Capstone 2 - Can Neural Networks Recognize Objects in Style Transfer Generated Art?

Deep Learning with Convolutional Neural Networks

Statement of Purpose

The purpose of this project is to determine how well pretrained transfer learning convolutional neural networks can recognize objects in images that have been stylized through a style transfer model.

Specifically, we'll be testing VGG16, ResNet50, and DenseNet169 models for accuracy on a test set on nonstylized images and then using the most accurate of these models on images stylized through Magenta arbitrary style transfer.

The usefulness of this project comes from the fact that image detection and image generating models are becoming increasingly popular and increasingly sophisticated. This project attempts to bring together both image recognition and image generation.

Statement of Purpose cont.

The clients for this project include anyone who is interested in developing better image recognition models, whether they are individuals or companies. While the work done here is not cutting edge, it can offer potential insight into how image recognition models “see” art.

It goes beyond the usual recognition of everyday images, to those that are more abstract and obscure but still retain traces of actual objects. In this sense, this project also becomes useful for artists as it may allow them some understanding of how image recognition models process their work.

And from a theoretical perspective, hope this project also helps spark a discussion about AI as an art interpreter.

Project Summary

- Create stylized images using Magenta arbitrary style transferring.
- Test three CNN models - VGG16, ResNet50, DenseNet169 for loss and accuracy.
- Take the most accurate of the three models and test it on stylized images.
- Tune the model further through cyclical learning and test time augmentation.
- Use our best model to determine accuracy per feature - species, style, and interpolation weight.

Flower Dataset

Our flower dataset was acquired from a a Kaggle competition that attempt to classify flower species. The link can be found on the references slide.

There are five species represented - daisy, dandelion, rose, sunflower, and tulip. There are a total of 4242 images with species sizes ranging from 1055 (dandelion) to 734 (rose).

For training, validation, and testing purposes, each species set will be randomly divided 60-20-20 into training, validation, and test sets.

Flower Dataset cont.

Below is a sample image from each species. Top to bottom, left to right: daisy, dandelion, rose, sunflower, tulip.

Art Styles

We’re using five different art styles and chosen partly for their diversity in composition as well as for their potential effects on our sample images. All are selected from Creative Commons. A brief summary on each style:

Abstract expressionism: characterized by its perceived spontaneity and arbitrary appearance. Paint drops and splatters appear to be haphazardly strewn across a canvas.

Cyberpunk: futuristic settings and characters, typically dense cityscapes and cybernetically enhanced beings; neon colors, particularly pink, blue, and purple against dark backdrops.

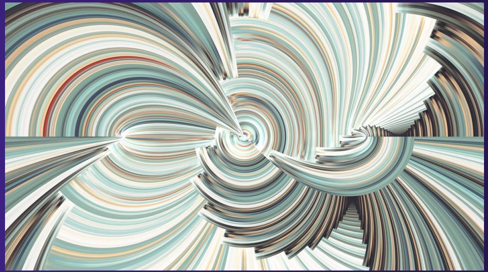
Fractal: repeating patterns, typically in the form of spirals and swirls or iterations of the Julia and Mandelbrot sets.

Pop Art: bold colors and is known to use pop culture icons as its subject matter.

Post-Impressionism: often uses unnatural, arbitrary colors, while emphasising distorted geometric forms. Van Gogh’s Starry Night is a prime example.

Art Styles

The five style images that we'll be using. Top to bottom, left ot right - abstract expressionism, cyberpunk, fractal, pop art, post-impressionism.



Magenta Neural Style Transfer Model

Magenta arbitrary style transfer model has been trained on a variety of image types and can render a stylized image regardless of the style that is inputed. Though some attempts are more successful than others.

We'll be using a series of interpolation weights. They can scale from 0.0 to 1.0 where 0.0 has no influence from the style image and 1.0 has no influence from the content image.

We took 20 random images from each species and crossed them with our five styles, all for four interpolation weights - 0.2, 0.4, 0.6, 0.8 - to give us a total of 2000 stylized images.

