

Paint by GAN

DSCI6005: Machine Learning and Data Analysis 2

—
Vanessa Grass

Motivation for algorithmic art & design

Extends human creativity in ways that would otherwise never be possible!

- Nutella used an “algorithm” to create seven million different versions of its packaging
- “Come Swim” – short film using Neural Style Transfer (directed by Kristen Stewart, who even co-authored an [arXiv paper](#) on her team's technique)



Painter by Numbers (via kaggle)

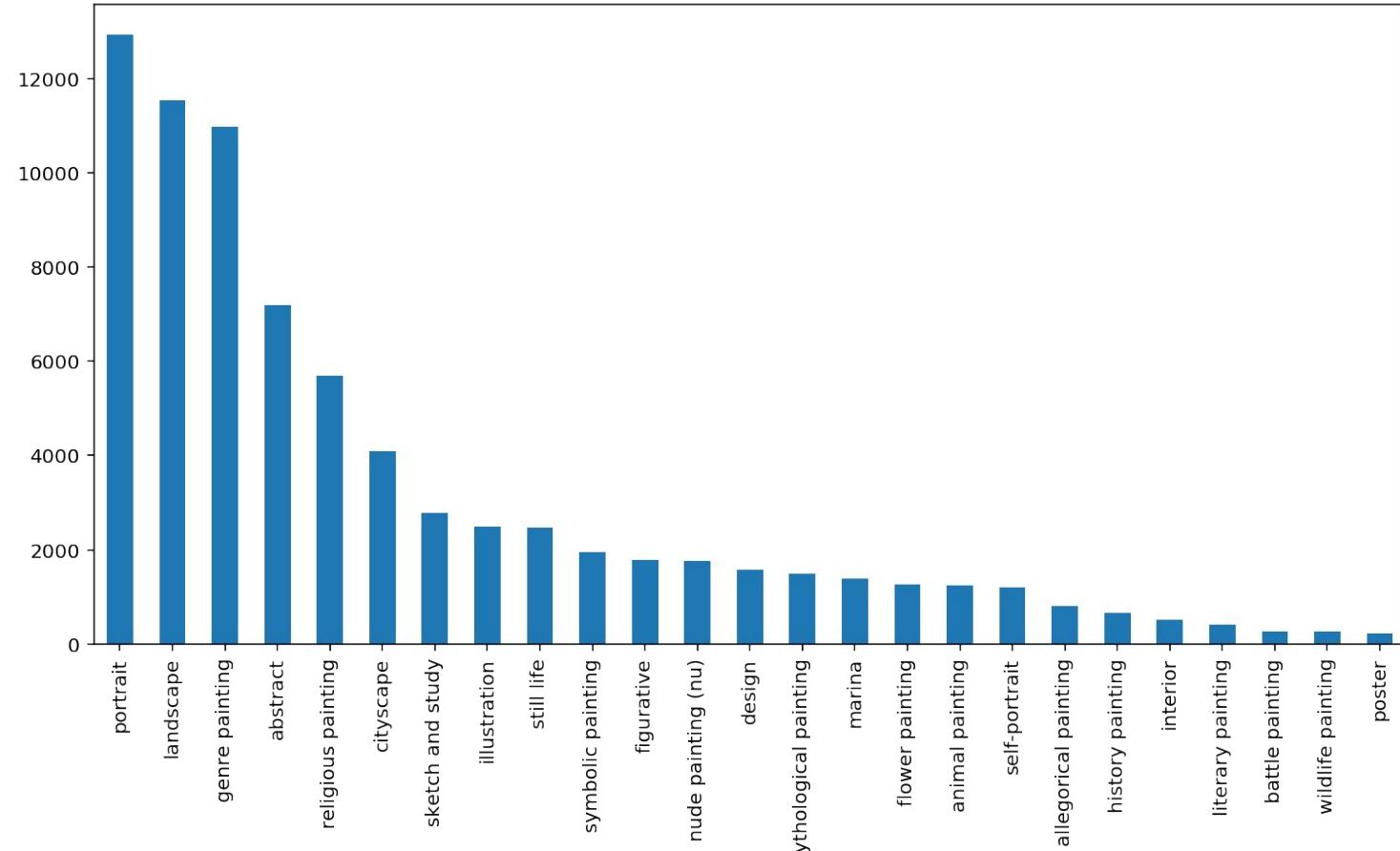
	artist	date	genre	pixelsx	pixelsy	size_bytes	source	style	title	artist_group	in_train	new_filename
0	Barnett Newman	1955.0	abstract	15530.0	6911.0	9201912.0	wikiart	Color Field Painting	Uriel	train_only	True	102257.jpg
1	Barnett Newman	1950.0	abstract	14559.0	6866.0	8867532.0	wikiart	Color Field Painting	Vir Heroicus Sublimis	train_only	True	75232.jpg
2	kiri nichol	2013.0	NaN	9003.0	9004.0	1756681.0	NaN	Neoplasticism	NaN	test_only	False	32145.jpg
3	kiri nichol	2013.0	NaN	9003.0	9004.0	1942046.0	NaN	Neoplasticism	NaN	test_only	False	20304.jpg
4	kiri nichol	2013.0	NaN	9003.0	9004.0	1526212.0	NaN	Neoplasticism	NaN	test_only	False	836.jpg

103, 250 total images:

