaxPooling2D)	(None, 6, 6, 128)	0
conv2d_4 (Conv2D)	(None, 4, 4, 32)	36,896
max_pooling2d_4 (MaxPooling2D)	(None, 2, 2, 32)	0
flatten_1 (Flatten)	(None, 128)	0
dense_3 (Dense)	(None, 32)	4,128
dense_4 (Dense)	(None, 16)	528
dense_5 (Dense)	(None, 2)	34

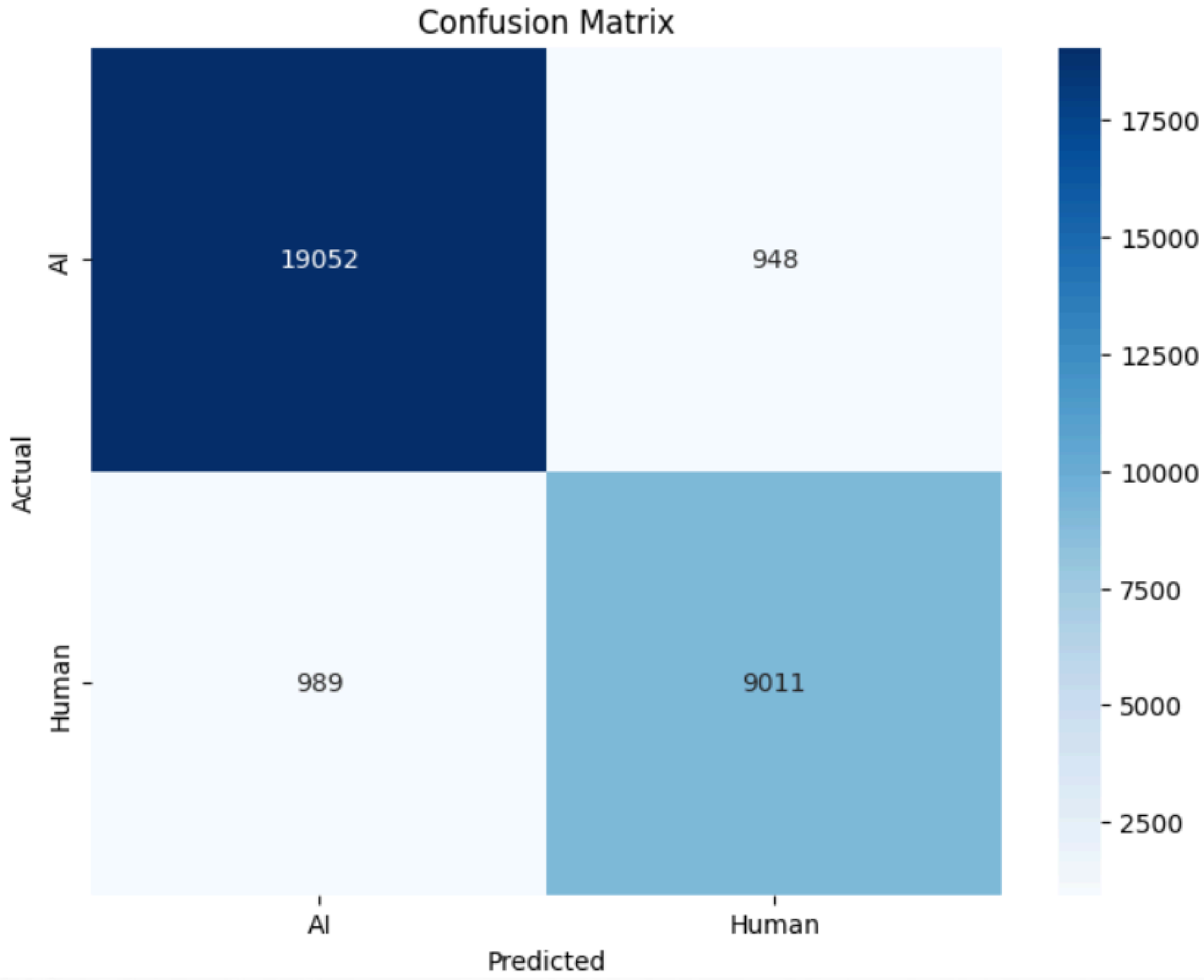
Table 6.2: Summary of the Second CNN model architecture

Results

CNN First Model

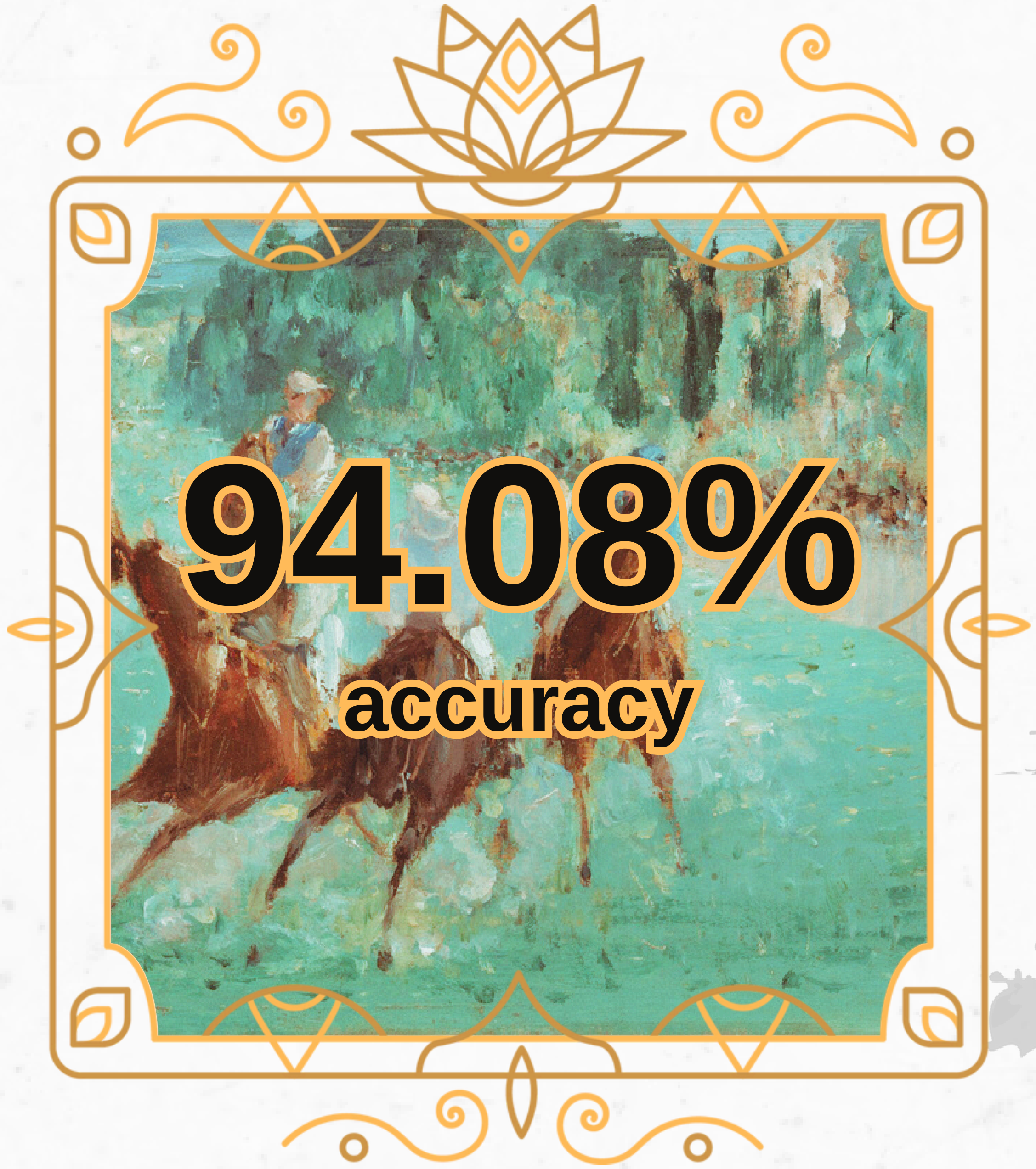


	precision	recall	f1-score	support
0	0.95	0.95	0.95	20000
1	0.90	0.90	0.90	10000
accuracy			0.94	30000
macro avg	0.93	0.93	0.93	30000
weighted avg	0.94	0.94	0.94	30000