Stylized Images

Below is a sample of a rose image that has been stylized through the fractal image with four different weights. From top to bottom, left to right - original, 0.2, 0.4, 0.6, 0.8.





Stylized Images cont.

In many cases, as the interpolation weights increase, it becomes more difficult to identify the type of flower. In some cases, it's hard to determine whether flowers are present in the image at all. In the following images, we see the original sunflower image contrasted with a 0.8 pop art stylized image. Without the reference image, we wouldn't know that the stylized image contained flowers.

Transfer Learning CNN Model

In addition to a baseline model, we'll be testing three pretrained models, using the Imagenet dataset. - VGG16, ResNet50, and DenseNet169.

Each of our models will have several layers:

Convolutional layers to create activation maps and max pooling layers to reduce volume and computation costs.

The convolutional layers will have Rectified Linear Units or ReLU's. ReLU's slopes don't plateau, avoiding the problem of vanishing gradients.

We'll also be using the Adam optimizer. It can adjust the learning rate separately for each layer, including reducing the learning rate as the model gets closer to convergence.

Transfer Learning Model cont.

A brief description of each of our three pretrained models:

VGG16 - has 12 convolutional layers interspersed with max pooling and 4 fully connected layers for a total of 16 layers. It also has a 1000-way softmax classifier.

ResNet50 - ResNets are often regarded as improvements over VGG models. A unique feature that they have is skip connections, which allow original input to be added to the output of the convolutional block. This can help with the vanishing gradient problem that many CNN models face.

DenseNet169 - DenseNets typically require fewer parameters than traditional CNNs. Additionally, each layer in a DenseNet has access to the original image and the gradients from the loss function. They also concatenate output and input feature maps rather than summing them.

Transfer Learning CNN Model cont.

The following table lists the accuracy and loss values per model on the test data. DenseNet169 was far more accurate than any of the other models.

Test Loss and Accuracy for Four CNN Models

	Test Loss	Test Accuracy
Baseline	0.87	69.1%
VGG16	0.50	84.1%
ResNet50	1.10	69%
DenseNet169	0.59	90%

Cyclical Learning Rate

The learning rate hyperparameter is a vital part of the gradient descent process and controls how much our model is changed in response to the estimated errors. One approach to tuning this hyperparameter is cyclical learning rate (CLR), which is a part of Tensorflow's optimizer library.

We could have used one of several approaches for increasing our accuracy, but what makes CLR appealing is that rather than sticking with a single learning rate for an entire training/test run, CLR allows for a change of the learning rate during the run.

This 'resetting' of the learning rate may allow the model to find other local minima if the loss rate appears to have stalled for a number of epochs. As a result, the model may find a more optimal minimum than it would have settled on otherwise.

Test Time Augmentation

As a final method to increase our accuracy, we'll be applying test time augmentation (TTA) to our CLR model. We had already augmented our data by using ImageDataGenerator, but we had made more significant augmentations to the training data than any of the other datasets.

TTA allows for creating multiple augmented copies of every image in the test set, creating, in a sense, more data for the model to analyze and, potentially, reducing generalization error.

For TTA, we'll be applying a modified version of the code written by Jason Brownlee. More information on the specifics of the code can be found on the references slide.