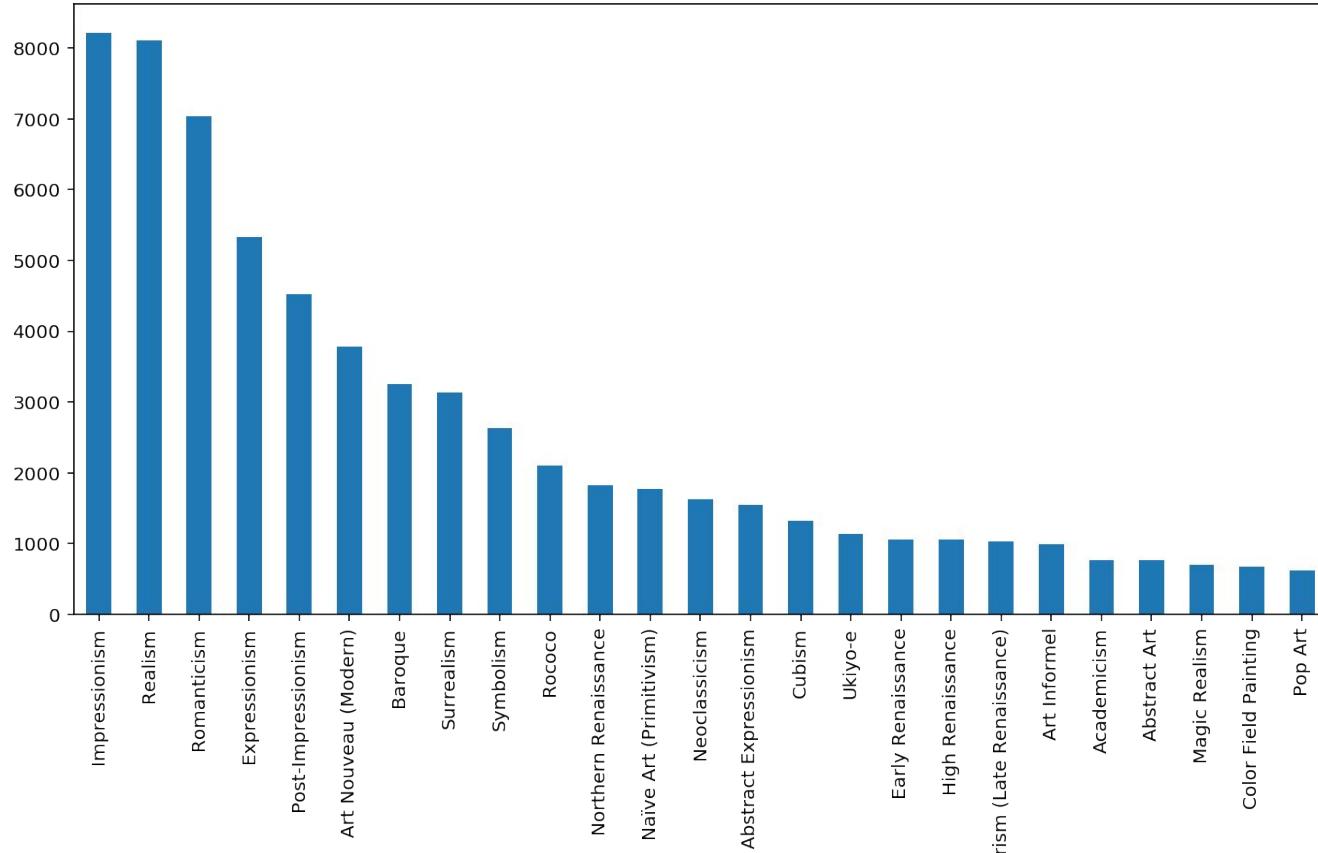
79,433 in training set | 23,817 in testing set



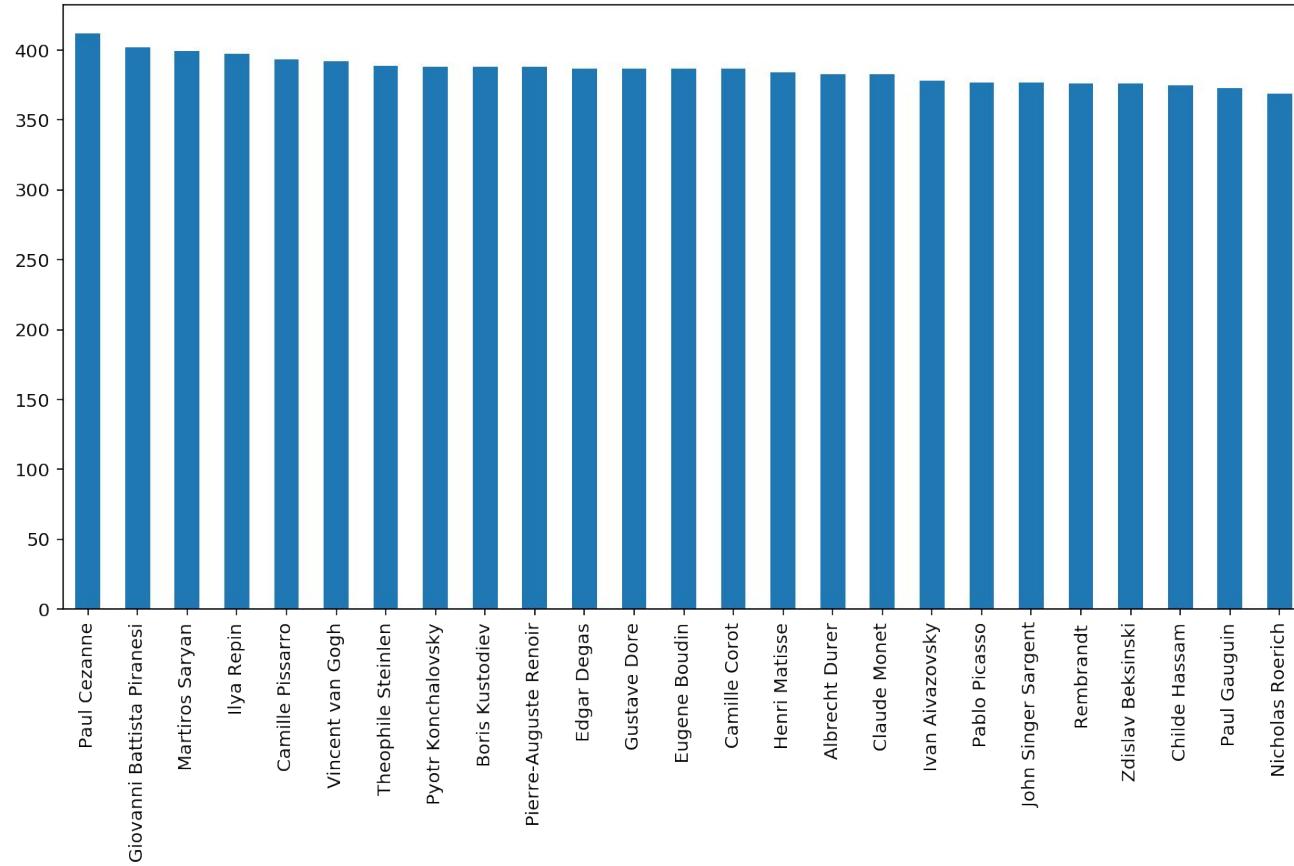
Top 25 Painting Genres (42 total unique)



