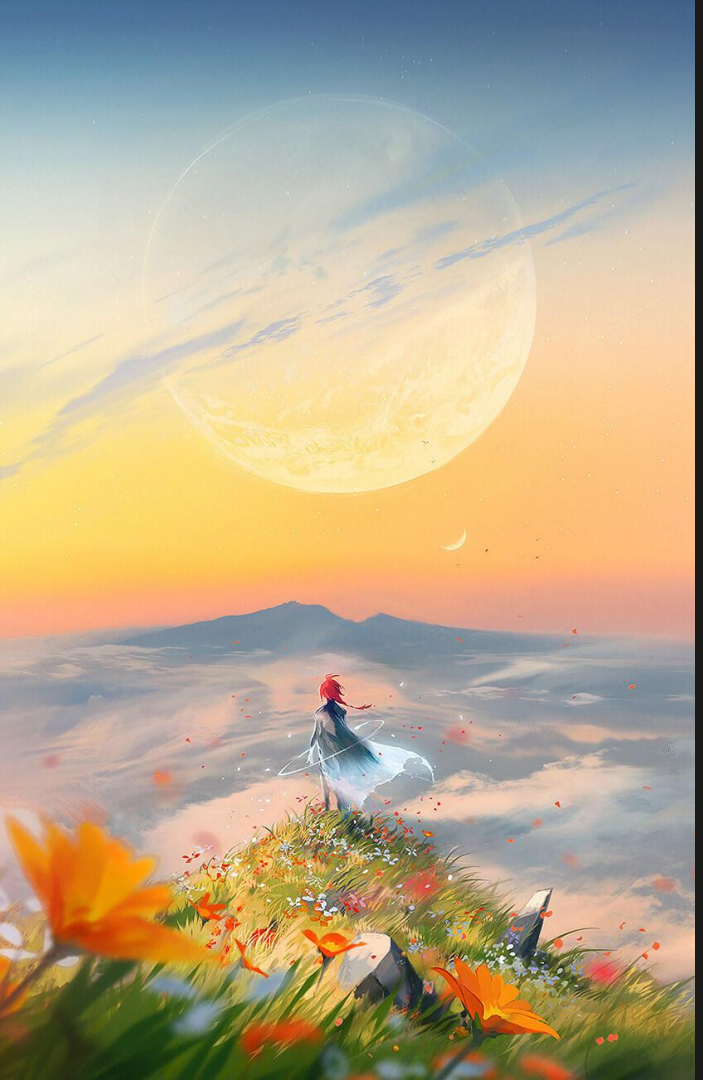


# Identifying Artist from Artwork with CNN Image Classification

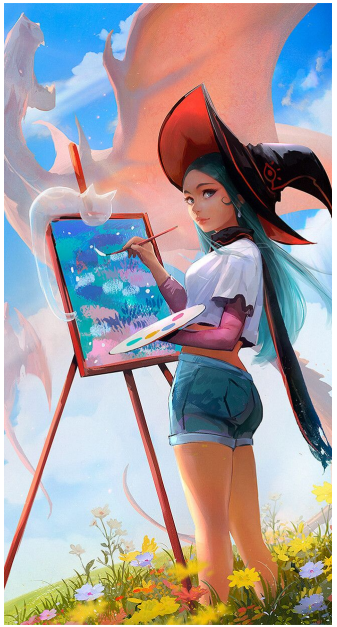
Alison Ding

# Introduction

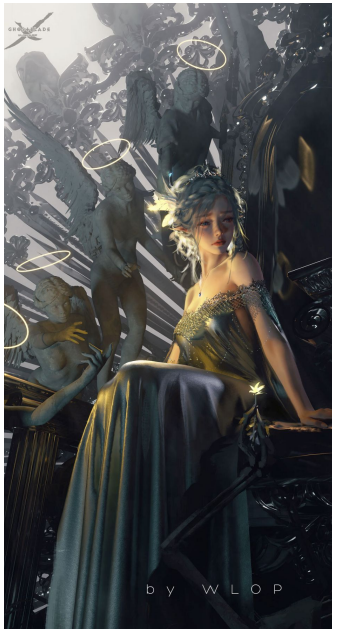
- Digital art theft - way to find original artist?


