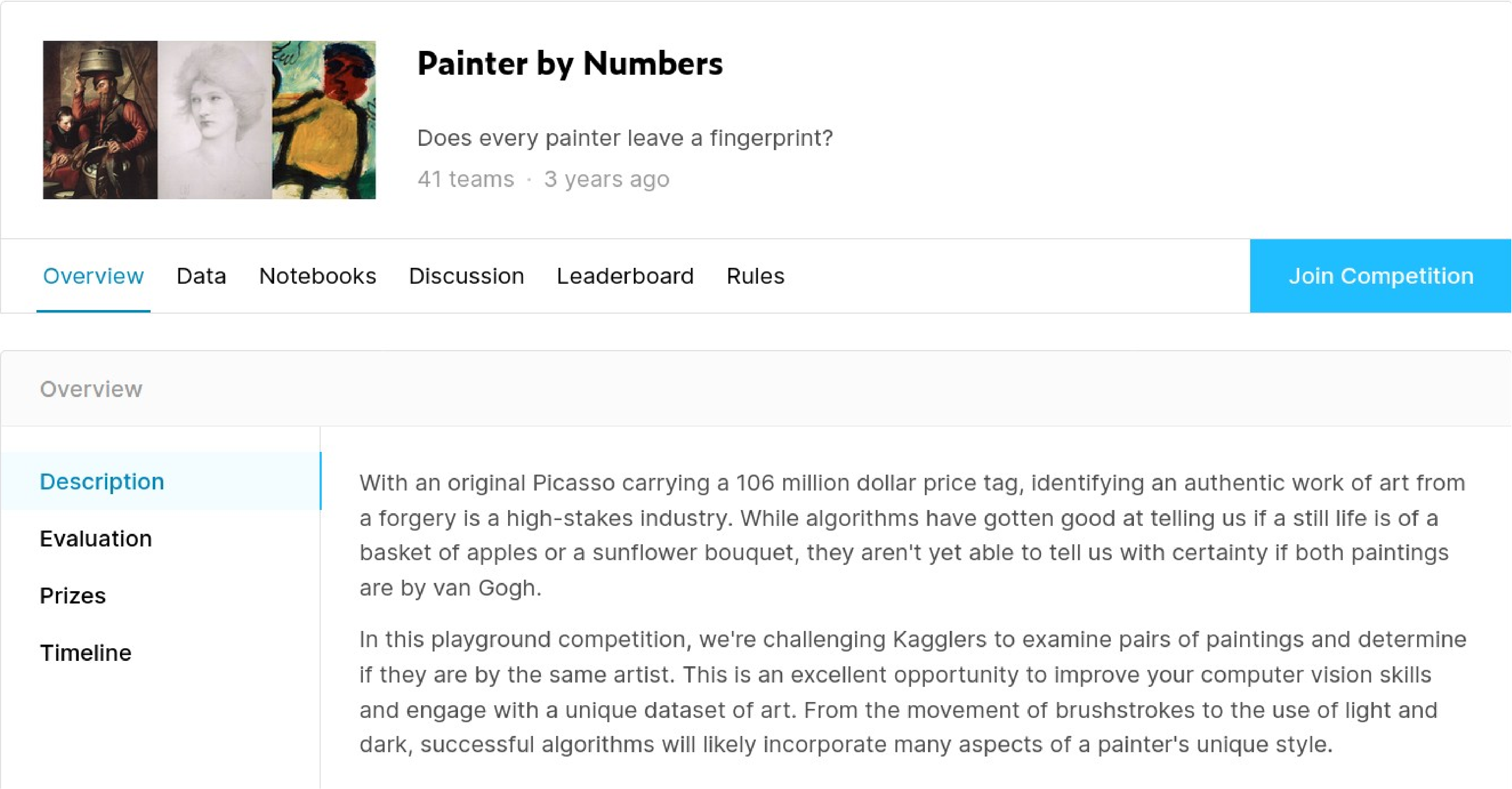
DL Case Study

Painter by Numbers Competition

https://www.kaggle.com/c/painter-by-numbers

https://www.kaggle.com/c/painter-by-numbers