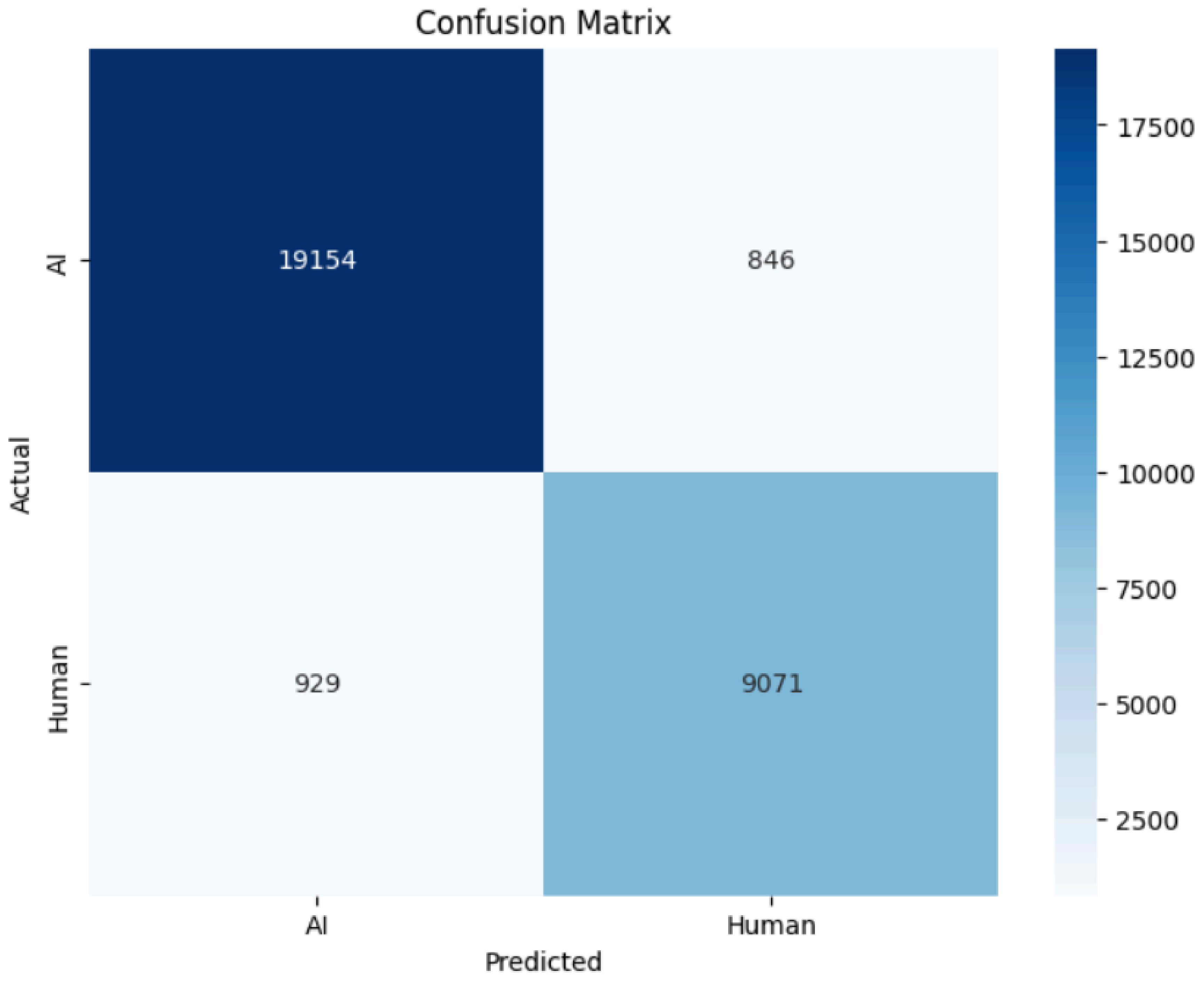


Results

CNN Second Model

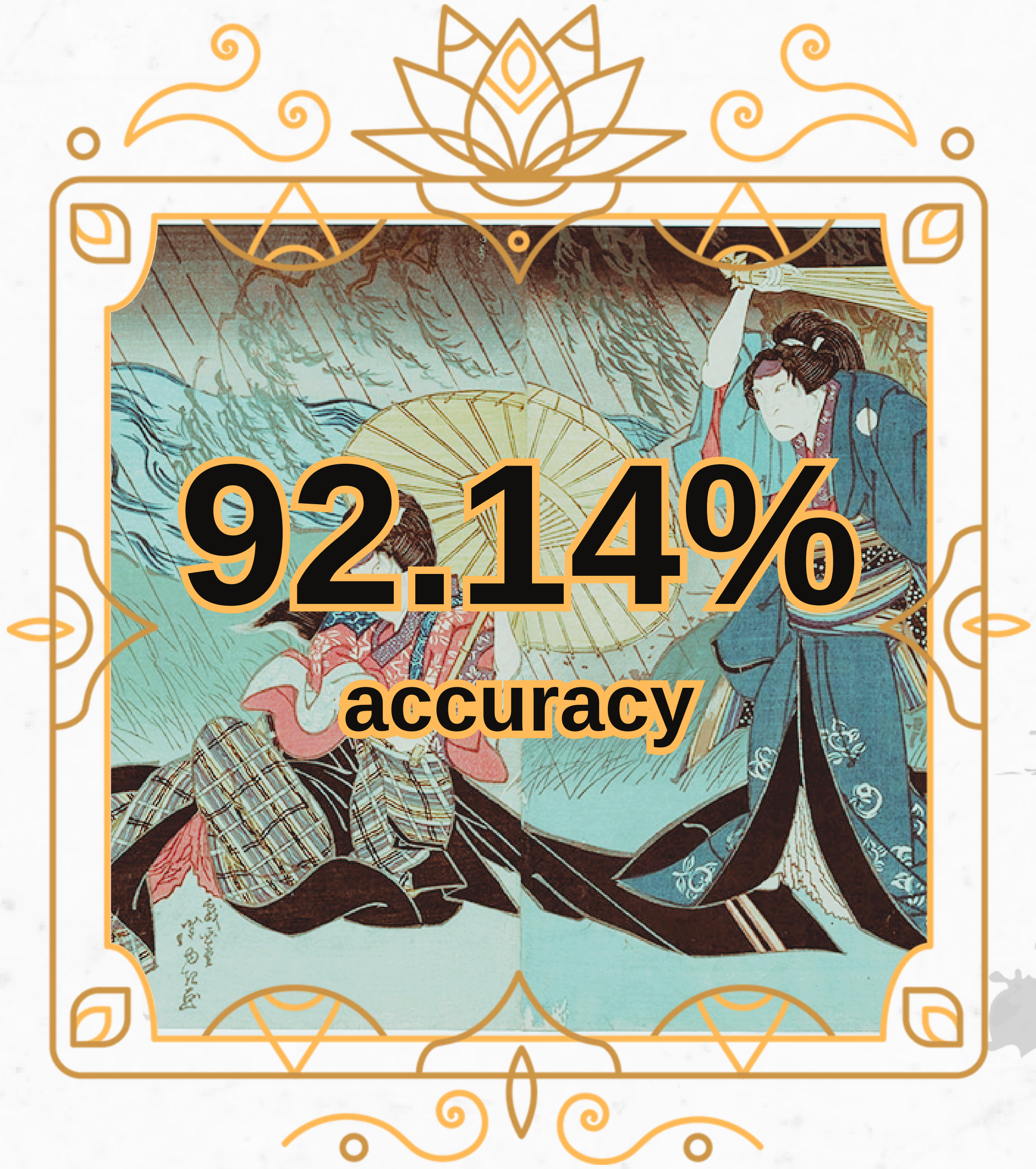


	precision	recall	f1-score	support
0	0.95	0.96	0.96	20000
1	0.91	0.91	0.91	10000
accuracy			0.94	30000
macro avg	0.93	0.93	0.93	30000
weighted avg	0.94	0.94	0.94	30000