Top 25 Painting Styles (135 total unique)



Top 25 Artists (2,074 total unique)



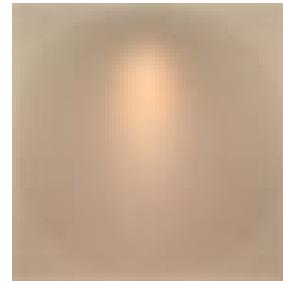
Data Preprocessing

Images range greatly in pixel size
and aspect ratio (anywhere from 30,000
 \times 29,605 to 283 \times 558)

- Initially resized to 72 \times 72, then to 28 \times 28
- Normalized pixel intensity values to 255
- ImageDataGenerator & .flow_from_directory

```
# loop over images and resize
for n, i, x in zip(num_files, filelist, filenames):
    train_img = load_img(source_path + filenames[i])
    train_img_resized = train_img.resize((72, 72))
    train_img_resized.save(dest_path + filenames[i])
    if n % 1000 == 0:
        print(n)
```

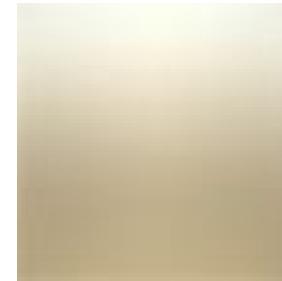
Baseline “Generative” Model