https://small-yellow-duck.github.io/painter_by_numbers.html



Painter by Numbers

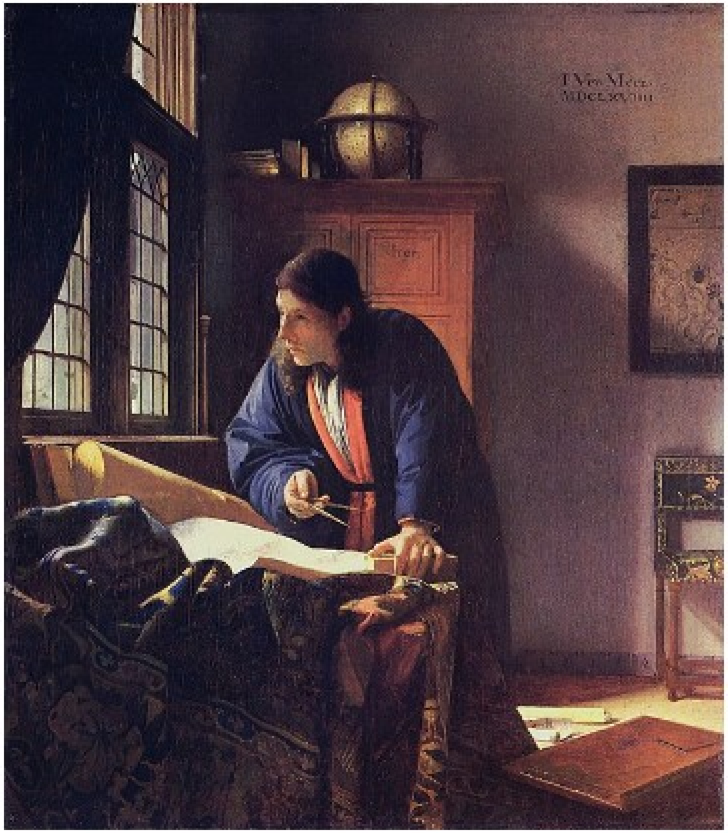
Does every painter leave a fingerprint?