DenseNet169 Tuning Results

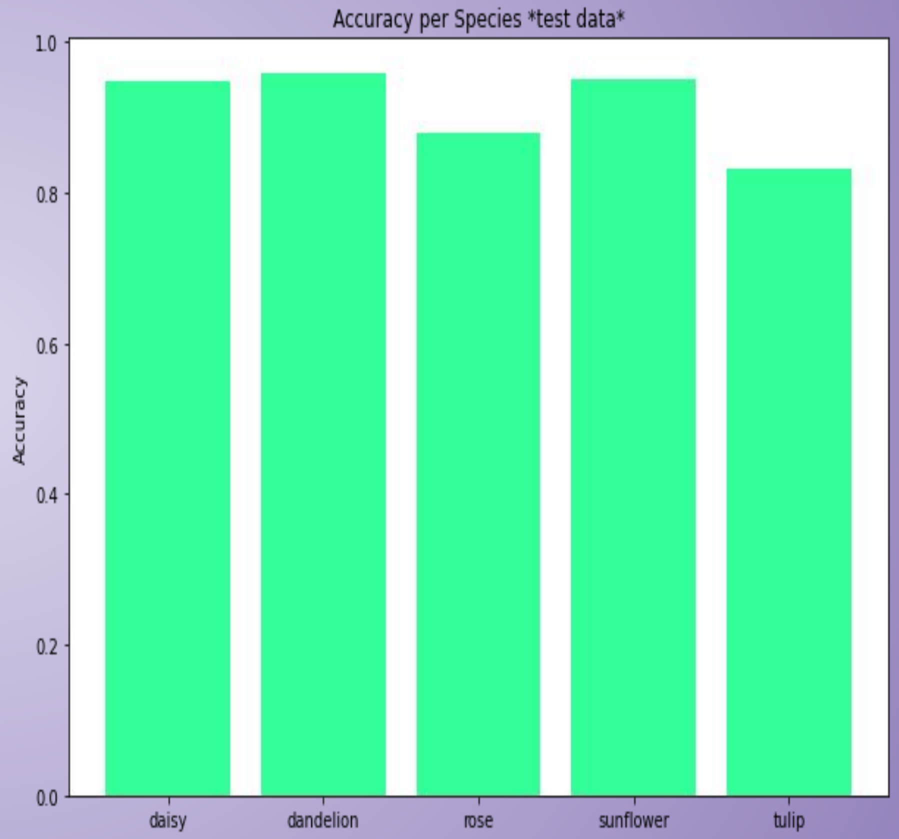
After tuning our best performing model, we received the following results on both our test and stylized datasets:

Model Version	Test Accuracy	Stylized Accuracy
DenseNet 169	90%	61%
DenseNet169 + CLR	91.1%	69.6%
DenseNet169 + CLR + TTA	95.6%	75.9%

Accuracy per Species, Style, and Weight

Test Species	Accuracy
daisy	0.948
dandelion	0.957
rose	0.88
sunflower	0.95
tulip	0.832

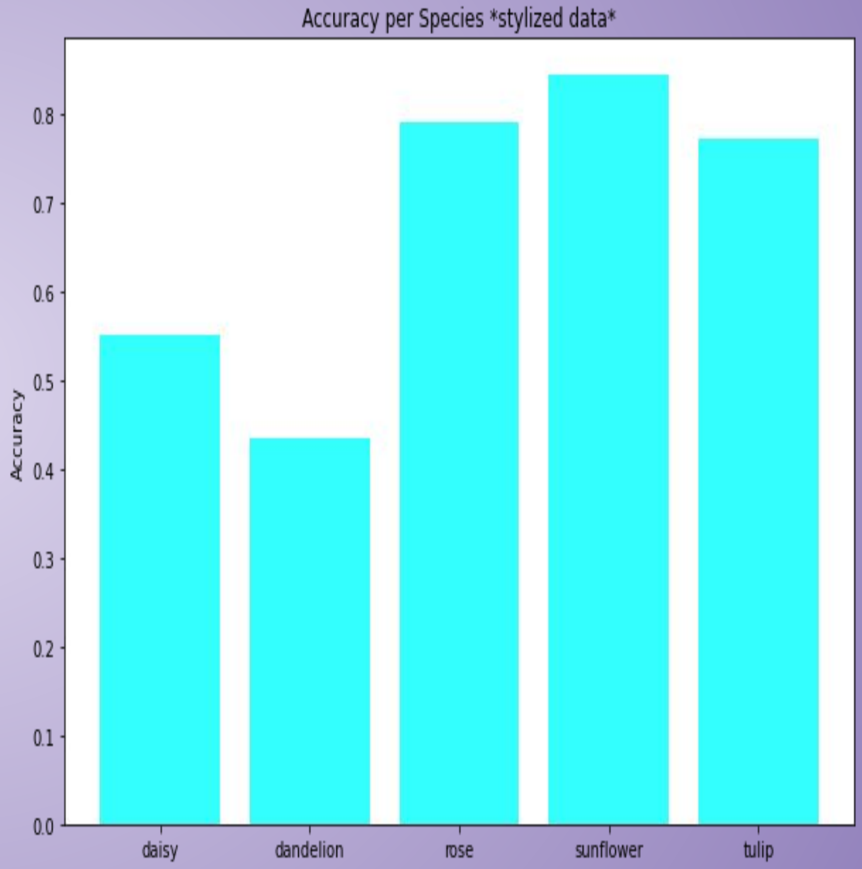
Roses and tulips performed the worse and were mistaken for each quite often. Almost half of the rose incorrect guesses were tulips, and more than half of the tulip mislabels were roses.