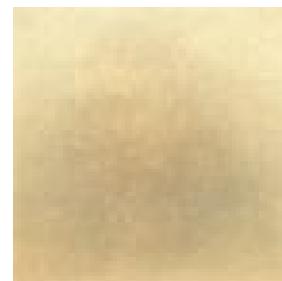
Portraits



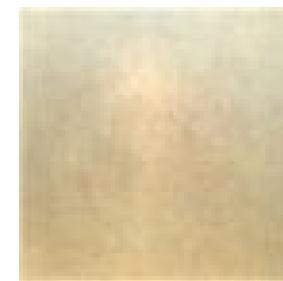
Landscapes



Impressionistic

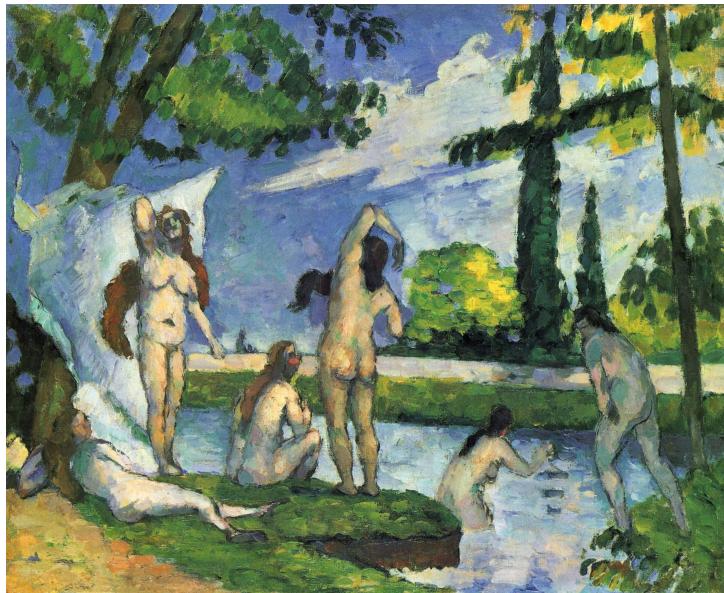


Van Gogh

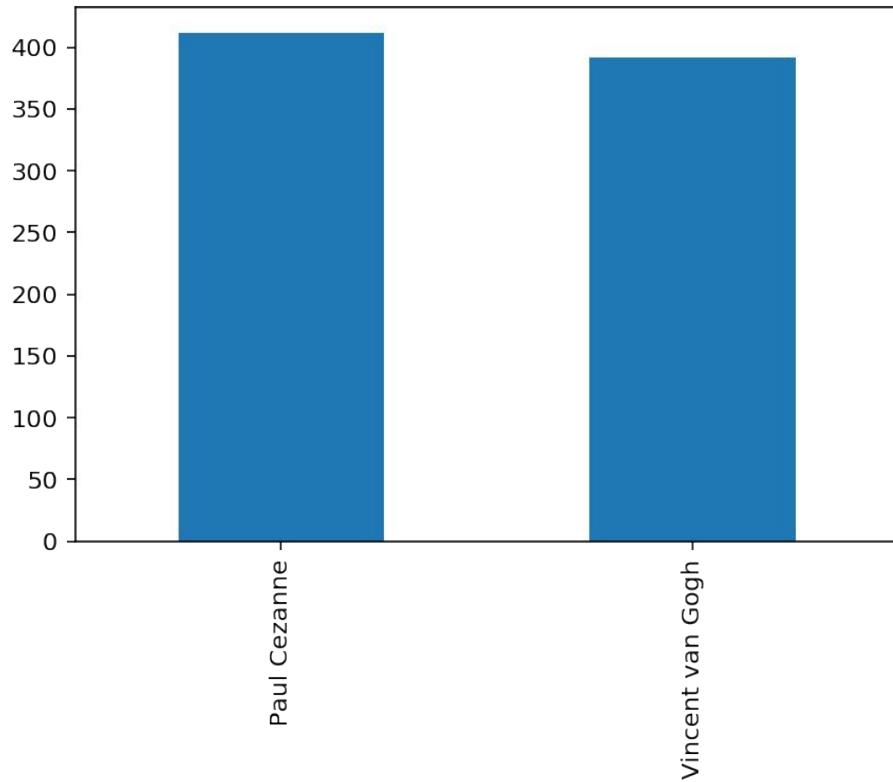


Cezanne

Subset Data for Simplicity



Class Imbalance (minimal)



- Training:
 - Cezanne – 412
 - van Gogh – 392
- Test:
 - Cezanne – 87
 - van Gogh – 102

Baseline Classification Model

Random Forest Classifier(n_estimators=10)

accuracy: 0.650793650794

	precision	recall	f1-score	support
0	0.66	0.75	0.70	102
1	0.64	0.54	0.59	87
avg / total	0.65	0.65	0.65	189

Baseline Model Error Analysis



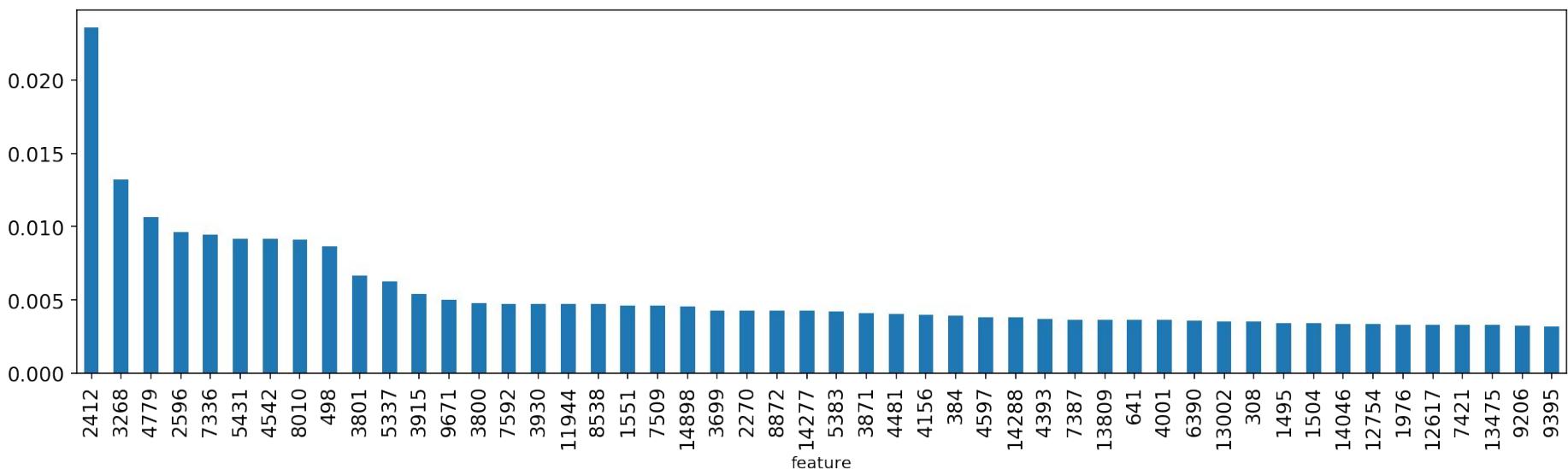
Prediction: Paul Cezanne (probability: 0.90)
True Label: Vincent van Gogh