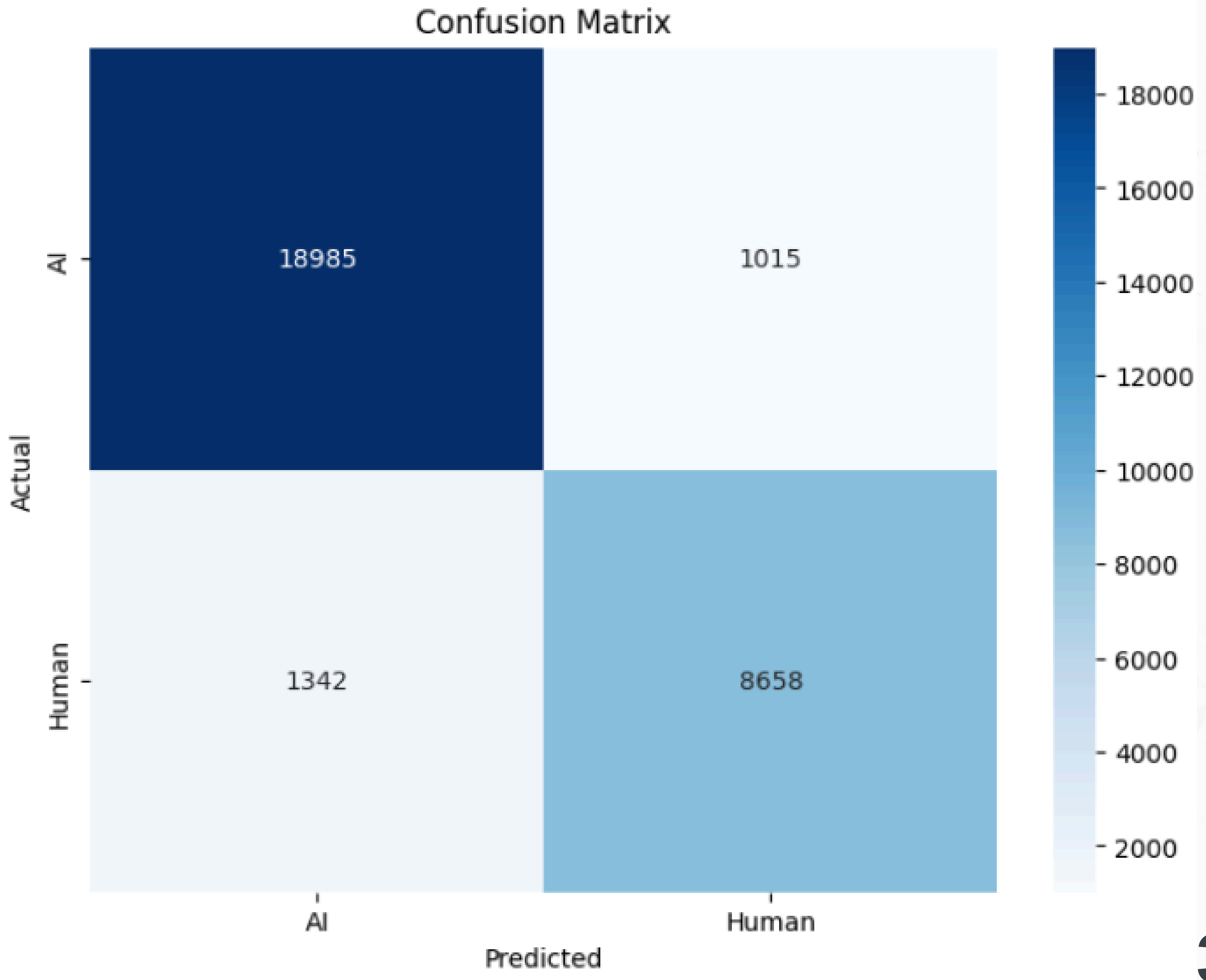


Results

VGGNet 19

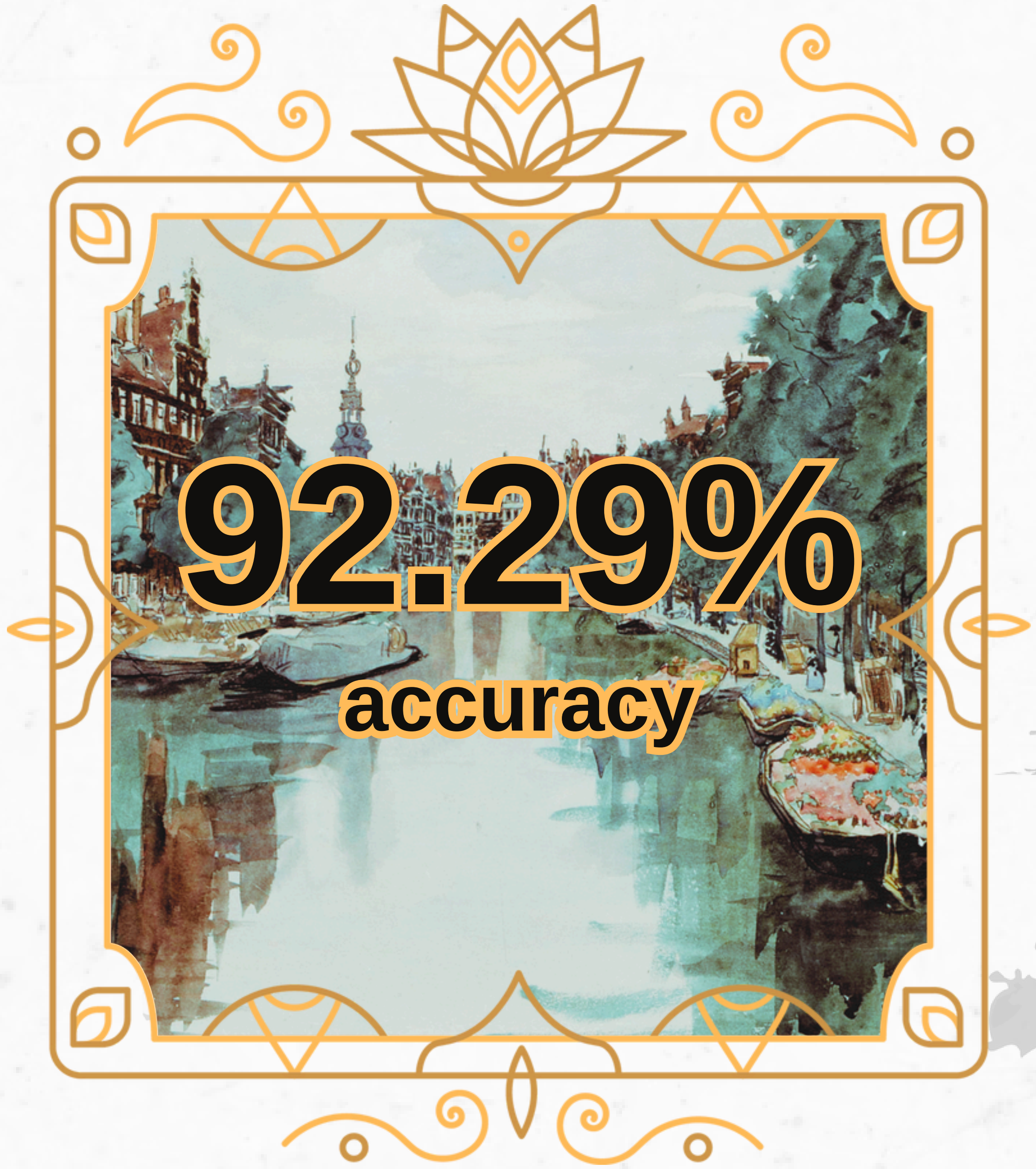


	precision	recall	f1-score	support
0	0.93	0.95	0.94	20000
1	0.90	0.87	0.88	10000
accuracy			0.92	30000
macro avg	0.91	0.91	0.91	30000
weighted avg	0.92	0.92	0.92	30000