Accuracy per Species, Style, and Weight cont.

Style Species	Accuracy
daisy	0.55
dandelion	0.434
rose	0.79
sunflower	0.843
tulip	0.772

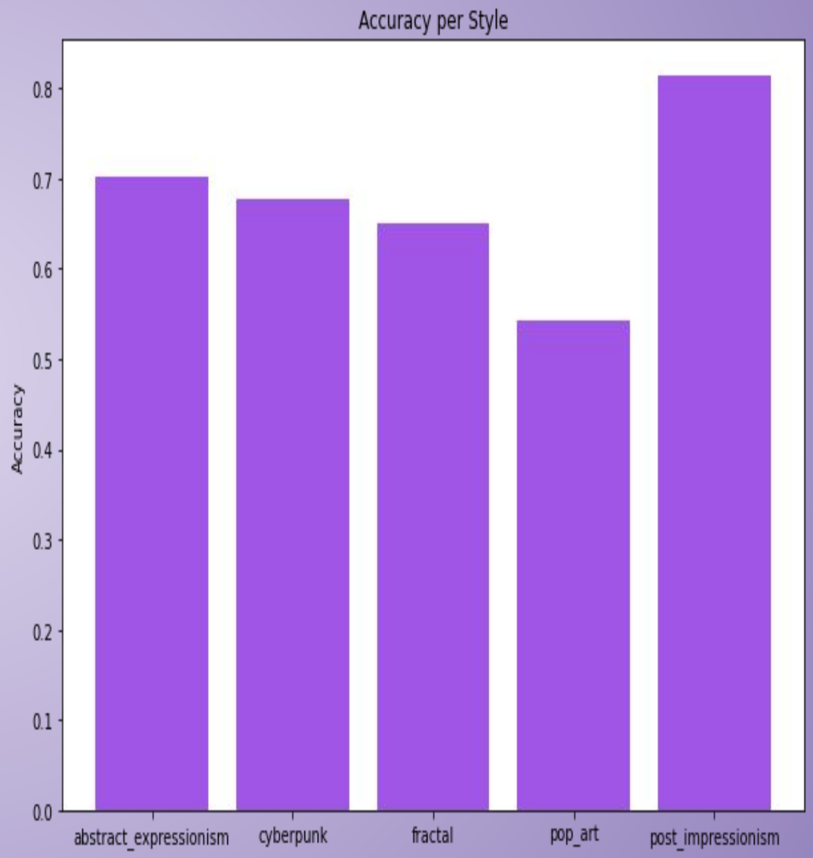
In contrast to the test images, dandelions perform the worse. It’s likely due to dandelions’ more delicate features being washed out or colored by the heavier weights and bolder styles.



Accuracy per Species, Style, and Weight cont.