Prediction: Vincent van Gogh (probability: 0.50)
True Label: Paul Cezanne

Baseline Model Feature Importances

Pixel positions? Likely too granular to generalize



Simple CNN

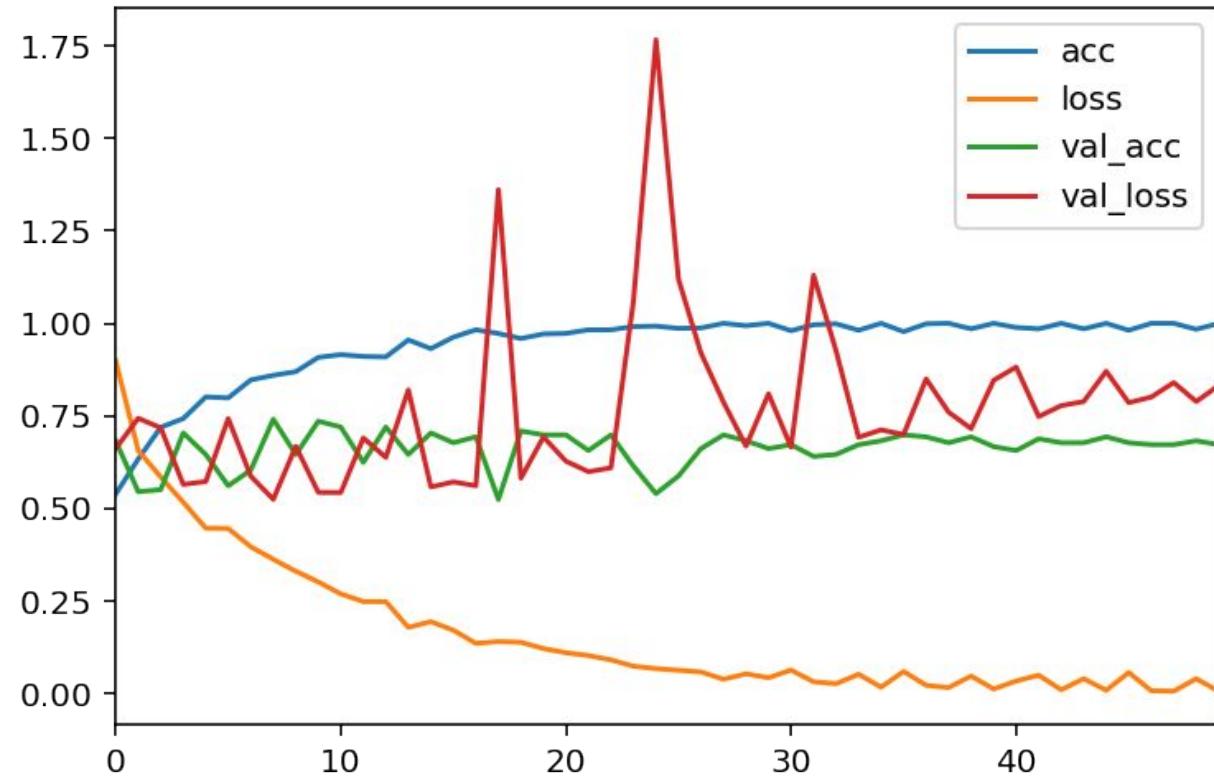
Model Architecture:
(Activation: ReLU)

Layer (type)	Output Shape	Param #
<hr/>		
conv2d_2 (Conv2D)	(None, 70, 70, 32)	896
activation_3 (Activation)	(None, 70, 70, 32)	0
max_pooling2d_2 (MaxPooling2D)	(None, 35, 35, 32)	0
flatten_2 (Flatten)	(None, 39200)	0
dense_2 (Dense)	(None, 1)	39201
activation_4 (Activation)	(None, 1)	0
<hr/>		
Total params: 40,097.0		
Trainable params: 40,097.0		
Non-trainable params: 0.0		

Simple CNN Performance

Model Accuracy:

0.67



Simple CNN Performance

Additional Evaluation Metrics

	precision	recall	f1-score	support
0	0.72	0.71	0.71	102
1	0.66	0.68	0.67	87
avg / total	0.69	0.69	0.69	189



