

# Object Classification in Artwork

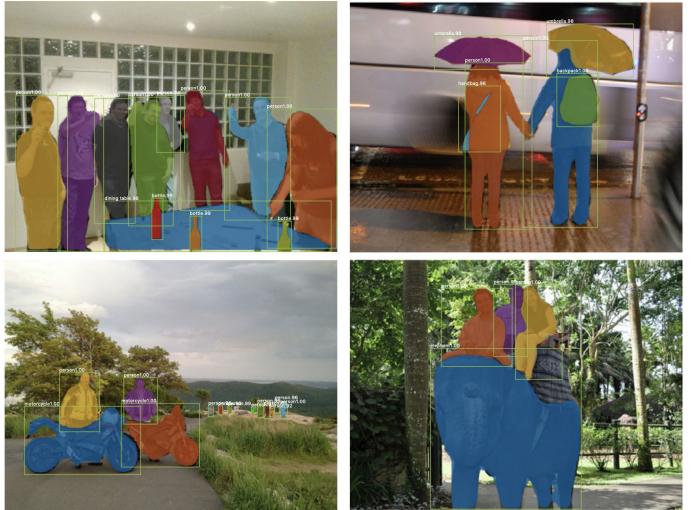
Isabel Tu (it2334), Dallas Scott (ds4015),  
Grayson Newell (gln2109)

# Motivation

Search an art database by object

Generate new compositions of pieces using the same objects

# Background



Mask R-CNN (2017)

<https://arxiv.org/pdf/1703.06870>

# Overview

# Pipeline: Part 1

Begin with classification/segmentation model trained on common objects database

Input artwork database & class list into model

Obtain dataset of cropped/labeled images



# Pipeline: Part 2

Train conditional GAN on segmented & labeled images

Input an object label & random noise vector into cGAN

Obtain generated image containing specified object

# Implementation

# Datasets

- Common Objects in Context (COCO) val2017 common images dataset

<https://cocodataset.org/#detection-2017>

- Stylized versions of val2017 dataset

- Kaggle – Best Artworks of All Time dataset (test dataset)