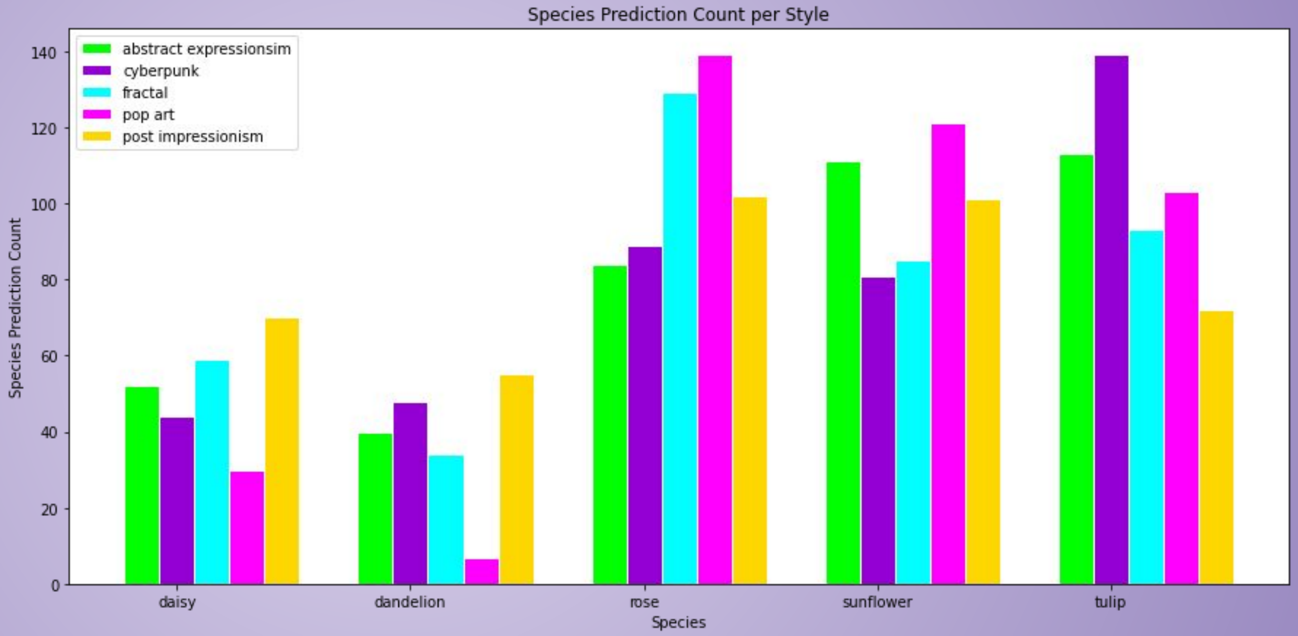
Style	Accuracy
Abstract expressionism	0.703
Cyberpunk	0.678
Fractal	0.65
Pop art	0.543
Post-impressionism	0.813

Our suspicions about the bolder styles are confirmed as the boldest style, pop art performs the worst. Its striking colors and blurring of nuances confused the model many times.



Accuracy per Species, Style, and Weight cont.

The model predicted roses, sunflowers, and tulips far more often than daisies and dandelions, regardless of the art style. The more colorful styles like abstract expressionism and pop art introduced new colors into images and tricked the model into thinking the most colorful flowers (roses, tulips and sunflowers) were present in more images than they actually were.