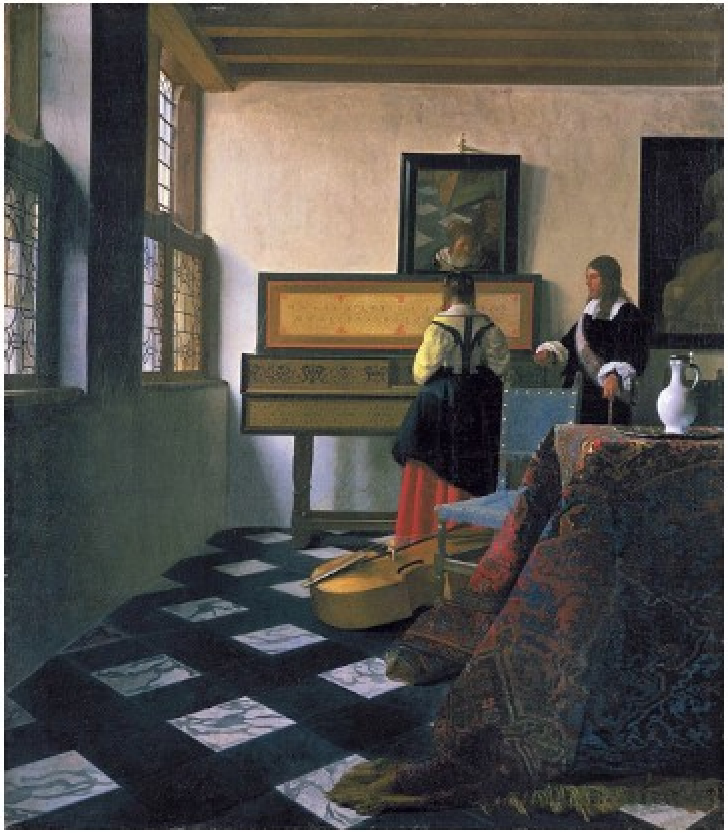
41 teams · 3 years ago

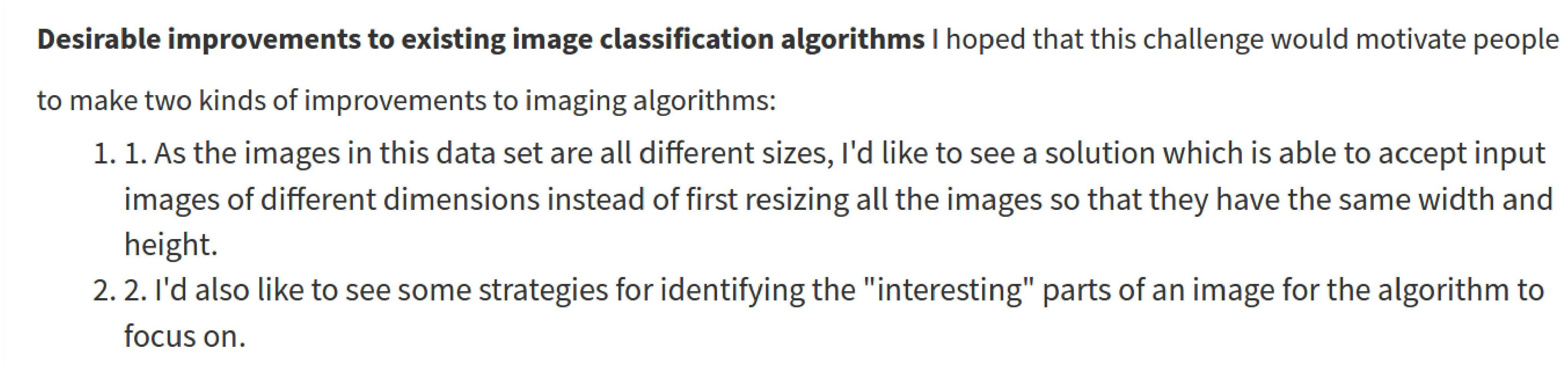
With an original Picasso carrying a 106 million dollar price tag, identifying an authentic work of art from a forgery is a high-stakes industry. While algorithms have gotten good at telling us if a still life is of a basket of apples or a sunflower bouquet, they aren't yet able to tell us with certainty if both paintings are by van Gogh.

In this playground competition, we're challenging Kagglers to examine pairs of paintings and determine if they are by the same artist. This is an excellent opportunity to improve your computer vision skills and engage with a unique dataset of art. From the movement of brushstrokes to the use of light and dark, successful algorithms will likely incorporate many aspects of a painter's unique style.

Are these two paintings by the same artist?







Desirable improvements to existing image classification algorithms I hoped that this challenge would motivate people to make two kinds of improvements to imaging algorithms:

1. 1. As the images in this data set are all different sizes, I'd like to see a solution which is able to accept input images of different dimensions instead of first resizing all the images so that they have the same width and height.
2. 2. I'd also like to see some strategies for identifying the "interesting" parts of an image for the algorithm to focus on.