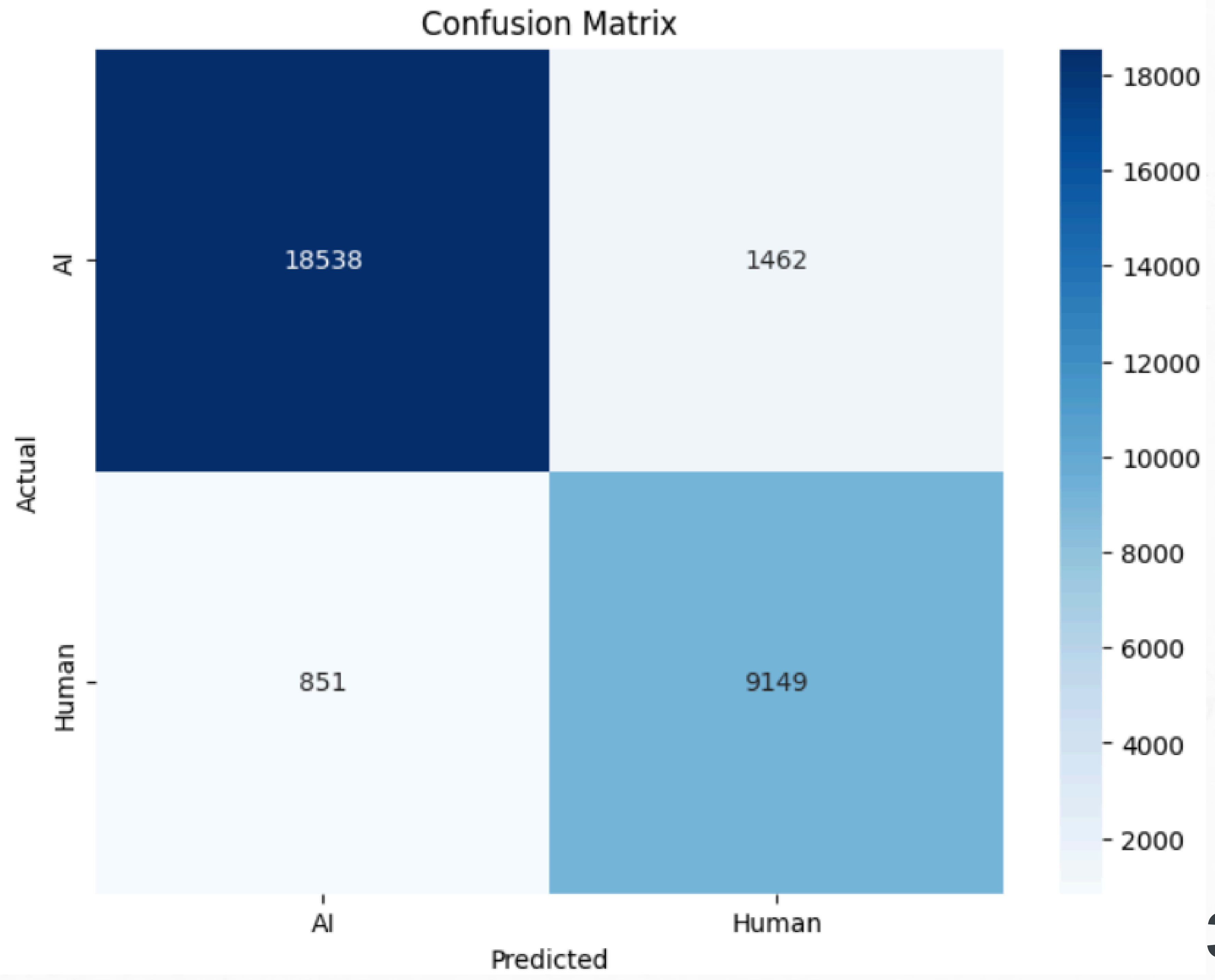


Results

ResNet 34

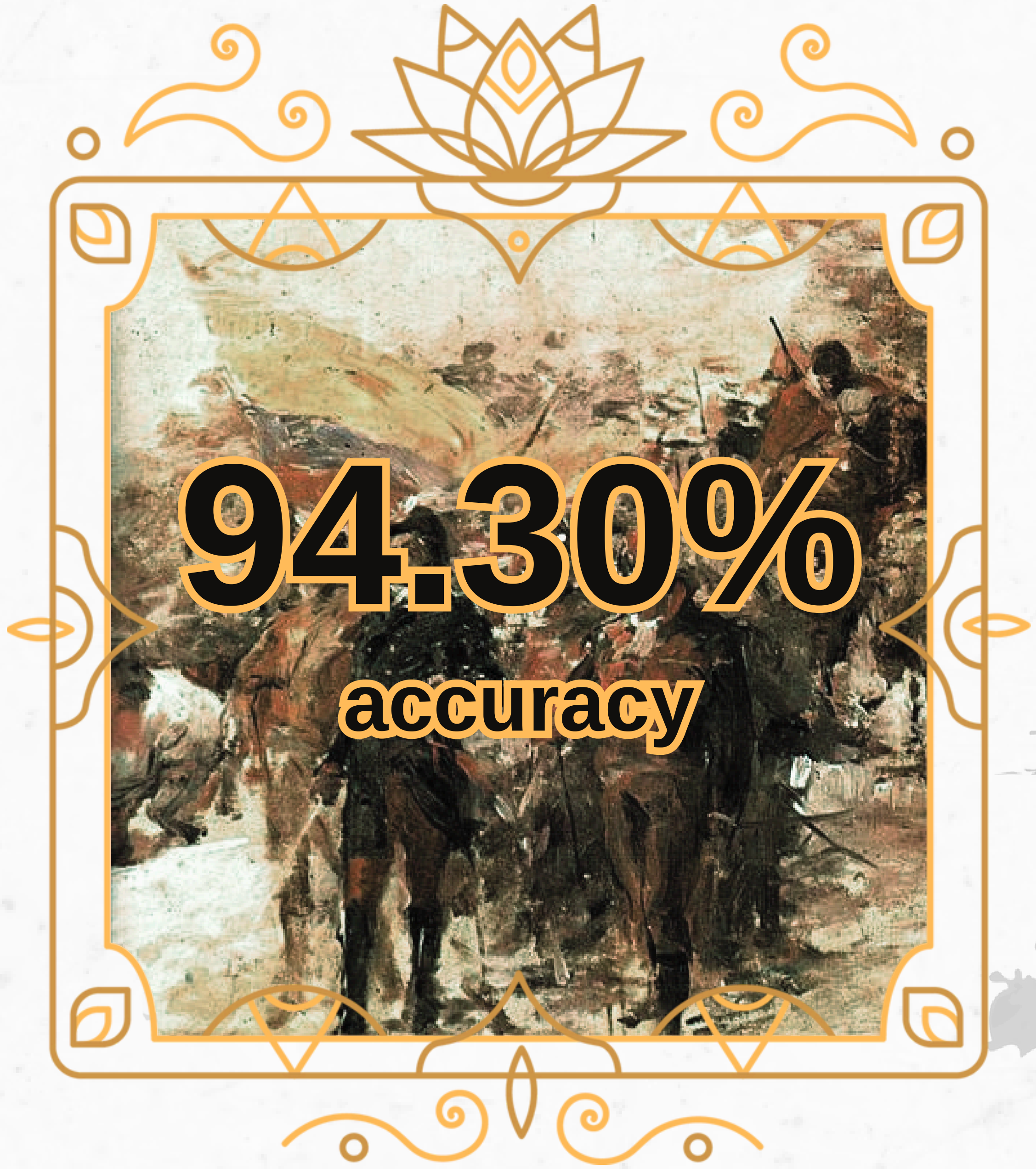


	precision	recall	f1-score	support
0	0.96	0.93	0.94	20000
1	0.86	0.91	0.89	10000
accuracy			0.92	30000
macro avg	0.91	0.92	0.91	30000
weighted avg	0.92	0.92	0.92	30000

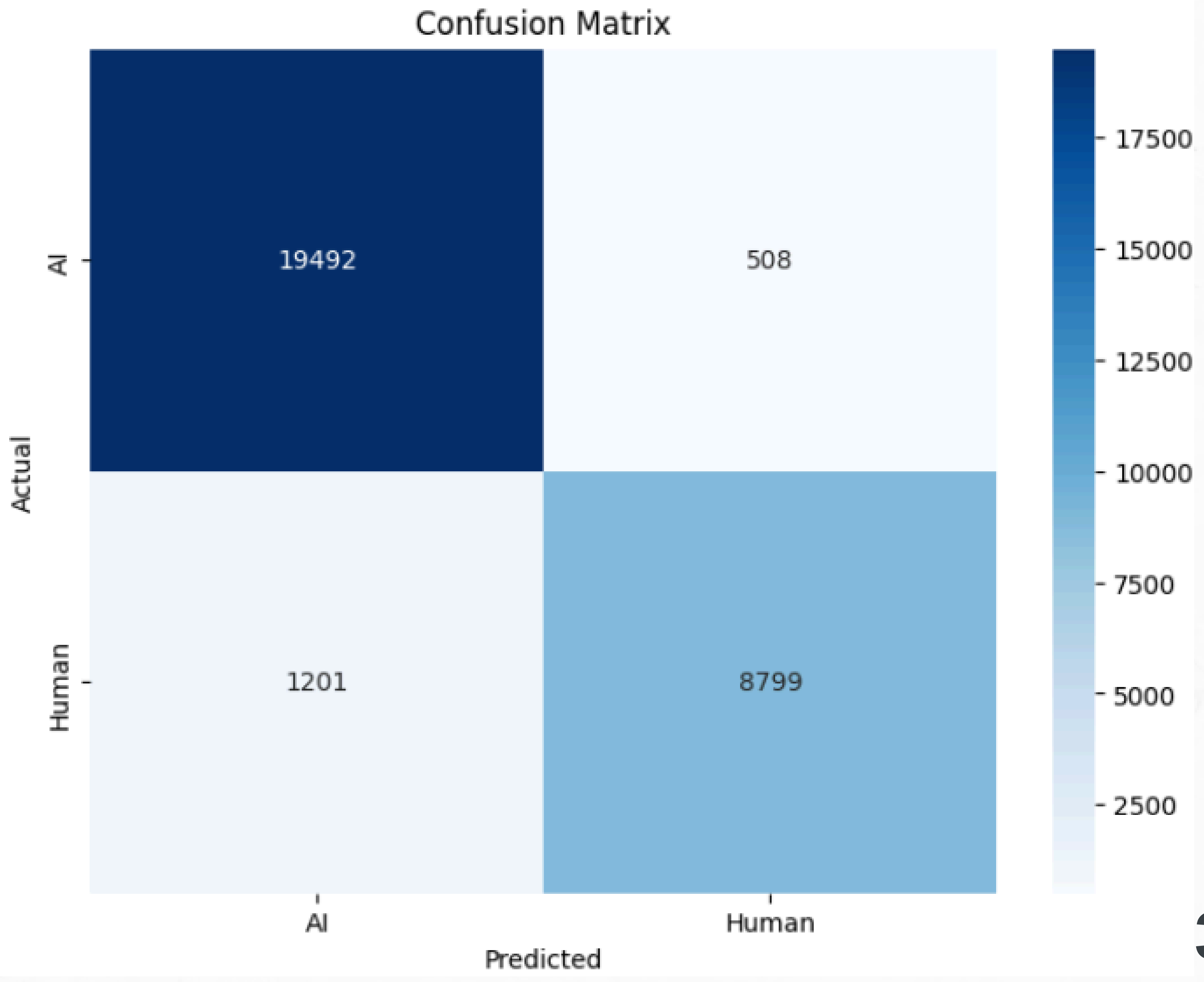


Results

DenseNet 121



	precision	recall	f1-score	support
0	0.94	0.97	0.96	20000
1	0.95	0.88	0.91	10000
accuracy			0.94	30000
macro avg	0.94	0.93	0.93	30000
weighted avg	0.94	0.94	0.94	30000



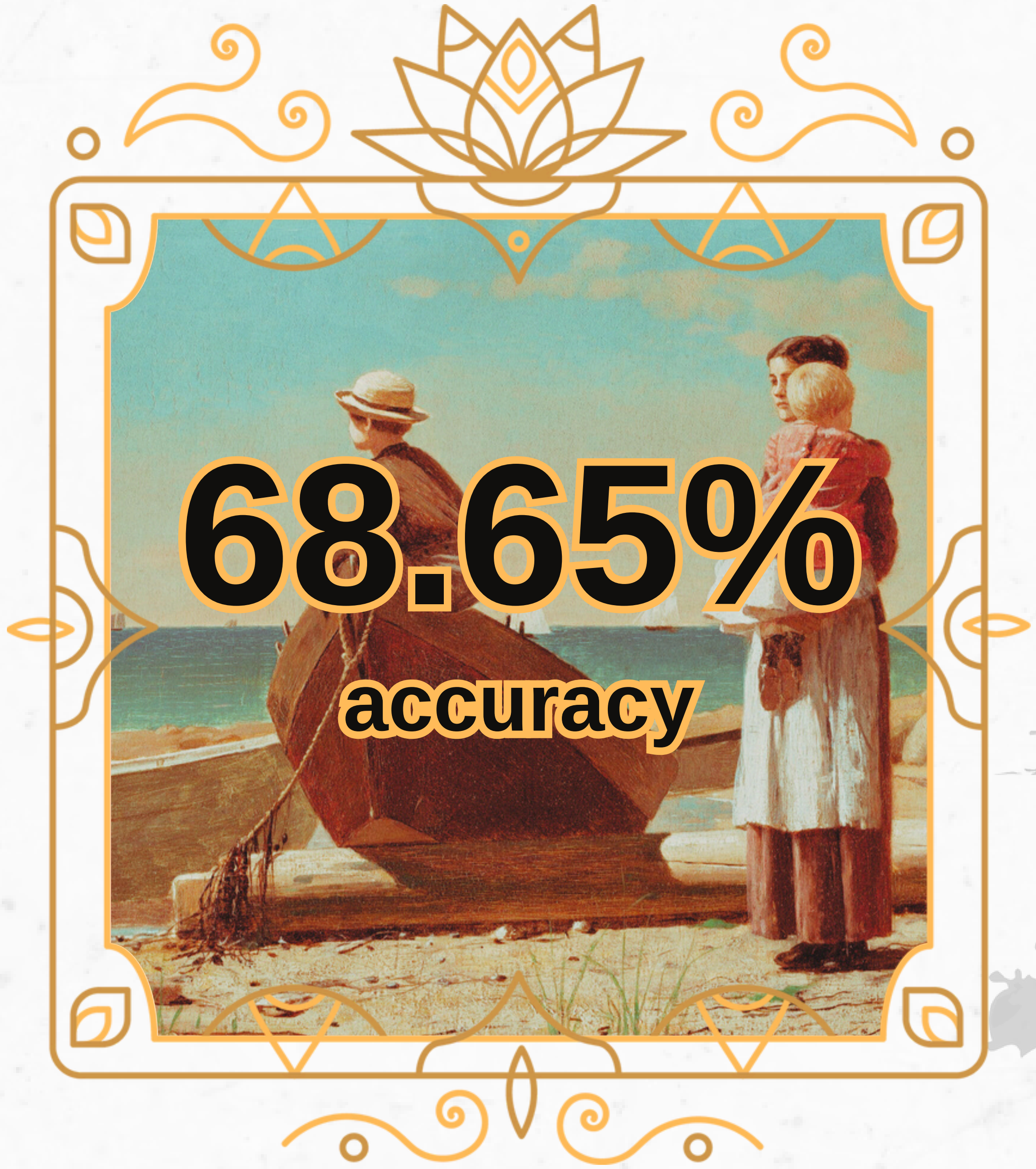
Thomas Eakins The Biglin Brothers Racing, 1872