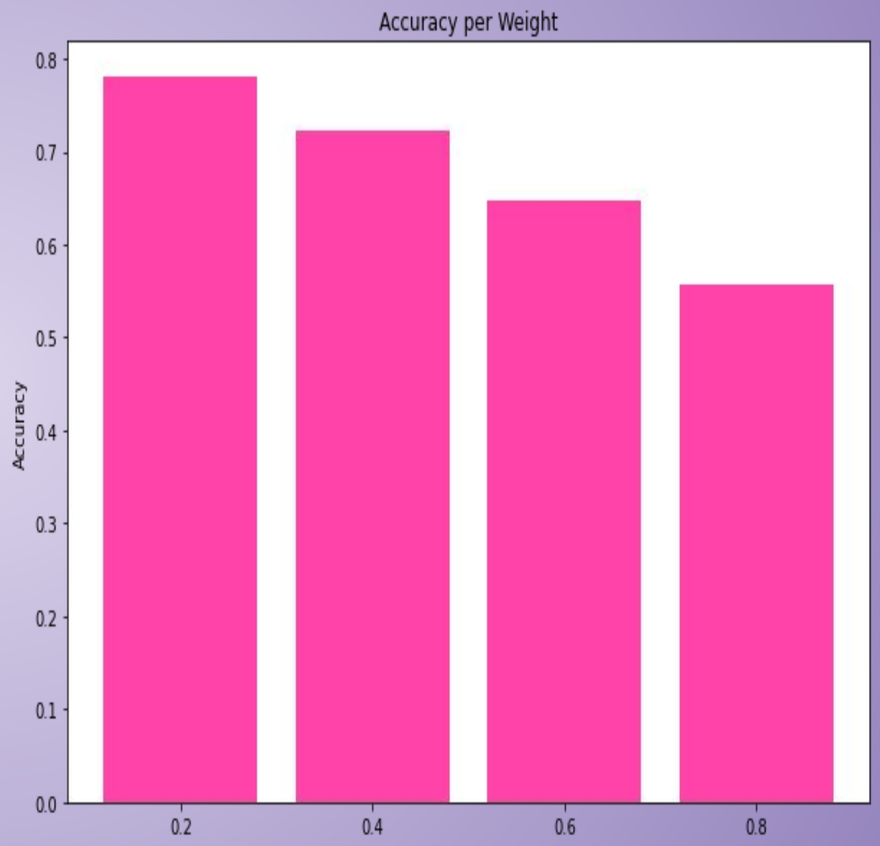


Accuracy per Species, Style, and Weight cont.

Interpolation Weight	Accuracy
0.2	0.78
0.4	0.722
0.6	0.648
0.8	0.558

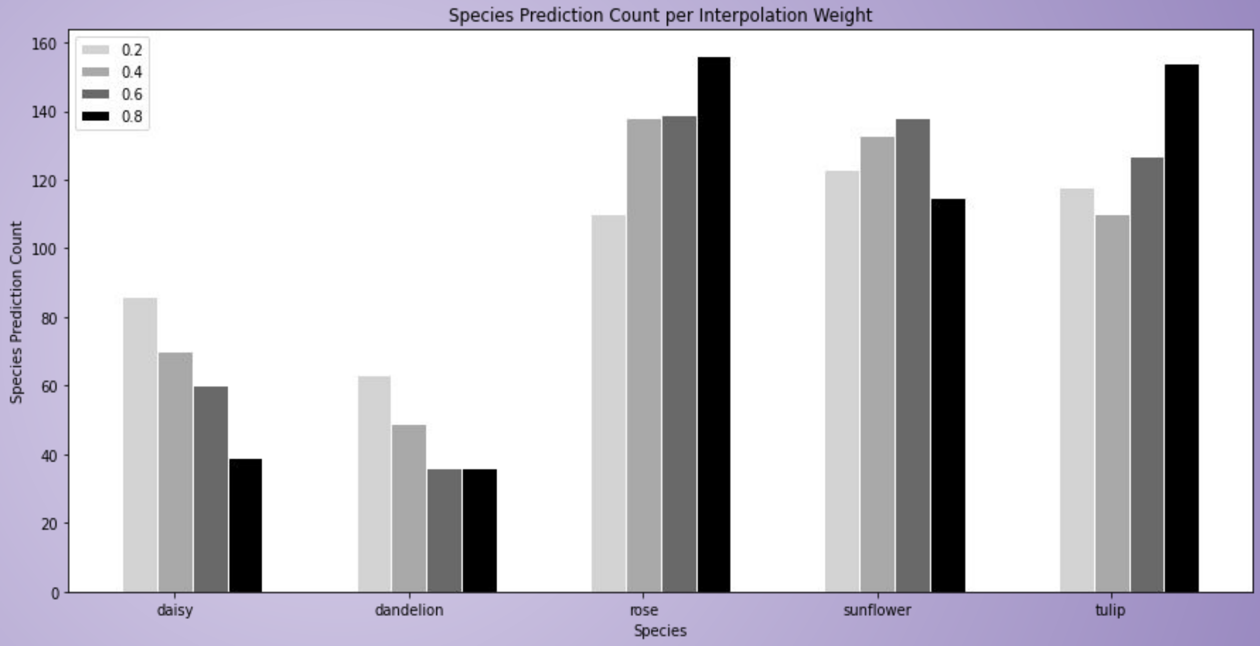
As expected the accuracy got worse as the weight increased and the image deviated further from the original.