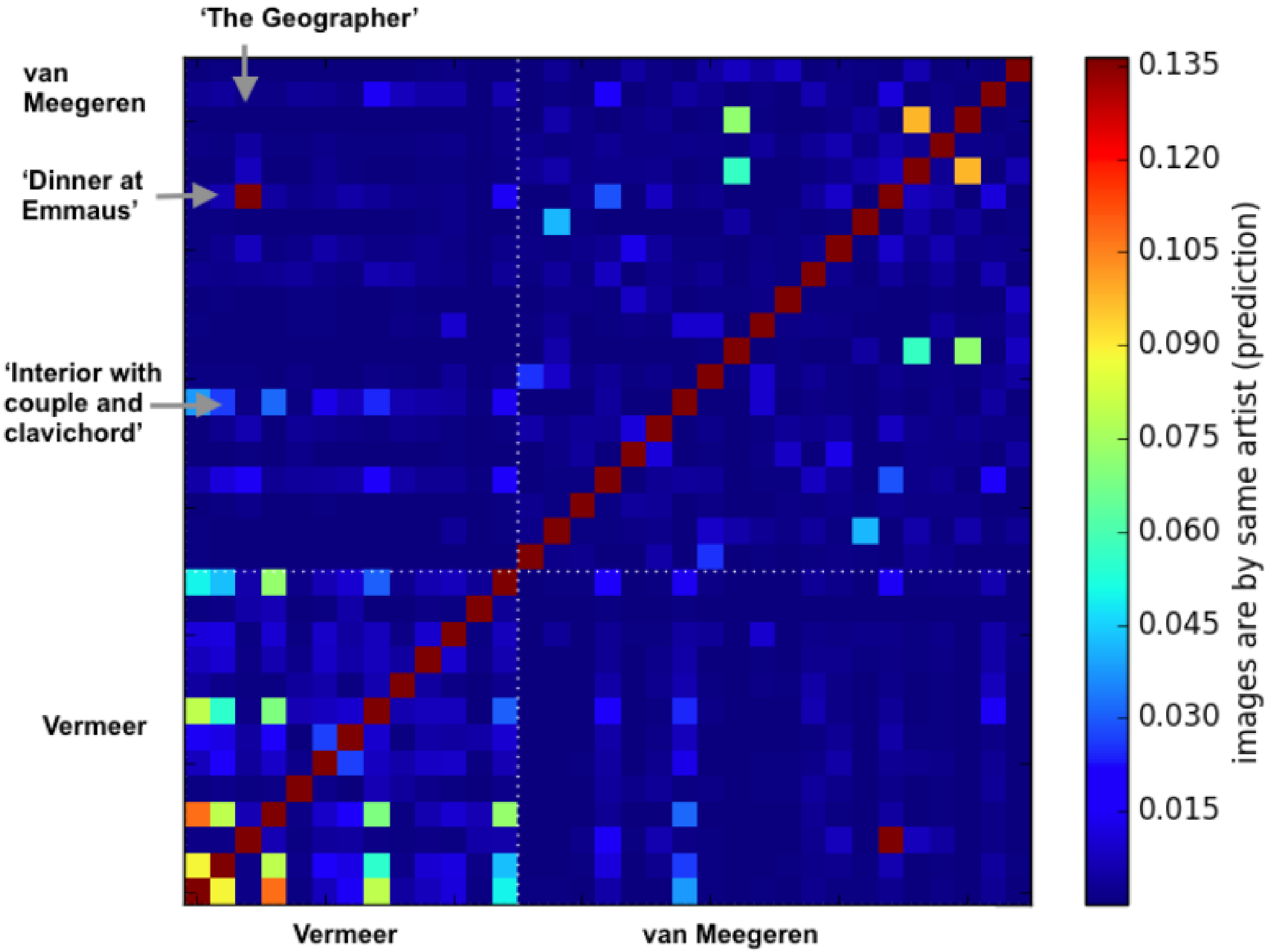
Data augmentation

Possible strategies that might be worth trying:

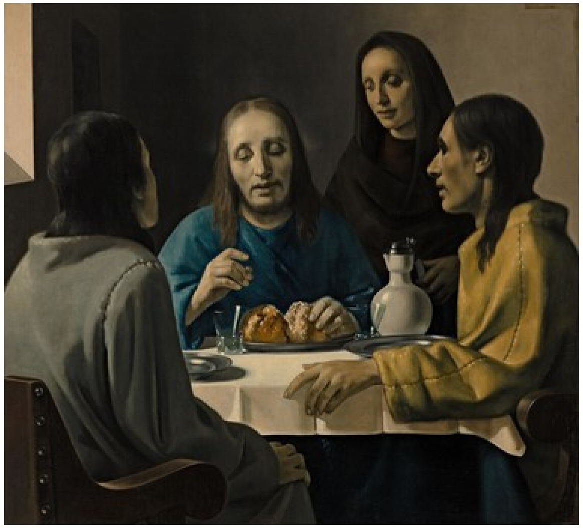
1. augment the data set by flipping the images left-to-right and labeling the mirrored images as belonging to a new artist
2. augment the data set by making small adjustments to the colour balance in the images
3. augment the data set by converting the images to greyscale
4. augment the data set by decreasing the resolution of the images
5. augment the data set by cropping the images randomly (or developing a scheme to crop the images)