- Train ML model to identify artist of a given artwork

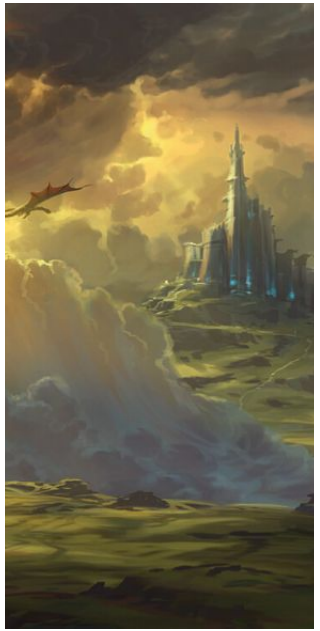
Ross Tran



WLOP




Philipp Ulrich

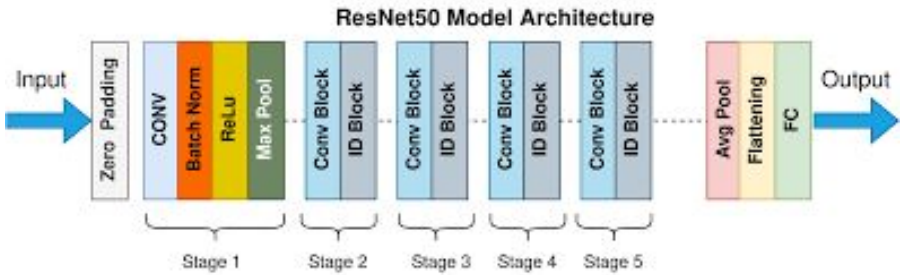


# Introduction

- ResNet50 (Convolutional Neural Network)

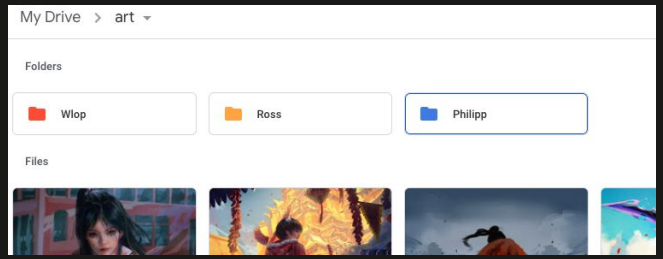
Analyze pixel info of an input image, learn to recognize and differentiate objects/aspects of it

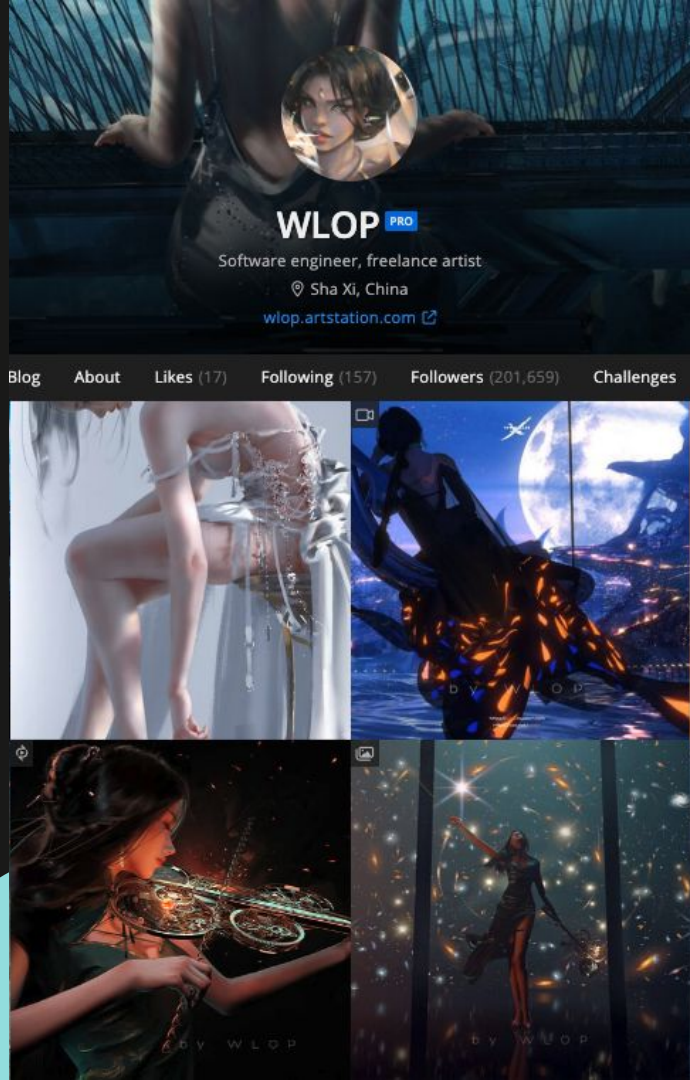
- Transfer Learning — ResNet50 pre-trained on pictures from ImageNet