Roses, sunflowers. And tulips still make up most of the predictions, but only compose 70% of the 0.2 predictions, but 85% of the 0.8 guesses.



Accuracy per Species, Style, and Weight cont.

At heavier weights, we see more guesses for roses, sunflowers, and tulips. The heavier weights introduce more colors, but it also leads to a blurring of features, which favors more solid flowers like roses and tulips and hurts dandelions, which have a lot of empty space.



Takeaways

It's not a surprise, as this project demonstrated, that as an image veers further from its original image, and further from the types of images on which a model has been trained, that a model's accuracy will decrease.

What was more of a surprise was that certain art styles were able to produce stylized images, by introducing new colors or blurring nuances, that tricked the AI into thinking flowers were present in an image that actually weren't.

That speaks to a larger issue of how style transfer models can be used to make new art pieces by using specific combinations of images, styles, and weights.

Takeaways cont.

This project gives us some sense of the challenges present in recognizing real world objects under less than ideal conditions. For example, we can draw analogies to an autonomous driving system trying to identify traffic lights, stop signs, cars, pedestrians, etc in rain, fog, or snow. Think of the weather conditions as filters or distortions of an ideal image of a traffic light or pedestrian. We then get a better idea of how difficult image recognition can be for AI when conditions deviate from the ideal.

We had to use several techniques like image augmentation, CLR, and TTA just to get our accuracy to a respectable 76%. And while the tools in those disciplines are much more sophisticated,the challenges are similar - it takes a lot of work to get AI to see the world as well as we do, and even more work to see it better than we do.

Takeaways cont.

Touching briefly on AI and art interpretation, not much has been written about AI as an art interpreter or critic, partly because such a thing is even hard to imagine. Unlike the analytical, logical world of artificial intelligence, art interpretation seems so subjective and murky - something much more suited for humans than a machine.

Perhaps someday we'll develop sophisticated artificial general intelligence and ask if AI can actually interpret art and how such a process works or what it even means if it's just following an algorithm.

This project likely doesn't get us much closer to answering such questions. The model's very limited in its scope and abilities, but maybe it can still persuade us to take a second look at an artistic rendering of a dandelion that AI's labeled a rose and think, "Hmm, you know, maybe that dandelion does kind of look like a rose."

Future Work

Some suggestions for developing this project further or taking it in another direction:

- Test other pretrained models to determine if they outperform DenseNet169
- Build a robust model from scratch that isn't pretrained
- Expand the number of species and/or art styles
- Use a mix of initial images and stylized images to create new works of art through a GAN model.

References

Capstone 2 Code:
https://github.com/kjd999/Springboard-files/blob/master/Capstone%20Project%202/Transfer_Learning_Final_Model.ipynb

Capstone 2 Final Report:
https://github.com/kjd999/Springboard-files/blob/master/Capstone%20Project%202/Capstone%202%20Project%20Final%20Report.pdf

Flower Dataset: https://www.kaggle.com/alxmamaev/flowers-recognition

Magenta Arbitrary Style Transfer:
https://github.com/kjd999/Springboard-files/blob/master/Capstone%20Project%202/Magenta_Arbitrary_Style_Transfer.ipynb

Hey-simone's flower classification model analysis:
https://github.com/hey-simone/flowers-classifier/blob/master/Keras_Flowers_Classifier-V1-2.ipynb

Test Time Augmentation (Jason Brownlee):
https://machinelearningmastery.com/how-to-use-test-time-augmentation-to-improve-model-performance-for-image-classification/