Evaluating the algorithms

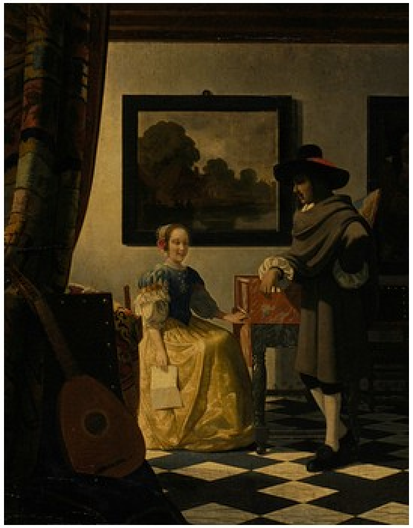
The AUC metric was used to evaluate the algorithms. **The three top-scoring algorithms all had AUCs greater than 0.9.** If I had a surplus of time and ambition, I would build an app to serve pairs of paintings up to human viewers in order to determine if the algorithm is outperforming people.



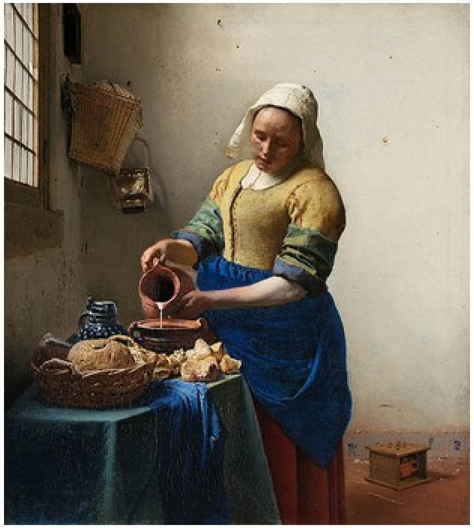
One thing I did do to compare the top algorithm to human performance was to slip a famous forger into the test set. Han van Meegeren was a Dutch artist who created several paintings in the style of Dutch masters from the 1700's. van Meegeren was bitter about the lack of commercial success he had with his own art, so he took up creating "new" paintings by old masters and he even sold some forgeries to the Nazis during World War II. I generated a pairwise comparison table for first-place winner orange-nejc's predictions for van Meegeren and Vermeer paintings in the test set (see at right).



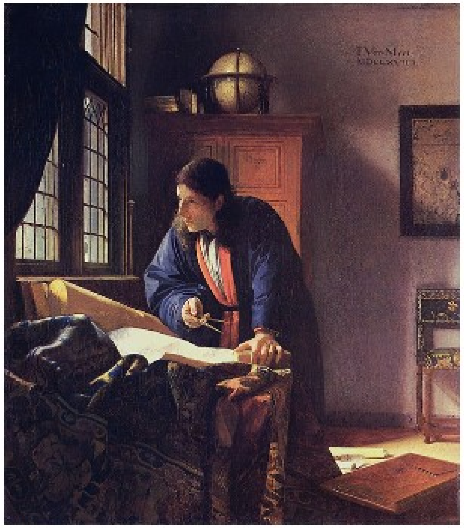
Dinner at Emmaus (van Meegeren)



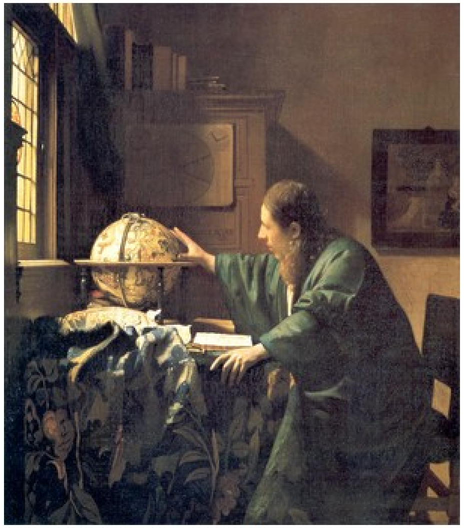
Interior with couple and clavichord (van Meegeren)