# Data

- Download All Images (Google Extension)
- ArtStation - Portfolios
- Hand sorted through pictures (Colab)

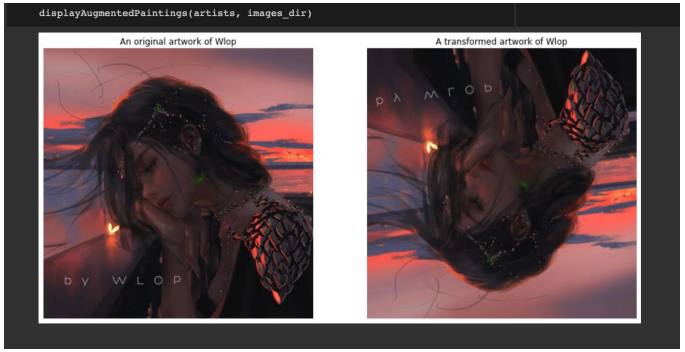


# Data

- Accessed using os module
- .csv file listing artists, number of paintings, and calculated class weights

| | name | paintings | class_weight |
|---|---|---|---|
| 0 | Philipp | 556 | 0.801788 |
| 1 | Ross | 358 | 0.938838 |
| 2 | Wlop | 234 | 1.113248 |



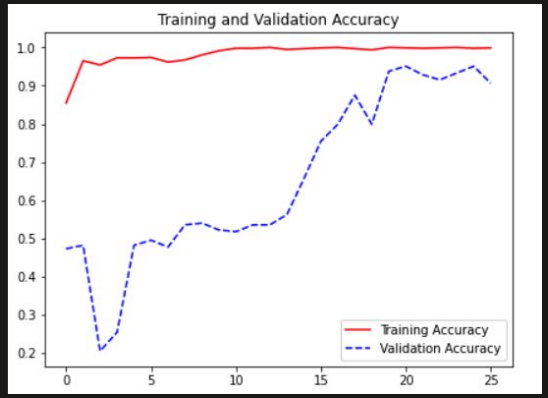

- Data generators to get training/validation data (80-20% split)
- Transformed data when training for more robust classification.

# Training the Model

Train model for 10 epochs. Reduce learning rate if improvement plateaus.

```
accuracy: 0.9913
val_accuracy: 0.5223
```

Freeze core ResNet50 layers. Train model for 25 more epochs. Stop training if no improvement for 5 epochs.

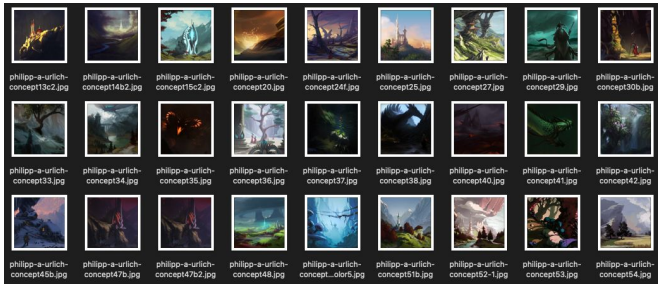



Test completed model on additional data.

```
accuracy: 0.9989
val_accuracy: 0.9062
```

# Challenges

1. Not enough data initially.



Solved: only used artists who had > 200 pieces in portfolio.