Extra Testing

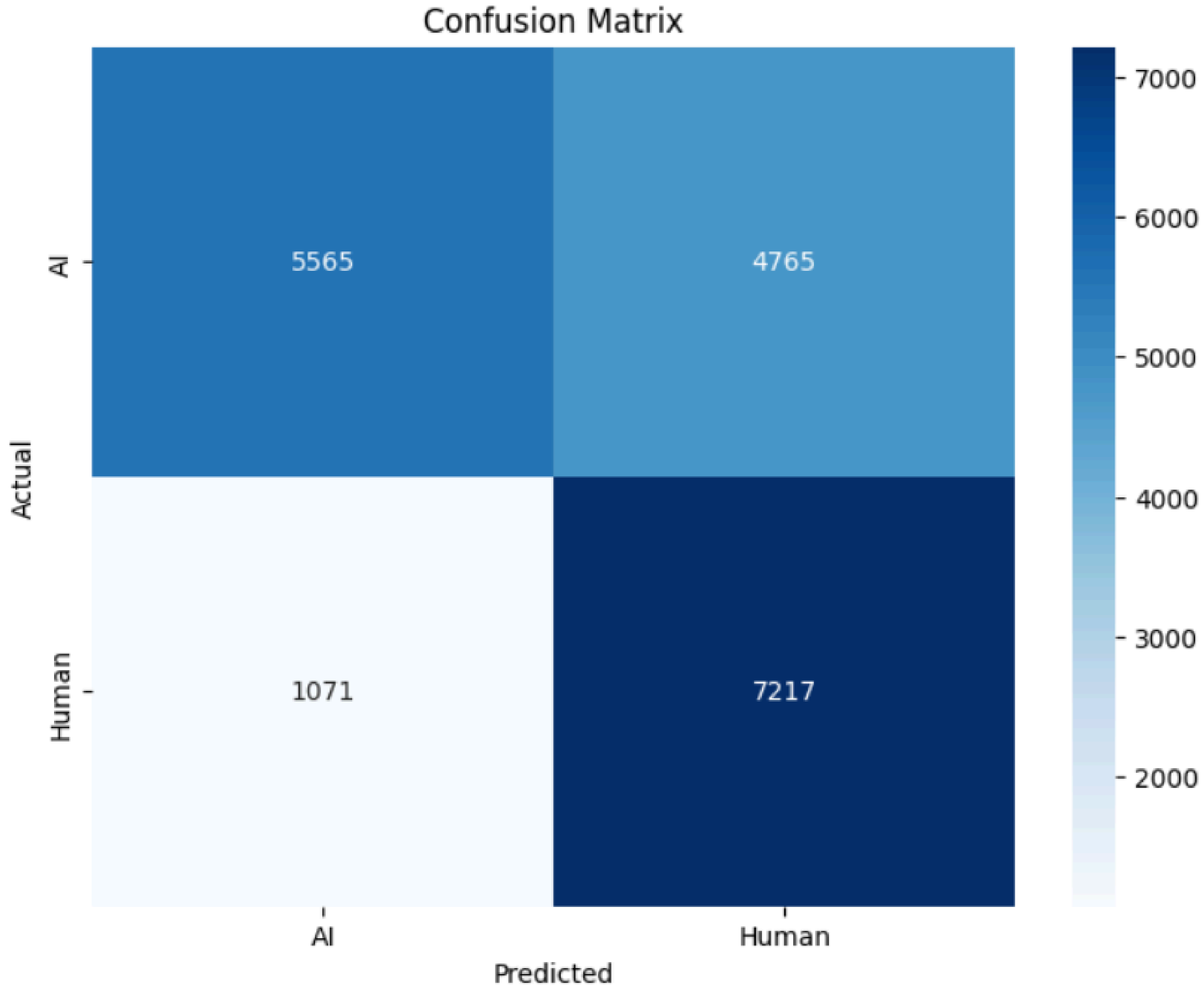
Testing the same models on the second dataset without retraining

Results

CNN First Model

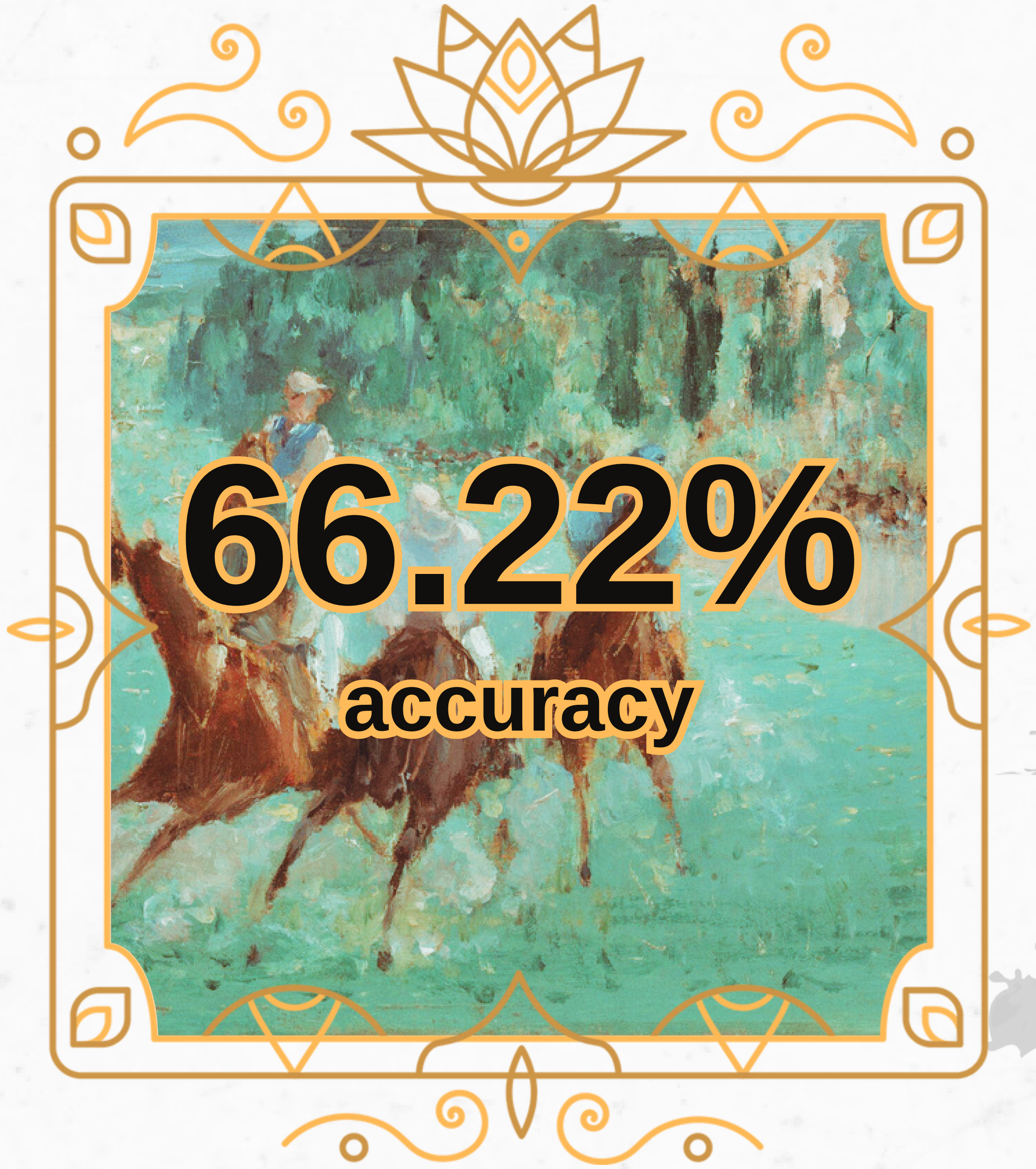


	precision	recall	f1-score	support
0	0.84	0.54	0.66	10330
1	0.60	0.87	0.71	8288
accuracy			0.69	18618
macro avg	0.72	0.70	0.68	18618
weighted avg	0.73	0.69	0.68	18618

