Detecting Art Forgeries Using Machine Learning

Seoho Hahm

Art Forgery Detection

$$$$$$$$

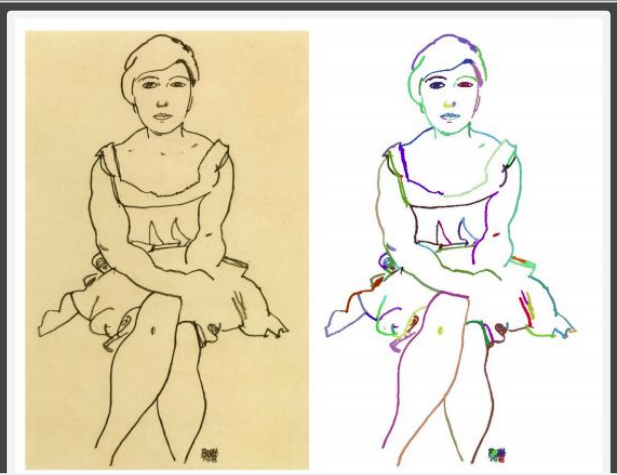
Past:

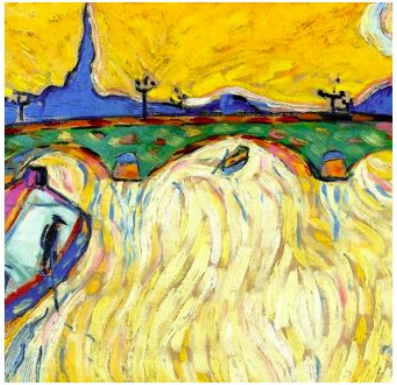
- Relied on human expertise & expensive technology
- Time consuming & unreliable

Machine Learning & Art Forgery

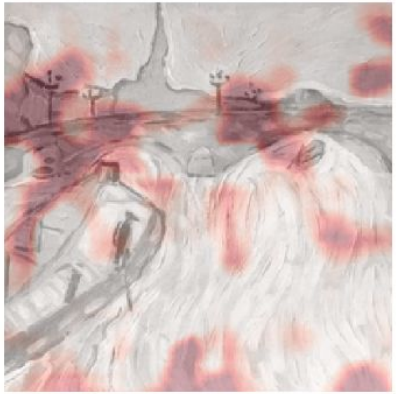
Present & Future:

- Machine learning analyzing artists' line/brush strokes
- More exact & efficient