2. Model kept overfitting.

```
acy: 1.0000
5e-07.
- accuracy: 1.0000 - val_loss: 0.2547 - val_accuracy: 0.9152 - lr: 1.0000e-06

- accuracy: 1.0000 - val_loss: 0.2345 - val_accuracy: 0.9152 - lr: 1.0000e-07

- accuracy: 1.0000 - val_loss: 0.2705 - val_accuracy: 0.9062 - lr: 1.0000e-07

- accuracy: 1.0000 - val_loss: 0.2593 - val_accuracy: 0.9062 - lr: 1.0000e-07
```

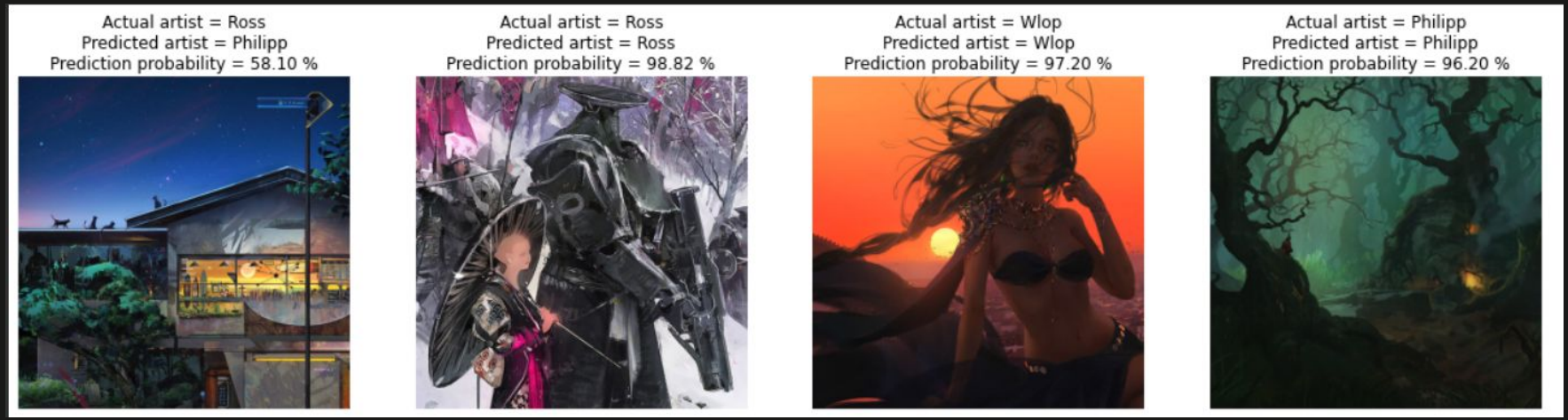
Solved: Early stopping after 5 epochs.

# Conclusions

**98%** on training data

```
<ipython-input-17-40304a0a99fe>:2: UserWarning: `Model.evaluate_generator` is deprecated
  score = myModel.model.evaluate_generator(train_generator, verbose=1)
59/59 [==============================] - 16s 266ms/step - loss: 0.0695 - accuracy: 0.9818
Prediction accuracy on train data = 0.9817596673965454

<ipython-input-17-40304a0a99fe>:6: UserWarning: `Model.evaluate_generator` is deprecated
  score = myModel.model.evaluate_generator(valid_generator, verbose=1)
15/15 [==============================] - 4s 268ms/step - loss: 0.3415 - accuracy: 0.8874
Prediction accuracy on CV data = 0.887445867061615
```