<https://www.kaggle.com/datasets/ikarus777/best-artworks-of-all-time>

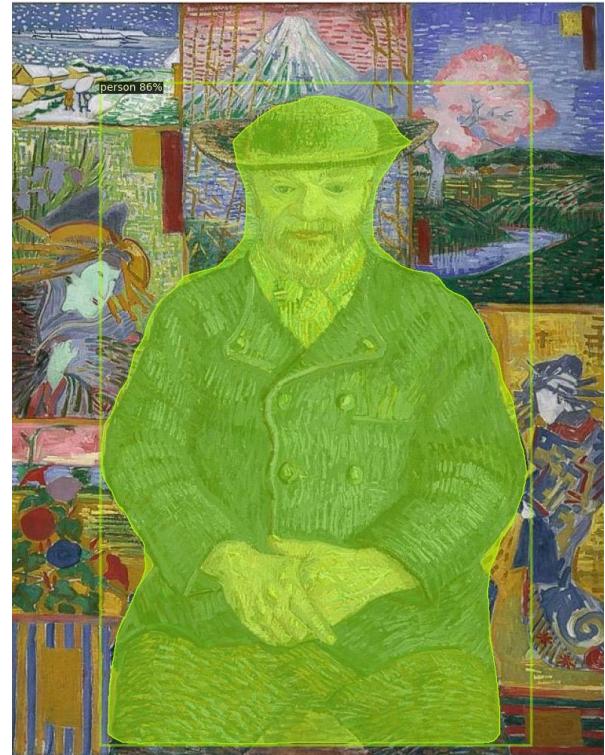
<b>id</b>	<b>name</b>	<b>years</b>	<b>genre</b>	<b>nationality</b>
0	Amedeo Modigliani	1884 - 1920	Expressionism	Italian
1	Vasily Kandinsky	1866 - 1944	Expressionism,Abstractionism	Russian
2	Diego Rivera	1886 - 1957	Social Realism,Muralism	Mexican
3	Claude Monet	1840 - 1926	Impressionism	French
4	Rene Magritte	1898 - 1967	Surrealism,Impressionism	Belgian
5	Salvador Dali	1904 - 1989	Surrealism	Spanish
6	Eduard Manet	1832 - 1883	Realism,Impressionism	French
7	Andrei Rublev	1360 - 1430	Byzantine Art	Russian
8	Vincent van Gogh	1853 - 1890	Post-Impressionism	Dutch
9	Gustav Klimt	1862 - 1918	Symbolism,Art Nouveau	Austrian
10	Hieronymus Bosch	1450 - 1516	Northern Renaissance	Dutch
11	Kazimir Malevich	1879 - 1935	Suprematism	Russian
12	Mikhail Vrubel	1856 - 1910	Symbolism	Russian
13	Pablo Picasso	1881 - 1973	Cubism	Spanish
14	Peter Paul Rubens	1577 - 1640	Baroque	Flemish
15	Pierre-Auguste Renoir	1841 - 1919	Impressionism	French
16	Francisco Goya	1746 - 1828	Romanticism	Spanish
17	Frida Kahlo	1907 - 1954	Primitivism,Surrealism	Mexican
18	El Greco	1541 - 1614	Mannerism	Spanish,Greek
19	Albrecht Dürer	1471 - 1528	Northern Renaissance	German
20	Alfred Sisley	1839 - 1899	Impressionism	French,British
21	Pieter Bruegel	1525 - 1569	Northern Renaissance	Flemish
22	Marc Chagall	1887 - 1985	Primitivism	French,Jewish,Belorussian
23	Giotto di Bondone	1266 - 1337	Proto Renaissance	Italian
24	Sandro Botticelli	1445 - 1510	Early Renaissance	Italian
25	Caravaggio	1571 - 1610	Baroque	Italian
26	Leonardo da Vinci	1452 - 1519	High Renaissance	Italian
27	Diego Velazquez	1599 - 1660	Baroque	Spanish
28	Henri Matisse	1869 - 1954	Impressionism,Post-Impressionism	French
29	Jan van Eyck	1395 - 1441	Northern Renaissance	Flemish
30	Edgar Degas	1834 - 1917	Impressionism	French
31	Rembrandt	1606 - 1669	Baroque	Dutch
32	Titian	1488 - 1576	High Renaissance,Mannerism	Italian
33	Henri de Toulouse-Lautrec	1864 - 1901	Post-Impressionism	French
34	Gustave Courbet	1819 - 1877	Realism	French
35	Camille Pissarro	1830 - 1903	Impressionism,Post-Impressionism	French
36	William Turner	1775 - 1851	Romanticism	British
37	Edvard Munch	1863 - 1944	Symbolism,Expressionism	Norwegian

# Object Segmentation: Detectron2

Detectron2: Image segmentation library by Facebook AI Research

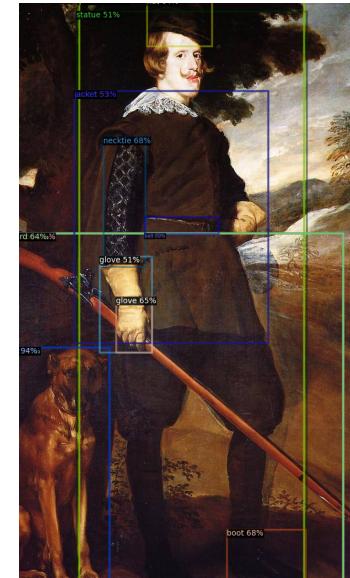
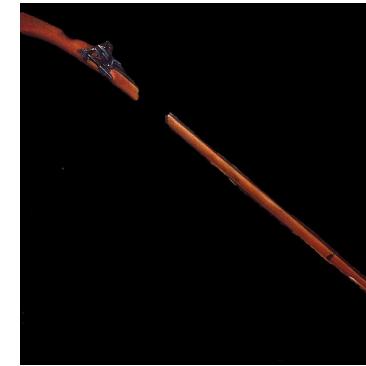
Mask R-CNN: Region-based CNN with pixel-level masks

R-CNN: Bounding Boxes Only



# Mask Object Extraction

- Segment Anything Model (SAM) used to generate masks of identified objects
- Swap after initial use of Mask R-CNN
- Masks rather than bounding boxes used to extract cropped images of objects
- Cropped masks used as input into GAN for new image synthesis



# Initial Attempt: COCO 2017 Dataset

Common Objects in Context – 2017 Validation Split:

5k annotated images; 80 Object classes / labels

05/04 17:25:26 d2.data.build]: Distribution of instances among all 80 categories:					
category	#instances	category	#instances	category	#instances
person	21554	bicycle	628	car	3836
motorcycle	734	airplane	286	bus	566
train	380	truck	828	boat	848
traffic light	1268	fire hydrant	202	stop sign	150
parking meter	120	bench	822	bird	854
cat	404	dog	436	horse	544
sheep	708	cow	744	elephant	504
bear	142	zebra	532	giraffe	464
backpack	742	umbrella	814	handbag	1080
tie	504	suitcase	598	frisbee	230
skis	482	snowboard	138	sports ball	520
kite	654	baseball bat	290	baseball gl..	296
skateboard	358	surfboard	534	tennis racket	450
bottle	2026	wine glass	682	cup	1790
fork	430	knife	650	spoon	506
bowl	1246	banana	740	apple	472
sandwich	354	orange	570	broccoli	624
carrot	730	hot dog	250	pizza	568
donut	656	cake	620	chair	3542
couch	522	potted plant	684	bed	326
dining table	1390	toilet	358	tv	576
laptop	462	mouse	212	remote	566
keyboard	306	cell phone	524	microwave	110
oven	286	toaster	18	sink	450
refrigerator	252	book	2258	clock	534
vase	548	scissors	72	teddy bear	380
hair drier	22	toothbrush	114		
total	72670				