Beltracchi fake in the style of Max Pechstein

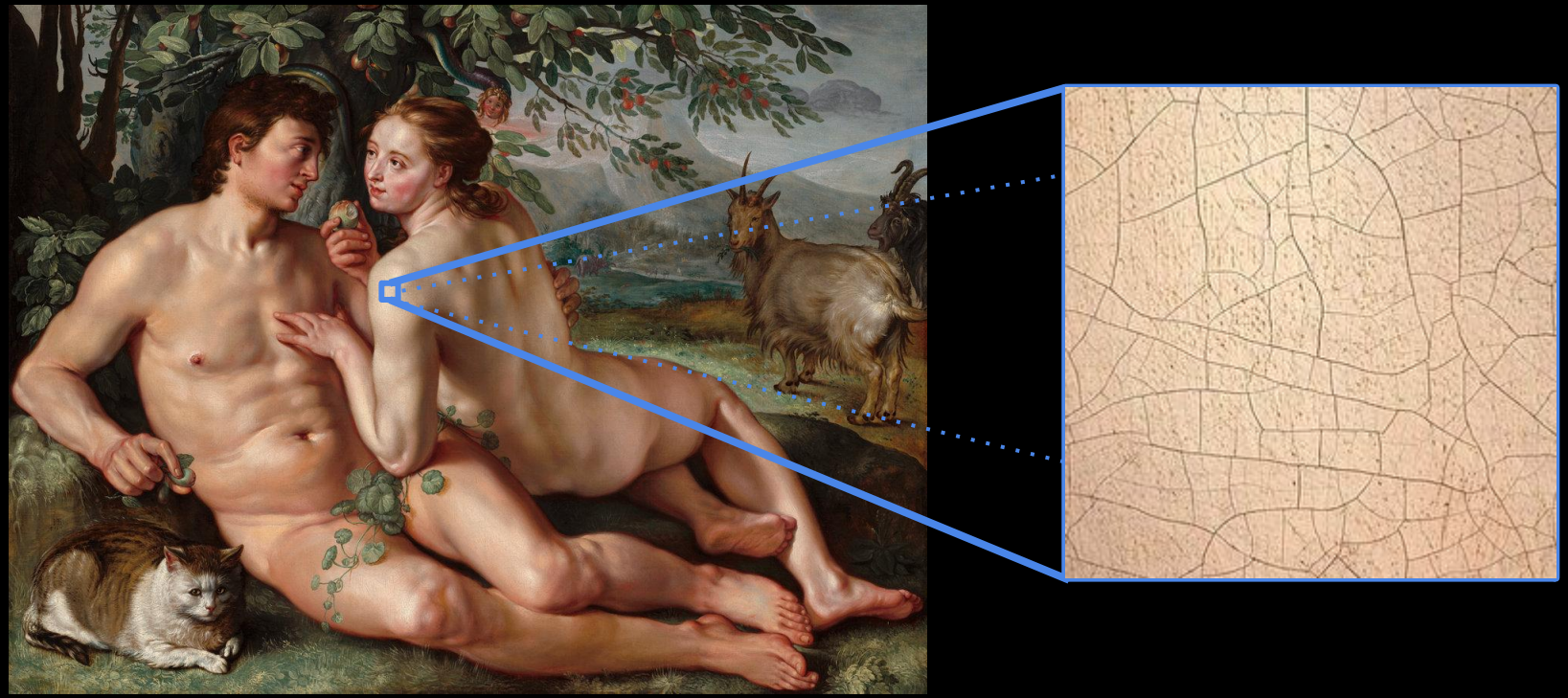


Art Recognition heat map highlighting areas of concern

This Project:
Classifying Craquelure Patterns by Region

*What is **craquelure**?*

- Cracks due to old age
- Different regions (The Netherlands, Italy, France, Belgium, etc.) have different craquelure patterns due to different mediums and materials used



Hendrick Goltzius - The Fall of Man (1616)
(Dutch)

Flemish

Dutch

The Data

- Given the time and resource constraints, this project looked specifically at:
  - 17th century
  - Oil paintings on canvas
  - Dutch vs Flemish (Belgian)
- Models based on this data could test whether a painting claiming to be a 17th c. Dutch/Flemish is indeed from then and there

Crop of craquelure (256 x 256 px)

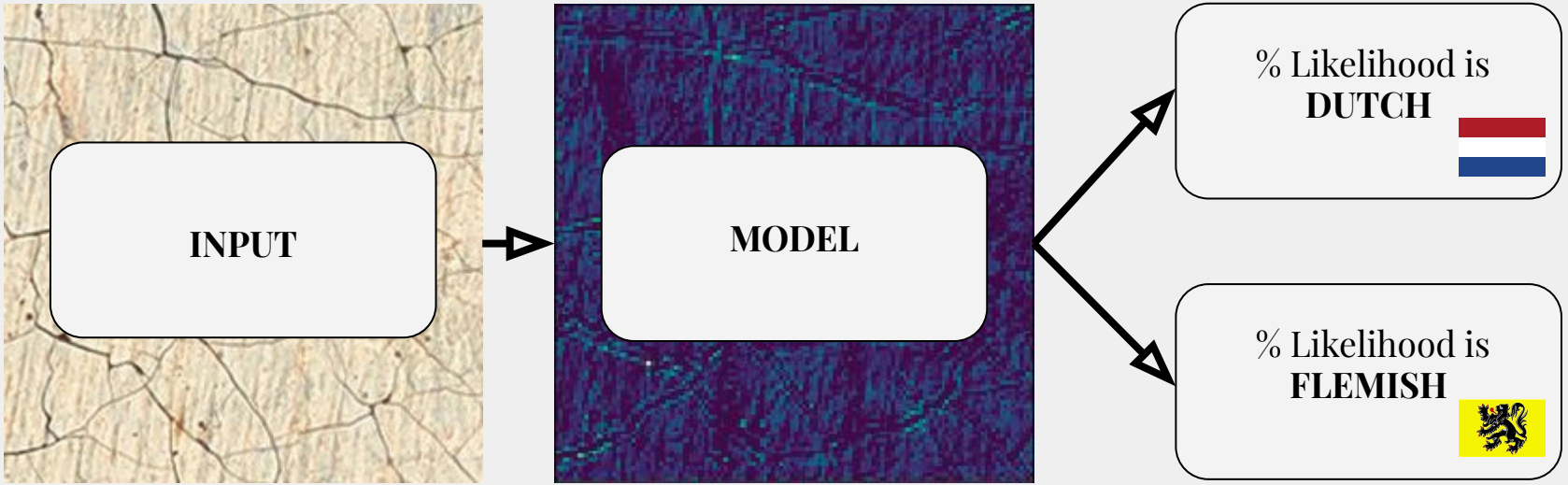
grayscale

color

black & white

Source: National Gallery of Art

Process



Model Performance

	Grayscale	Color	B&W
Training Accuracy	~ 100%	~ 97%	~ 92%
Validation Accuracy	~ 67%	~ 79%	~ 61%
Test Accuracy	~ 65%	~ 83%	TBD

Limitations & Future Directions

- Small & biased dataset
- Need automated methods for:
  - Collecting the craquelure images
  - Standardizing/scaling the images
- Expand model to classify more regions and centuries

Thank you!

Any questions?

You can find me at
Github: seoho926
seoho.h@gmail.com

APPENDIX

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 124, 124, 10)      3010
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 12, 12, 10)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 4, 4, 10)          2510
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 1, 1, 10)          0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 1, 1, 10)          110
_________________________________________________________________
average_pooling2d_1 (Average (None, 1, 1, 10)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 10)                0
_________________________________________________________________
dense_1 (Dense)              (None, 20)                220
_________________________________________________________________
dense_2 (Dense)              (None, 100)               2100
_________________________________________________________________
dense_3 (Dense)              (None, 200)               20200
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 201
=================================================================
Total params: 28,351
Trainable params: 28,351
Non-trainable params: 0
_________________________________________________________________
```


Summary of Convolution Neural Network (CNN) Color Model