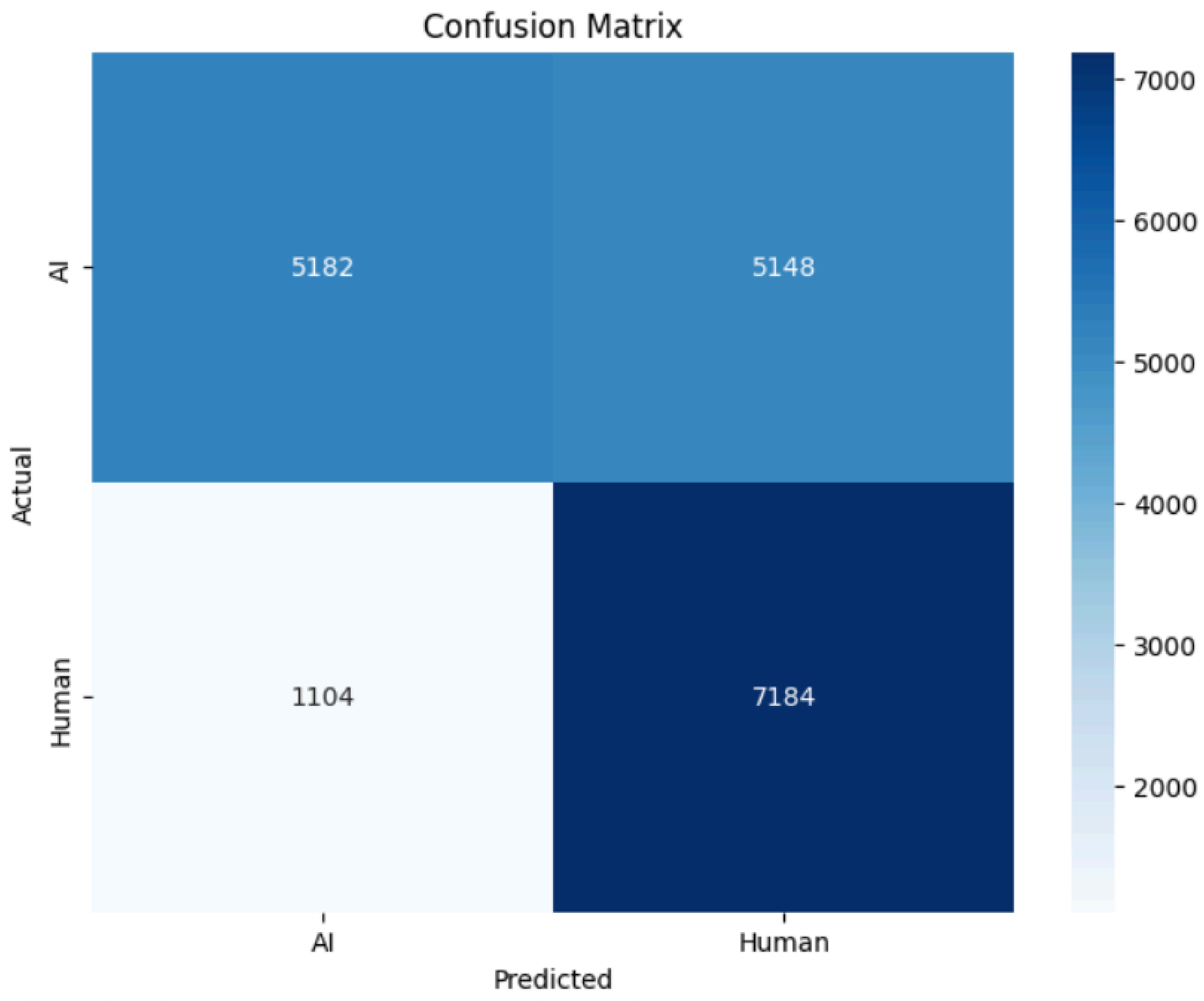


Results

CNN Second Model

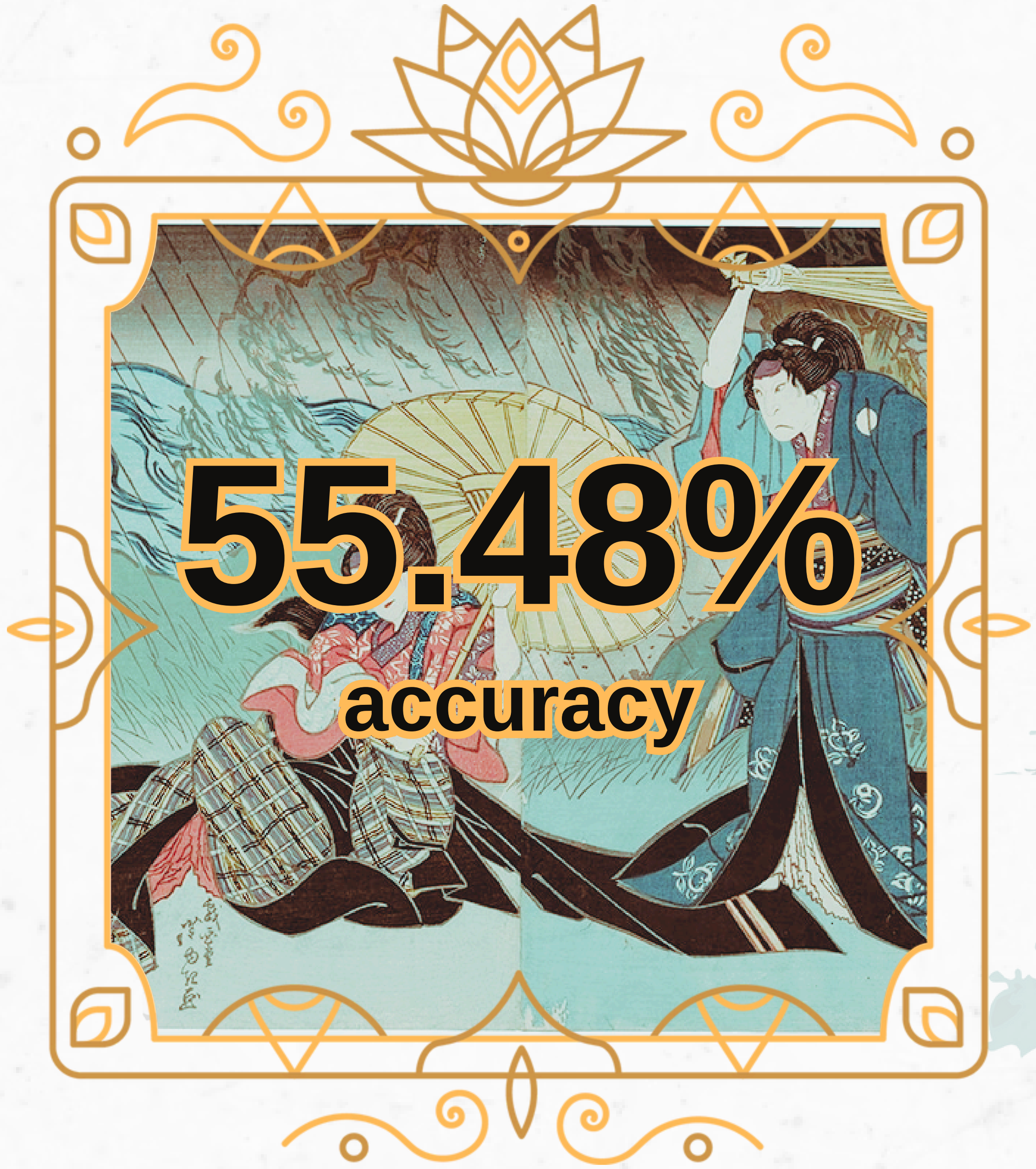


	precision	recall	f1-score	support
0	0.82	0.50	0.62	10330
1	0.58	0.87	0.70	8288
accuracy			0.66	18618
macro avg	0.70	0.68	0.66	18618
weighted avg	0.72	0.66	0.66	18618