I
WE NEED TO GO

DEEPER

“Deeper” CNN

```
model.add(Conv2D(32, (3, 3), input_shape=(72, 72, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

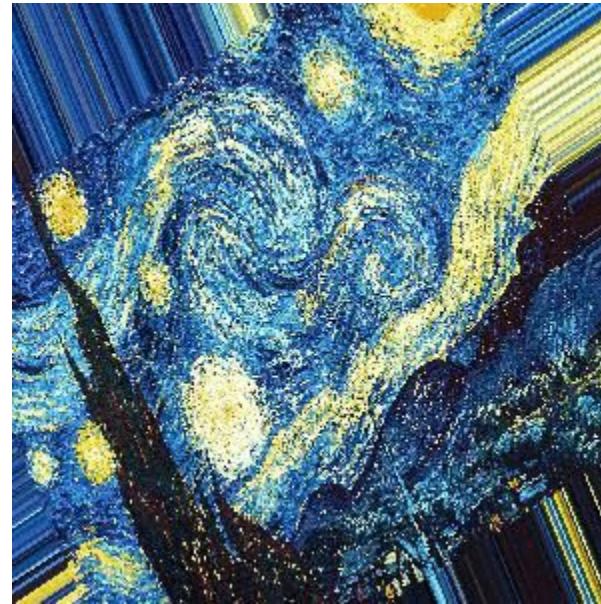
model.add(Conv2D(128, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(128))
model.add(Activation('relu'))
model.add(Dropout(0.25))

model.add(Dense(1))
model.add(Activation('sigmoid'))

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

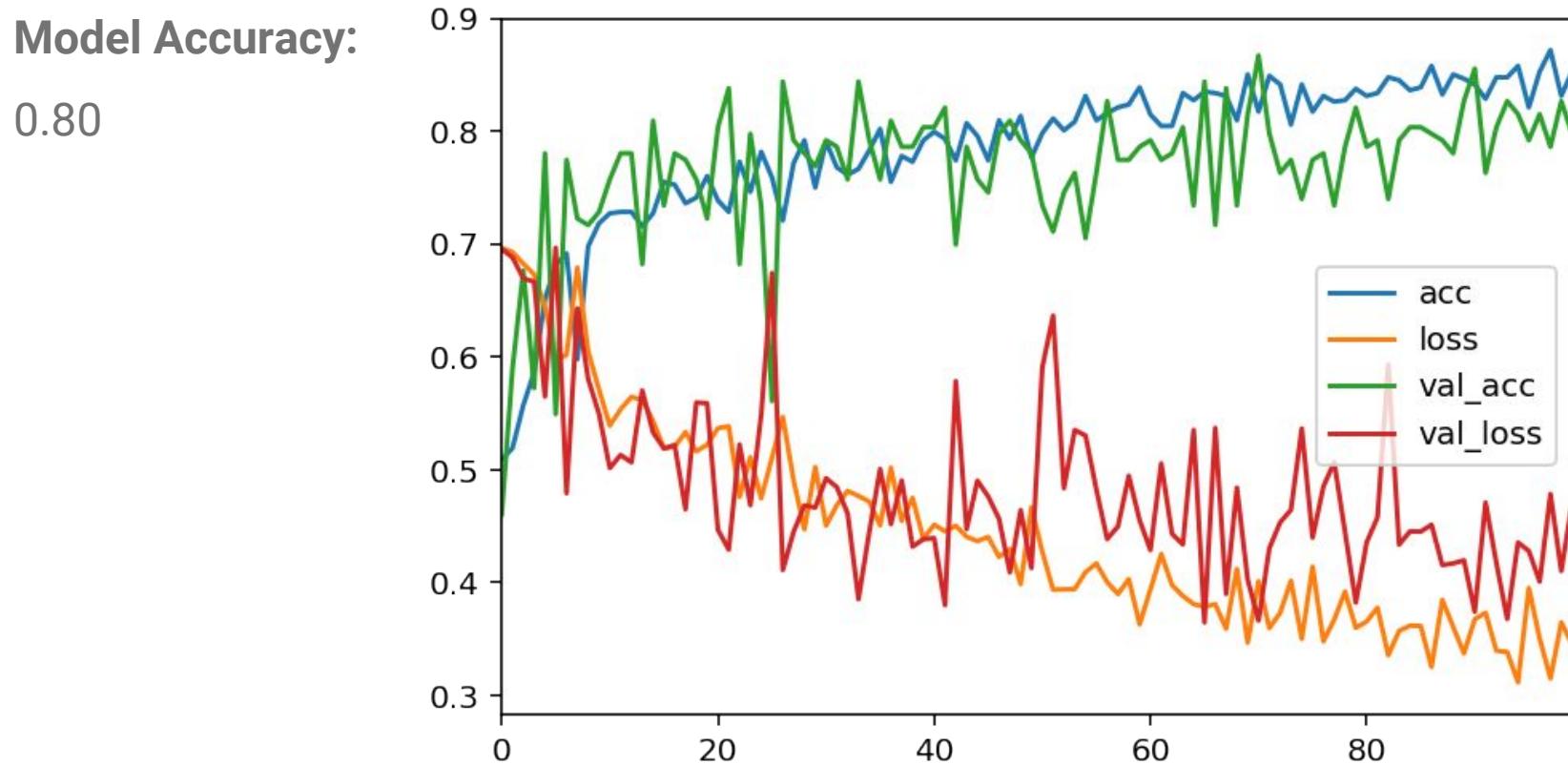
Data Augmentation



ImageDataGenerator & .flow_from_directory