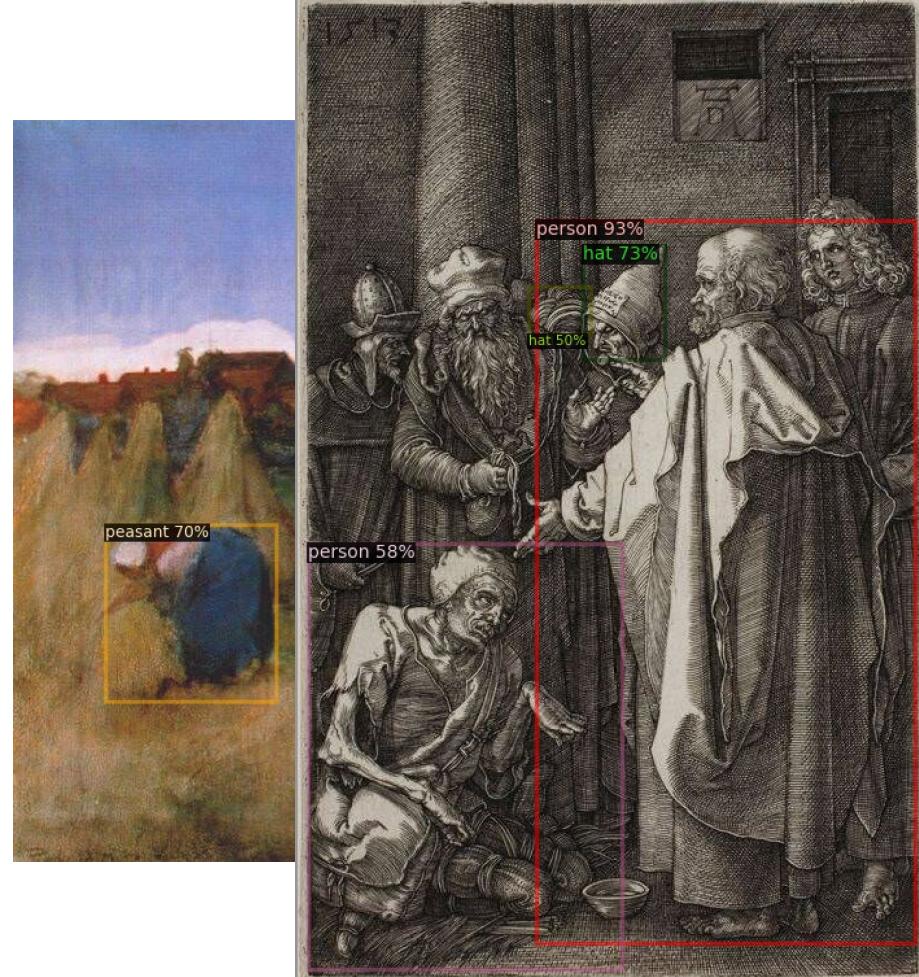


# Revision: Detic Model

Detic: 21k Classes, used with Detectron2

Detects more objects than COCO

Issue: Mainly from only a few classes



# Detic Model Revision

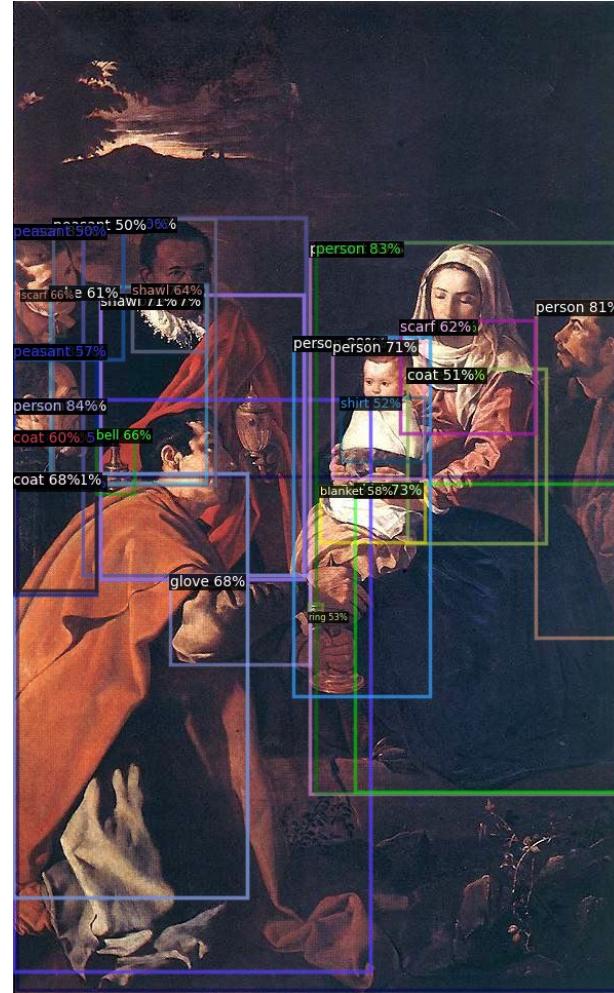
- Goal: Find a suitable custom vocabulary to use with Detic that pairs well with objects in artworks
- Initial attempt defaults to COCO's classes, only identifies limited objects (mainly “person”, “hat”, “coat”, “book”)
- Solution: Generate 1500 words using Synsets from WordNet

# WordNet Vocabulary

- Drawn from ‘artifact’, ‘natural\_object’, ‘person’, and ‘animal’ synsets
  - Generates 1500 generic terms that are best suited to objects in artworks

# Detic + WordNet

- Many more objects now being detected
- Fairly accurate, though some noted anomalies



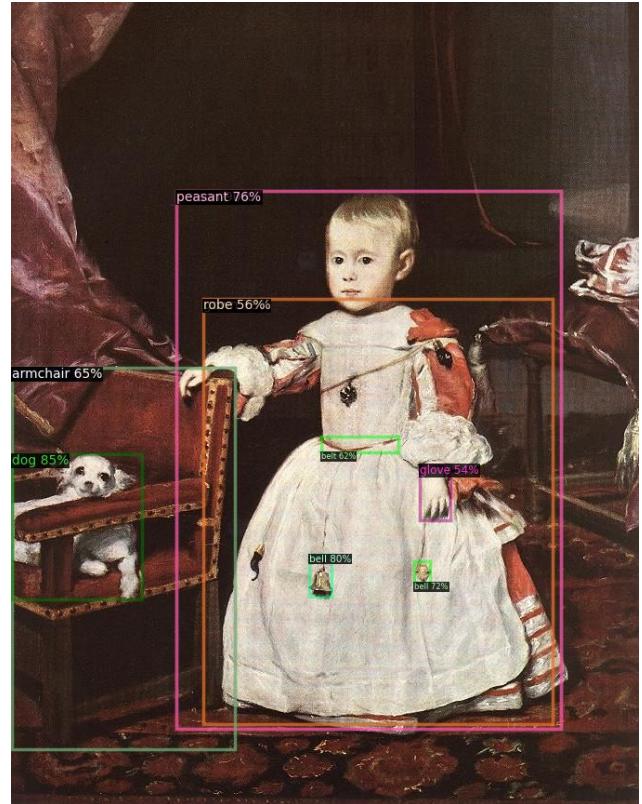
# Results: COCO + Stylization

- Initial attempts at testing with model trained on COCO dataset in conjunction with stylized versions did not identify many objects
- Too little overlap between COCO's 80 classes and objects found in classical artwork