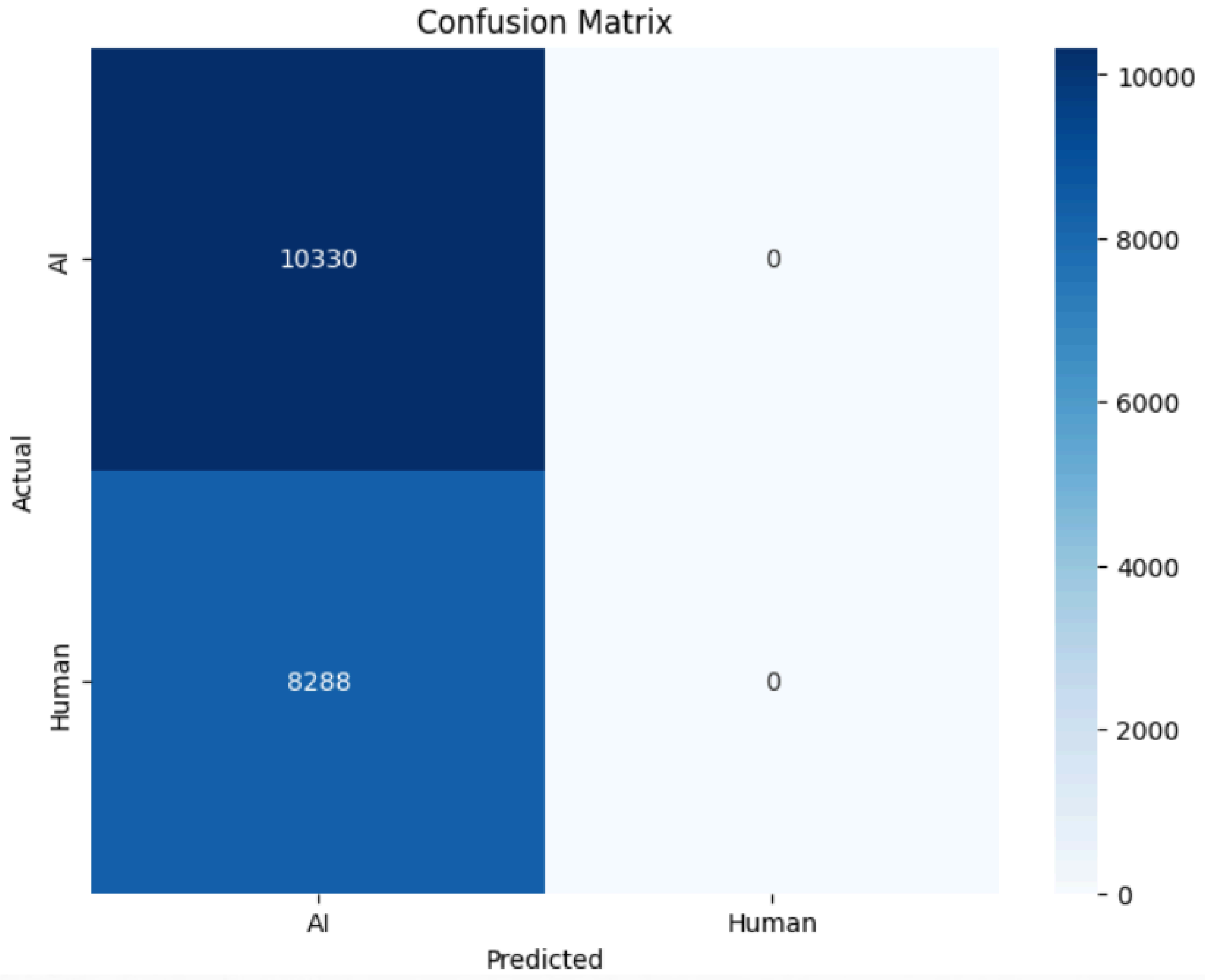


Results

VGGNet 19

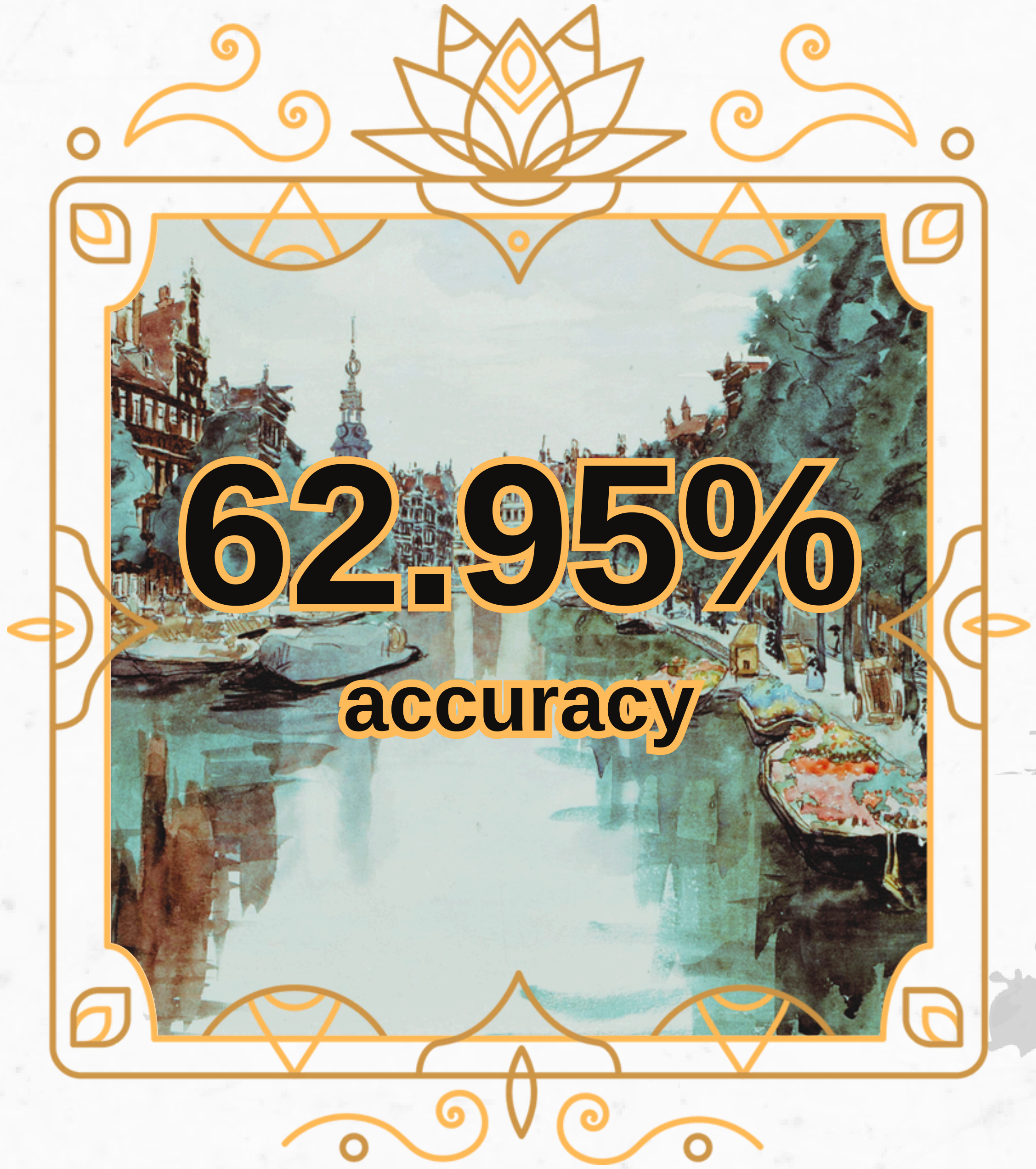


	precision	recall	f1-score	support
0	0.55	1.00	0.71	10330
1	0.00	0.00	0.00	8288
accuracy			0.55	18618
macro avg	0.28	0.50	0.36	18618
weighted avg	0.31	0.55	0.40	18618