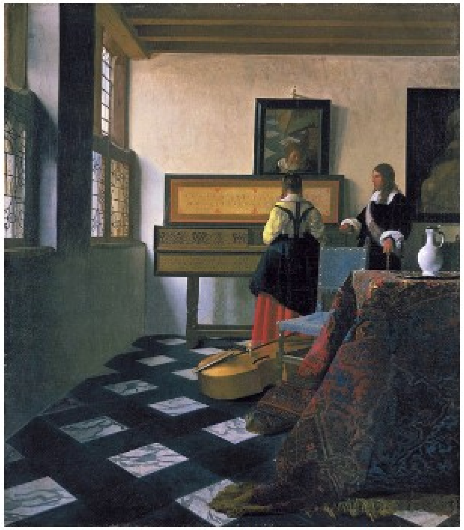
The Milkmaid (Vermeer)



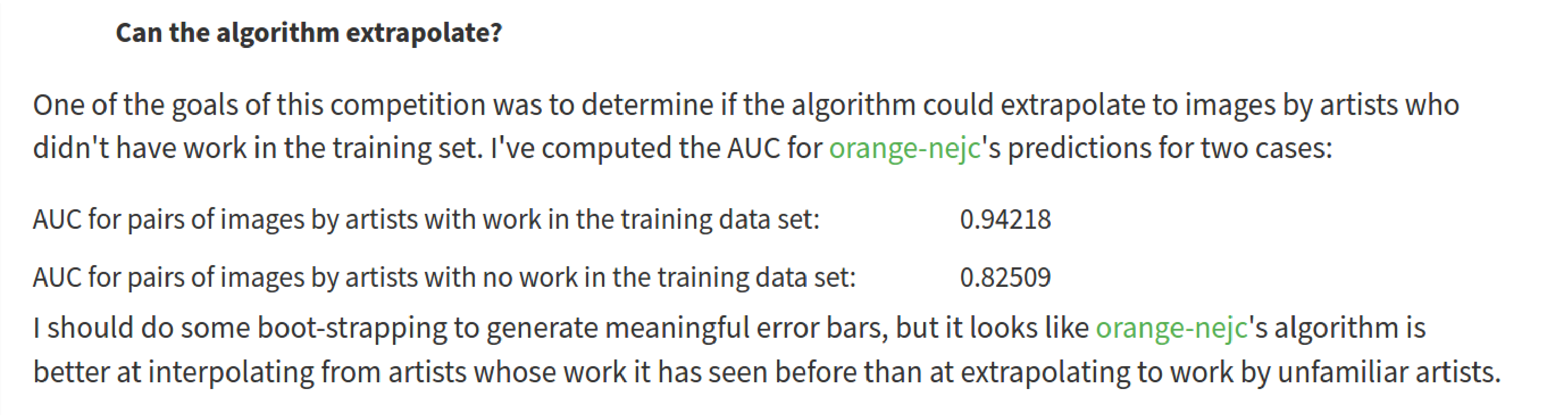
The Geographer (Vermeer)



The Astronomer (Vermeer)



Lady at the Virginals with a Gentleman (Vermeer)