Deeper CNN Performance

Model Accuracy:

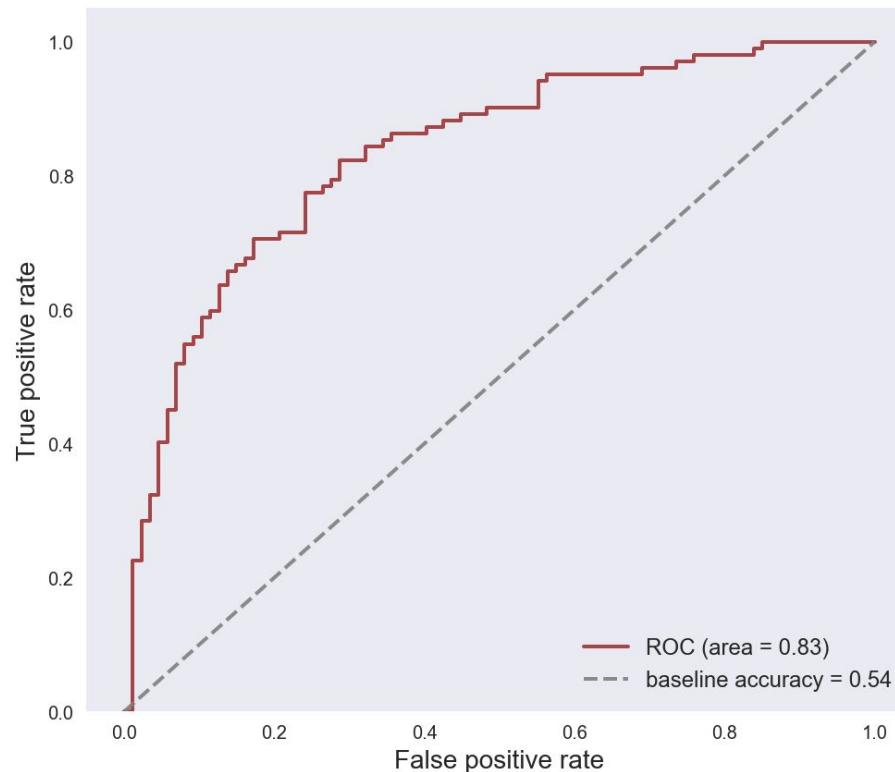


Additional Evaluation Metrics

	precision	recall	f1-score	support
0	0.71	0.83	0.76	87
1	0.83	0.71	0.76	102
avg / total	0.77	0.76	0.76	189

Deeper CNN Performance

Cezanne vs. van Gogh ROC curve



Deeper CNN Error Analysis



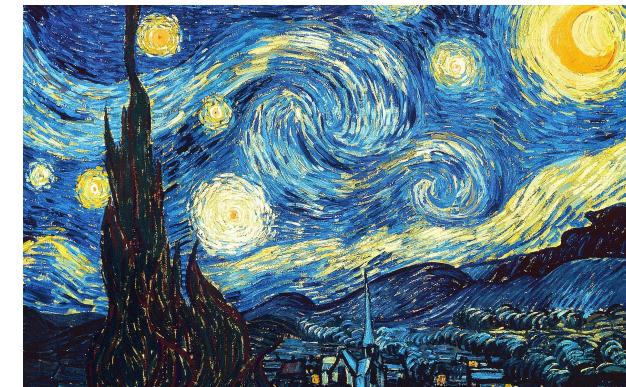
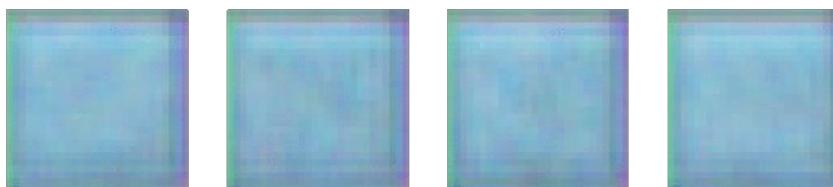
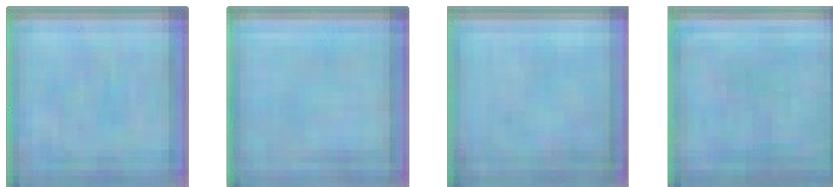
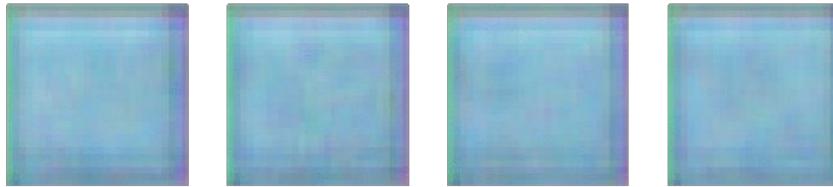
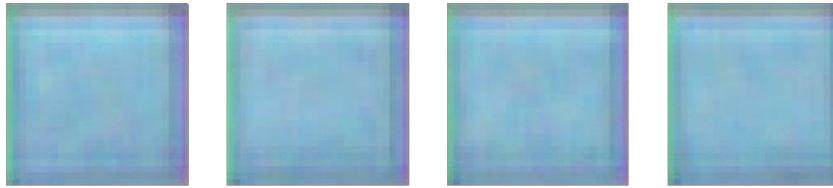