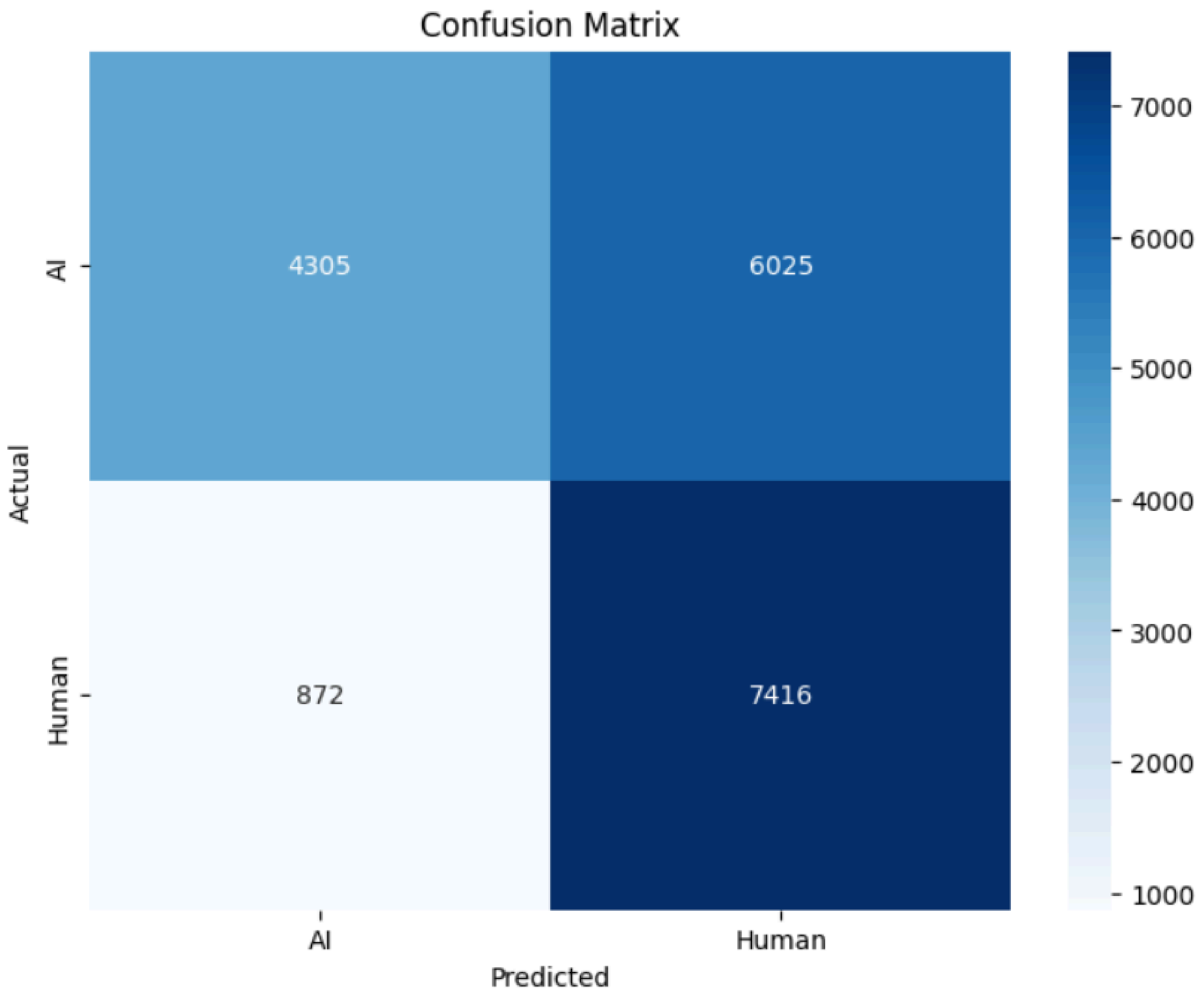


Results

ResNet 34

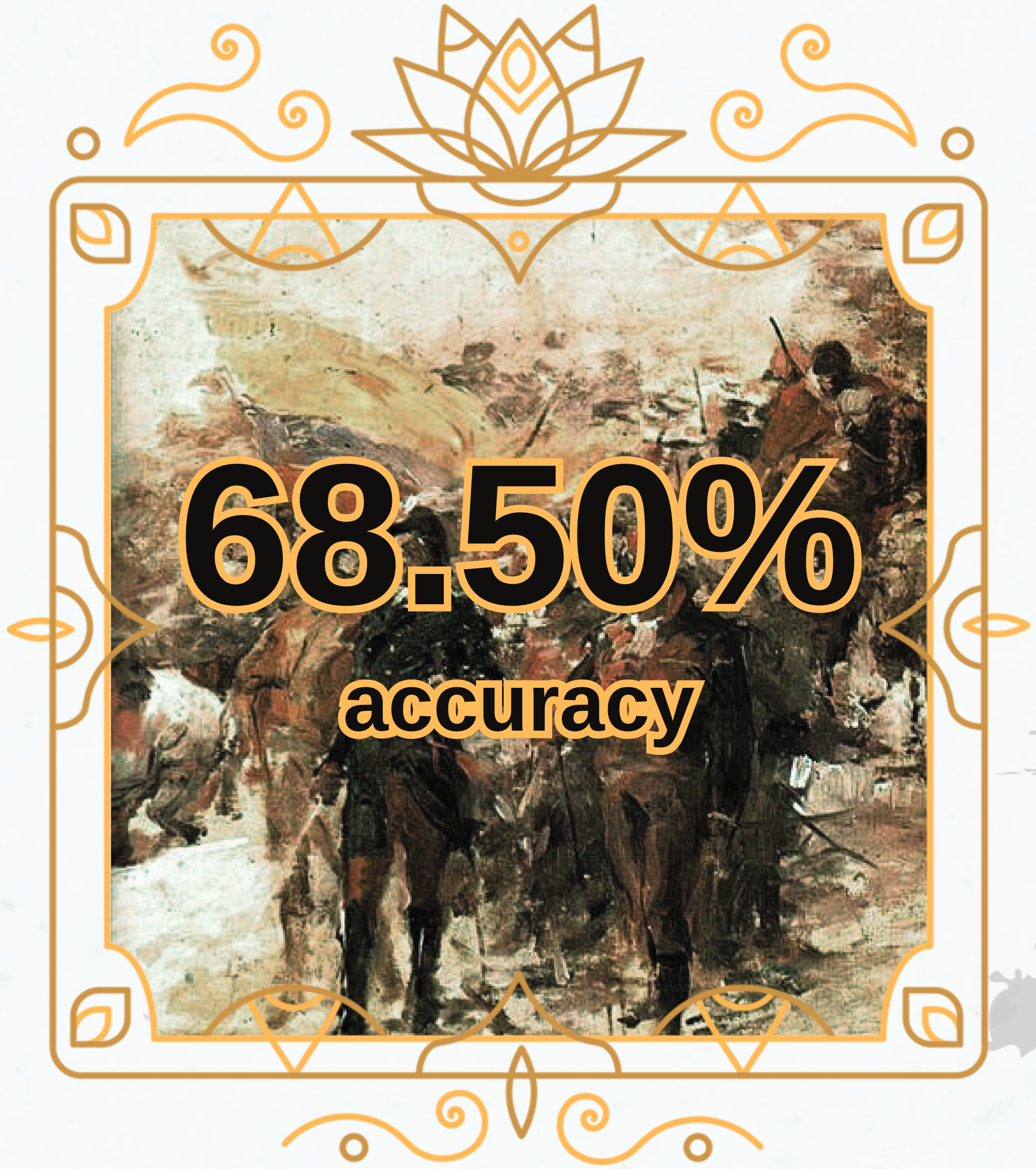


	precision	recall	f1-score	support
0	0.83	0.42	0.56	10330
1	0.55	0.89	0.68	8288
accuracy			0.63	18618
macro avg	0.69	0.66	0.62	18618
weighted avg	0.71	0.63	0.61	18618

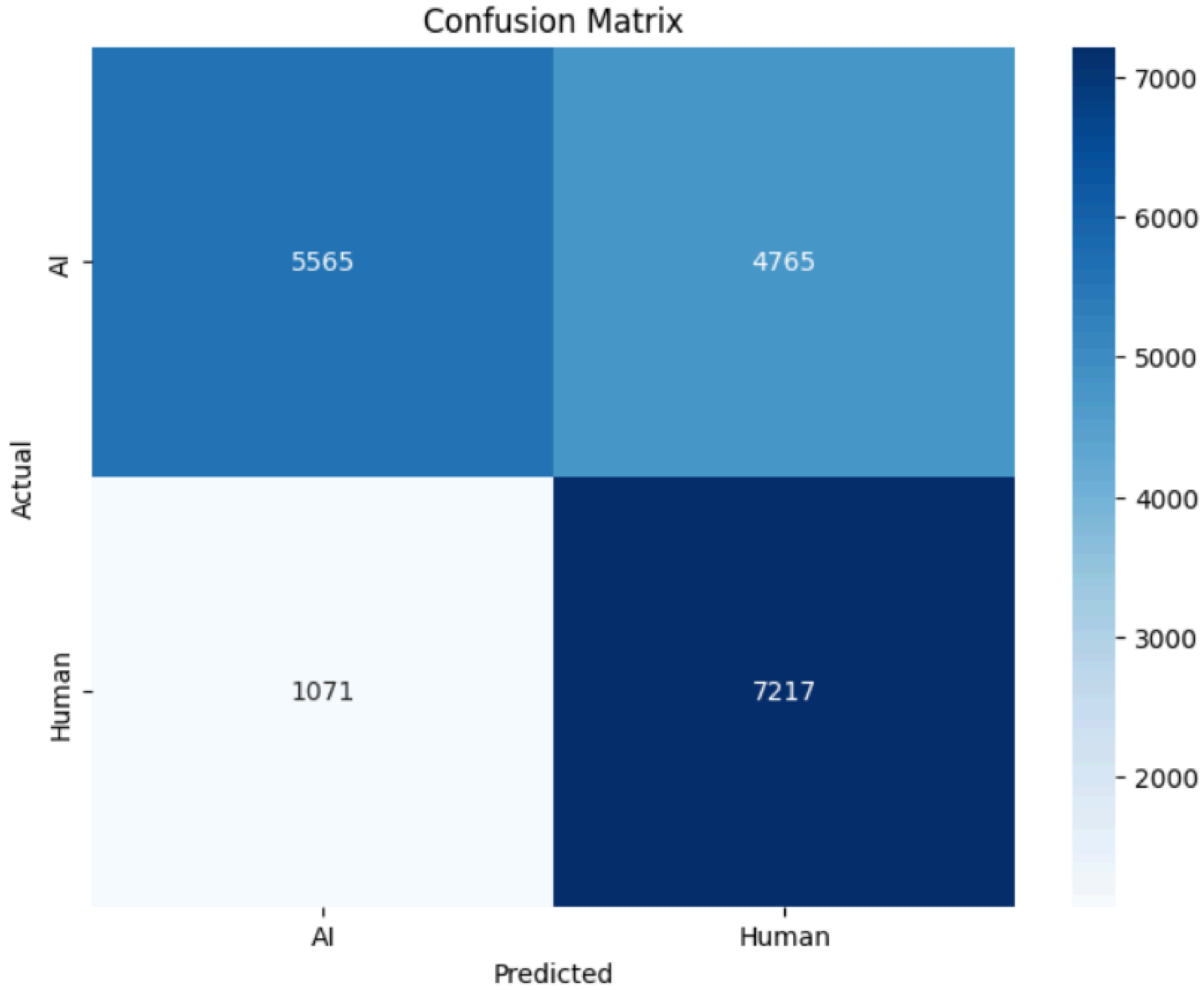


Results

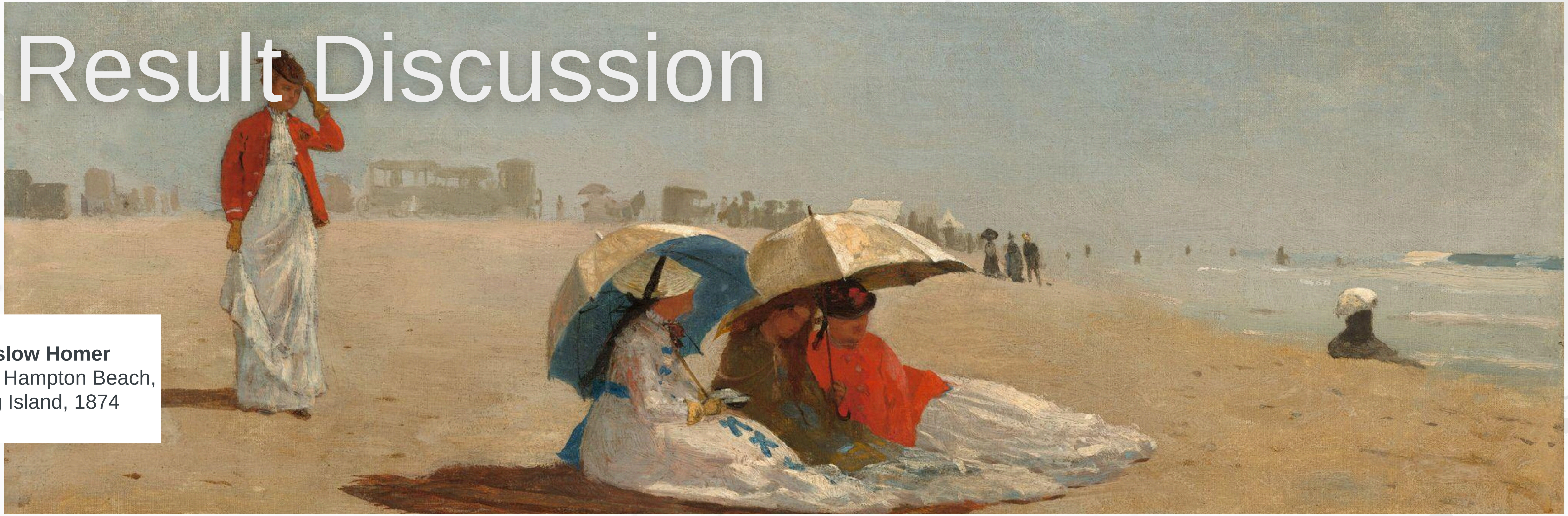
DenseNet 121



	precision	recall	f1-score	support
0	0.84	0.54	0.66	10330
1	0.60	0.87	0.71	8288
accuracy			0.69	18618
macro avg	0.72	0.70	0.68	18618
weighted avg	0.73	0.69	0.68	18618



Result Discussion



Winslow Homer
East Hampton Beach, Long Island, 1874

First Dataset

From the **results** we notice that we get really good accuracies of **more than 92%** when testing on the **first dataset.**

Second Dataset

When we test with the **second dataset** however we get **mixed** results of **less than 70%** accuracies.

Conclusion

- In conclusion, our project **successfully** demonstrates the ability to **distinguish** between **AI-generated** and **human-created** artworks.
- The **DenseNet** and **CNN** Models in particular show very promising results at **94.30%** and **94.08%** accuracy respectively.
- Future work could explore using a more **diverse** dataset in order to get more **generalized** models.

Jean-François Millet
Portrait of
a Man, c. 1845

Thanks for attending!

Do you have any questions?

Ask away!

Gustave Courbet La Grotte de la Loue, 1864