**89%** on validation data


Actual artist = Ross
Predicted artist = Philipp
Prediction probability = 58.10 %
Actual artist = Ross
Predicted artist = Ross
Prediction probability = 98.82 %
Actual artist = Wlop
Predicted artist = Wlop
Prediction probability = 97.20 %
Actual artist = Philipp
Predicted artist = Philipp
Prediction probability = 96.20 %

```
predictAny('drive/MyDrive/art/wlop2.jpg', "Wlop")
```

```
1/1 [==============================] - 0s 28ms/step
```

Actual artist = Wlop
Predicted artist = Wlop
Prediction probability = 92.54 %

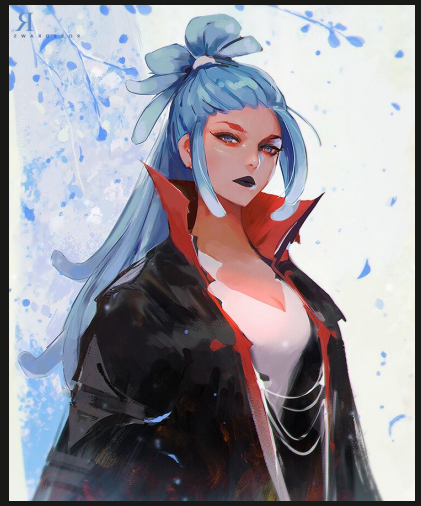
```
predictAny('drive/MyDrive/art/ross2.jpg', "Ross")
```

```
1/1 [==============================] - 0s 33ms/step
```

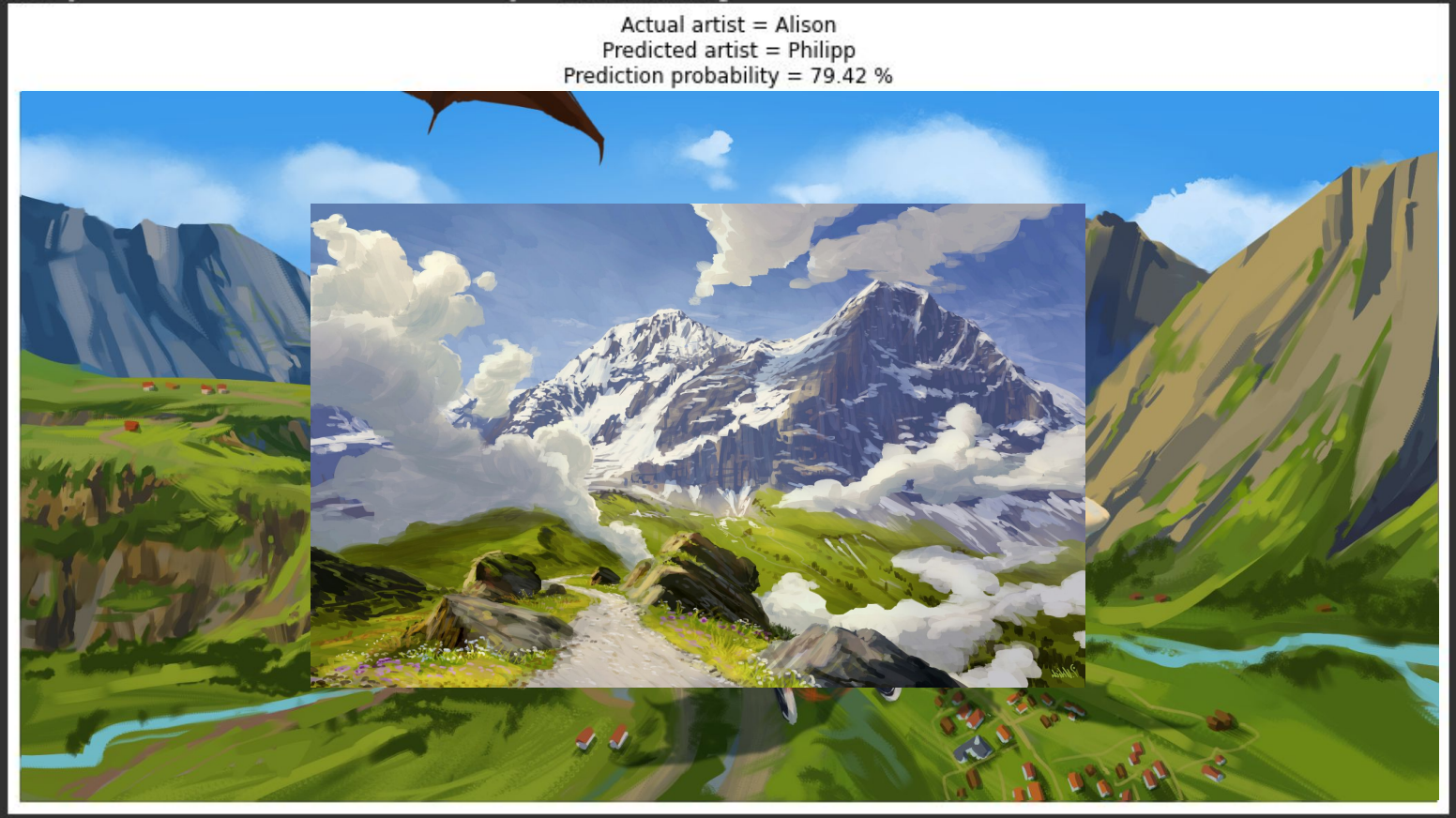
Actual artist = Ross
Predicted artist = Ross
Prediction probability = 94.14 %

```
predictAny('drive/MyDrive/art/katsura.png', "Alison")
```

```
1/1 [==============================] - 0s 29ms/step
```