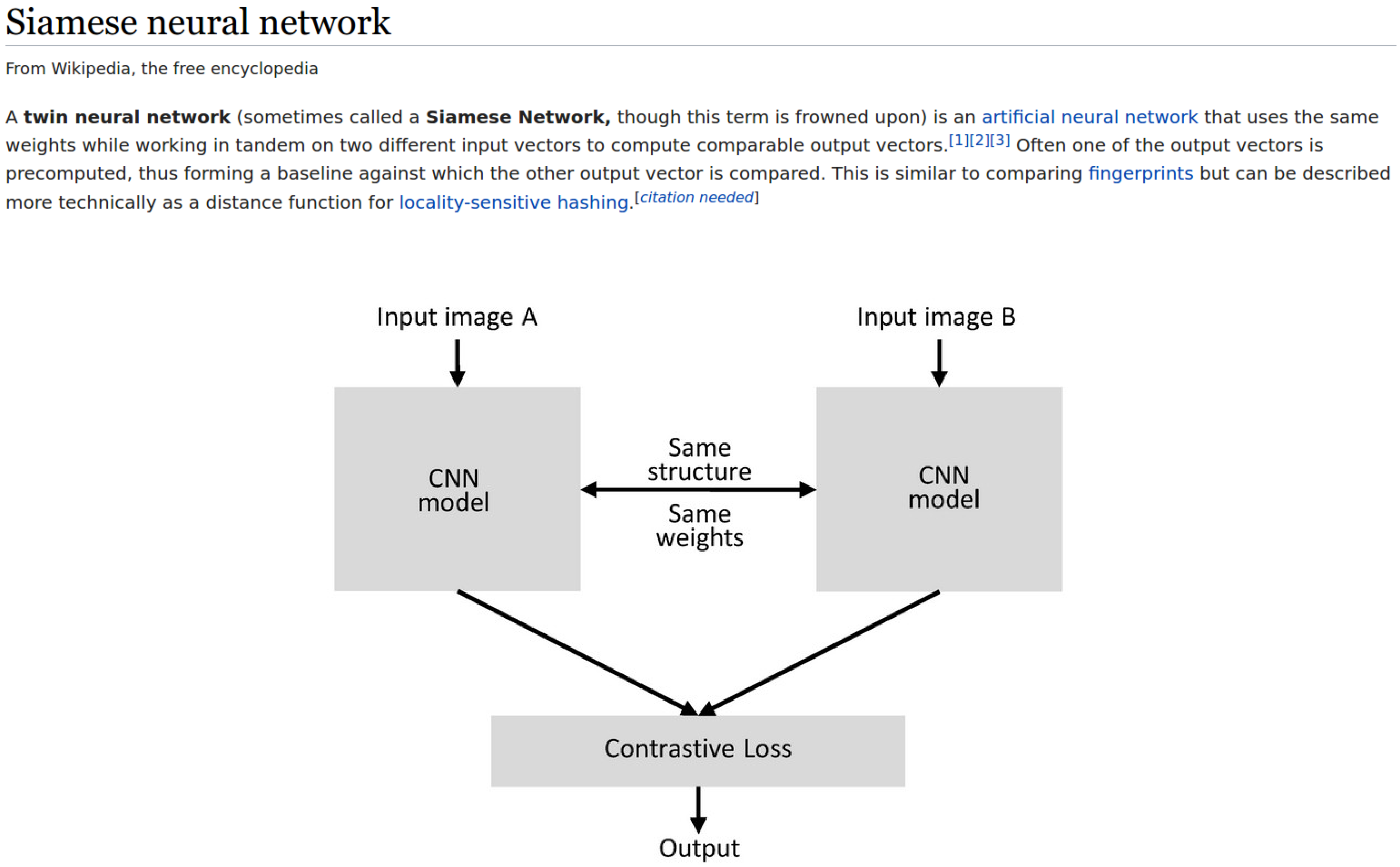


Can the algorithm extrapolate?

One of the goals of this competition was to determine if the algorithm could extrapolate to images by artists who didn't have work in the training set. I've computed the AUC for orange-nejc's predictions for two cases:

AUC for pairs of images by artists with work in the training data set:	0.94218
AUC for pairs of images by artists with no work in the training data set:	0.82509

I should do some boot-strapping to generate meaningful error bars, but it looks like orange-nejc's algorithm is better at interpolating from artists whose work it has seen before than at extrapolating to work by unfamiliar artists.



Siamese neural network

From Wikipedia, the free encyclopedia

A **twin neural network** (sometimes called a **Siamese Network,** though this term is frowned upon) is an artificial neural network that uses the same weights while working in tandem on two different input vectors to compute comparable output vectors.[1][2][3] Often one of the output vectors is precomputed, thus forming a baseline against which the other output vector is compared. This is similar to comparing fingerprints but can be described more technically as a distance function for locality-sensitive hashing.[*citation needed*]

1st Place Winner:

https://medium.com/kaggle-blog/painter-by-numbers-competition-1st-place-winners-interview-nejc-ileni%C4%8D-4eaab5e6ce9d

https://github.com/inejc/painters