Prediction: Paul Cezanne (probability: 0.55)
True Label: Vincent van Gogh



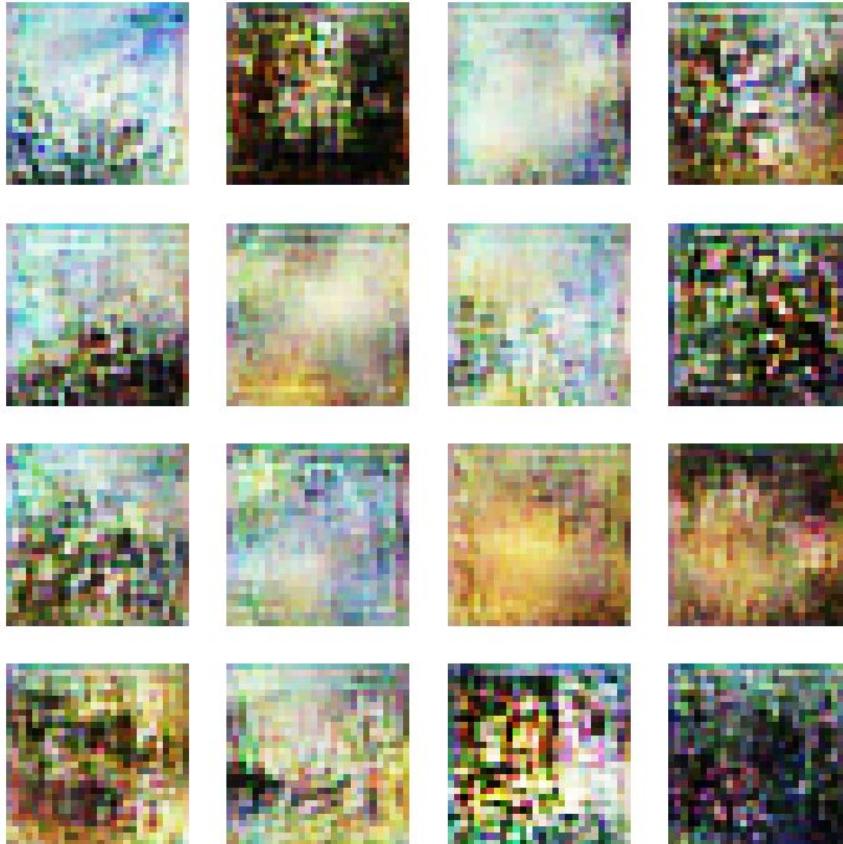
Prediction: Paul Cezanne (probability: 0.50)
True Label: Vincent van Gogh

LIVE DEMO TIME!

Fun with GANs!



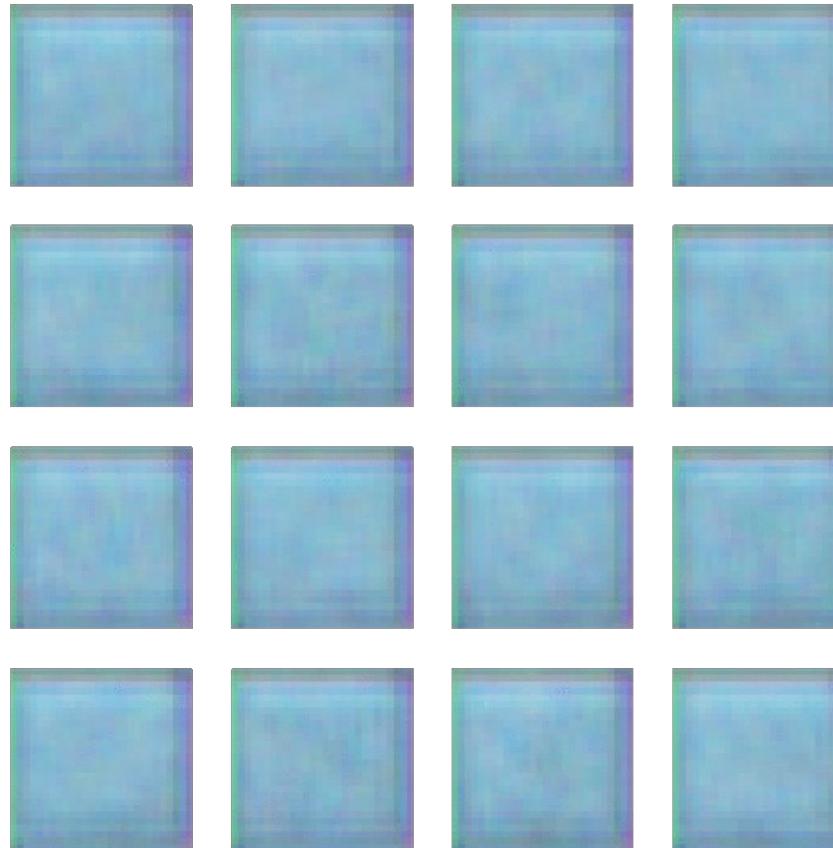
GANs creating... Impressionistic paintings?



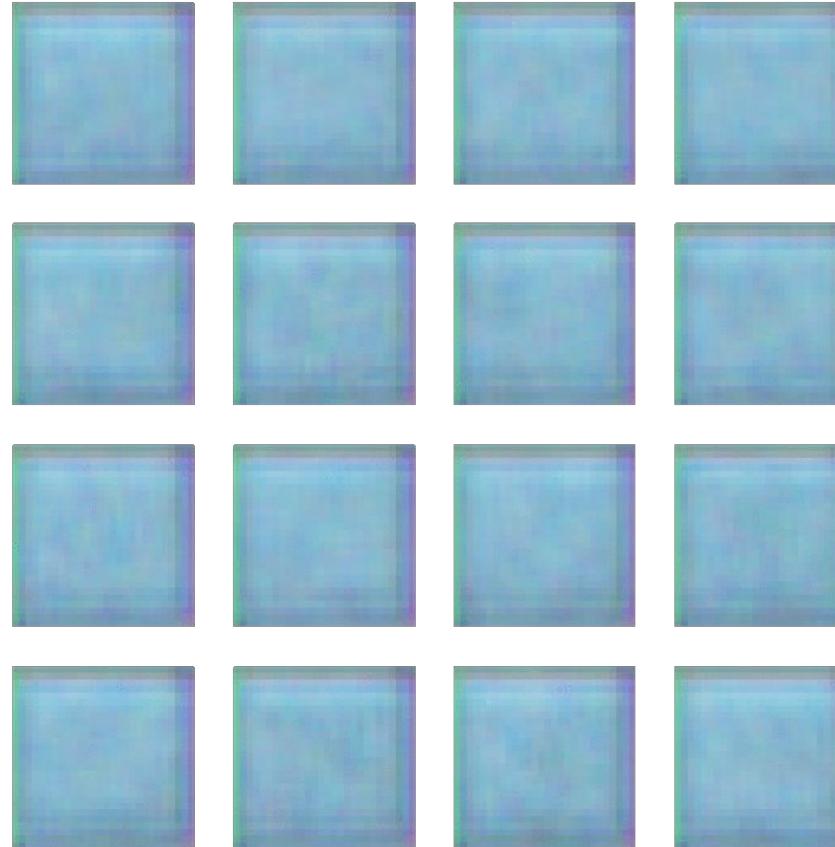
Claude Monet:
founder of French Impressionism



GANs creating... Impressionistic paintings?



GANs creating... Portraits?



Pokemon GO GAN