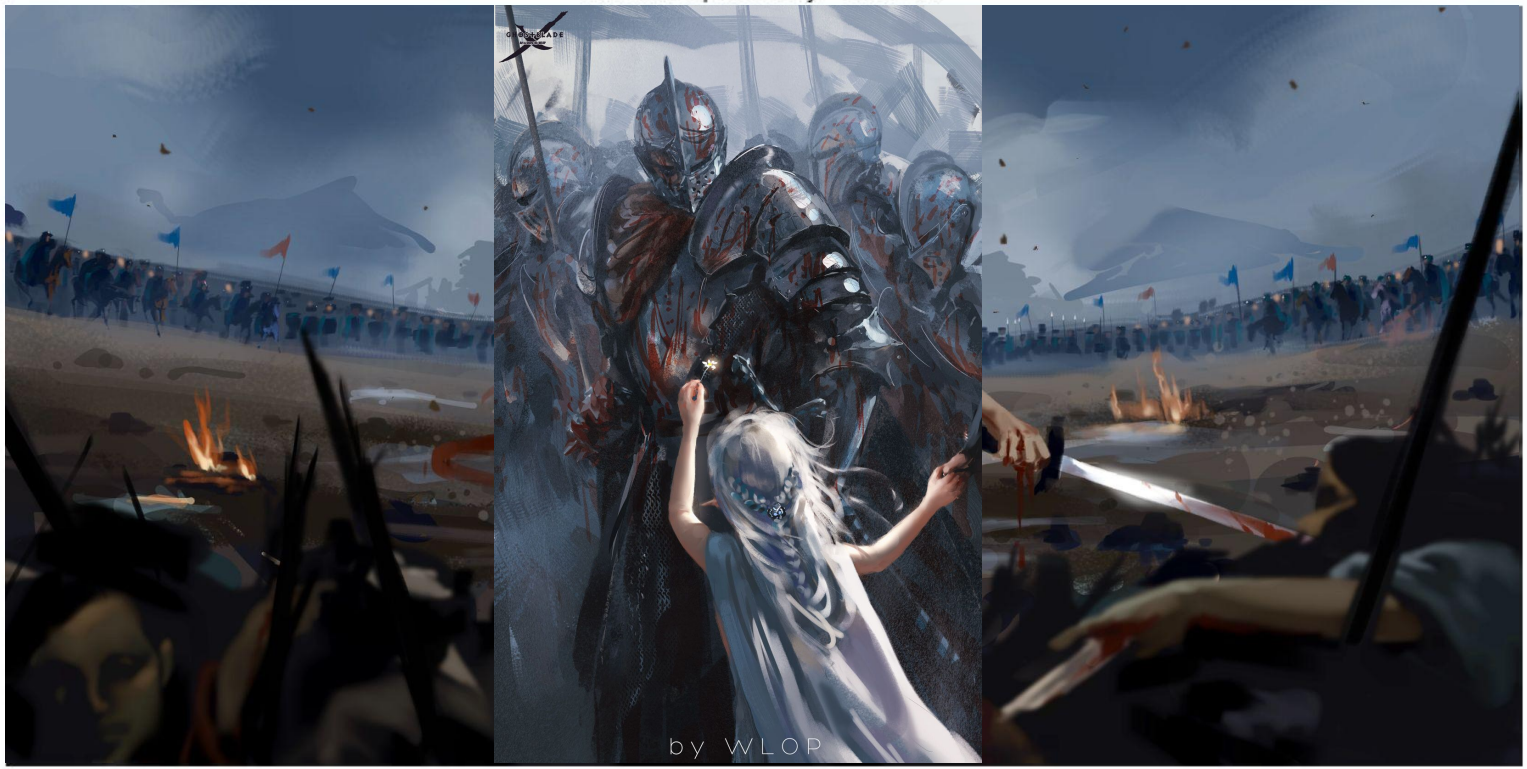
```
[51] predictAny('drive/MyDrive/art/plane.png', "Alison")

1/1 [==============================] - 0s 27ms/step
```



```
predictAny('drive/MyDrive/art/samurai.jpg', "Alison")
```

```
1/1 [==============================] - 0s 28ms/step
```

Actual artist = Alison
Predicted artist = Wlop
Prediction probability = 95.66 %



# Pros and Cons

## Pros

If expanded, could be useful to people who need to trace an artwork to its original artist.

- Art enthusiasts/students
- Companies/employers

## Cons

- Currently, only works for 3 artists.
- Doesn't help small artists who don't have enough training images.

# Takeaways

- Data quantity & quality matter a lot
- Overfitting and how to fix it
- Google Colab + GPU Runtime

# Thank you!

# Bibliography


[1] Chollet, François. Keras. 2015, https://github.com/fchollet/keras.

[2] da Costa-Luis, Casper O. “Tqdm: A Fast, Extensible Progress Meter for Python and CLI.” Journal of Open Source Software, vol. 4, no. 37, May 2019, p. 1277. DOI.org (Crossref), https://doi.org/10.21105/joss.01277.

[3] Developers, TensorFlow. TensorFlow. Zenodo, 16 Nov. 2022. Zenodo, https://doi.org/10.5281/zenodo.7604251.

[4] Harris, Charles R., et al. “Array Programming with NumPy.” Nature, vol. 585, no. 7825, Sept. 2020, pp. 357–62. DOI.org (Crossref), https://doi.org/10.1038/s41586-020-2649-2.

[5] Hunter, John D. “Matplotlib: A 2D Graphics Environment.” Computing in Science & Engineering, vol. 9, no. 3, 2007, pp. 90–95. DOI.org (Crossref), https://doi.org/10.1109/MCSE.2007.55.



[6] “Os — Miscellaneous Operating System Interfaces.” Python Documentation, https://docs.python.org/3/library/os.html. Accessed 12 Feb. 2023.



[6] “Os — Miscellaneous Operating System Interfaces.” Python Documentation, https://docs.python.org/3/library/os.html. Accessed 12 Feb. 2023.

[7] Team, The Pandas Development. Pandas-Dev/Pandas: Pandas. v1.5.3, Zenodo, 19 Jan. 2023. DOI.org (Datacite), https://doi.org/10.5281/ZENODO.7549438.

[8] WLOP. https://wlop.artstation.com/ Accessed 13 Feb. 2023.

[9] Urlich, Philipp A. https://urlich.art/ Accessed 13 Feb. 2023.

[10] Tran, Ross. https://rossdraws.artstation.com/projects Accessed 13. Feb. 2023.