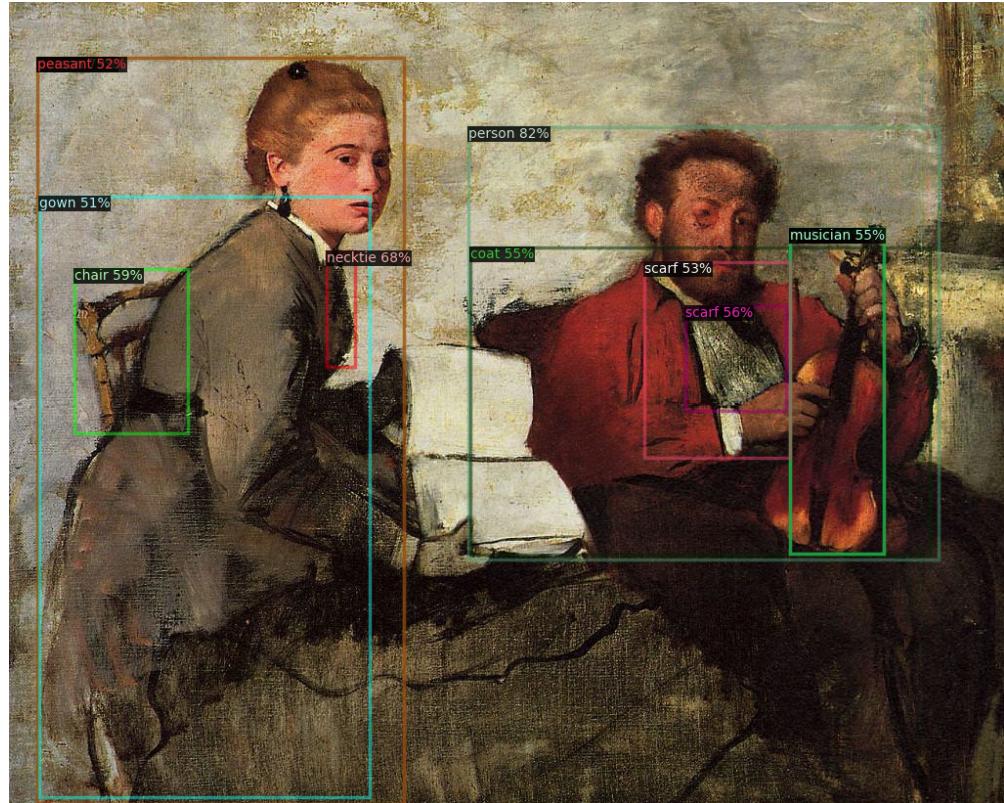
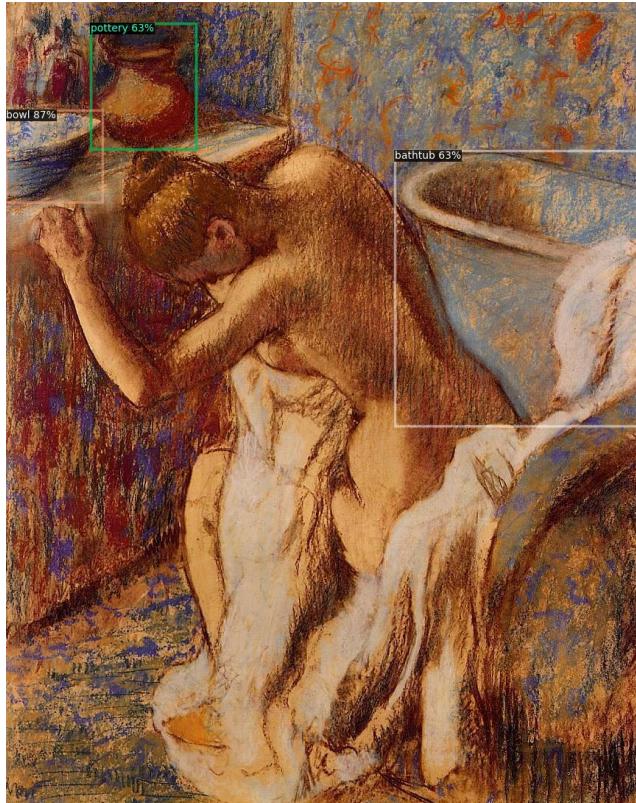


# Results: Detectron2 + Detic + Synsets

- Much more variety in objects being detected, with greater accuracy
- Still some noted anomalies, e.g., identification of “person” with “peasant”
- However, many objects are detected accurately



# Example Outputs – Detic/Detectron2



# Example Outputs – SAM Mask Crops



# GAN Attempt

Tried to create abstract renditions by training cGAN on cropped images

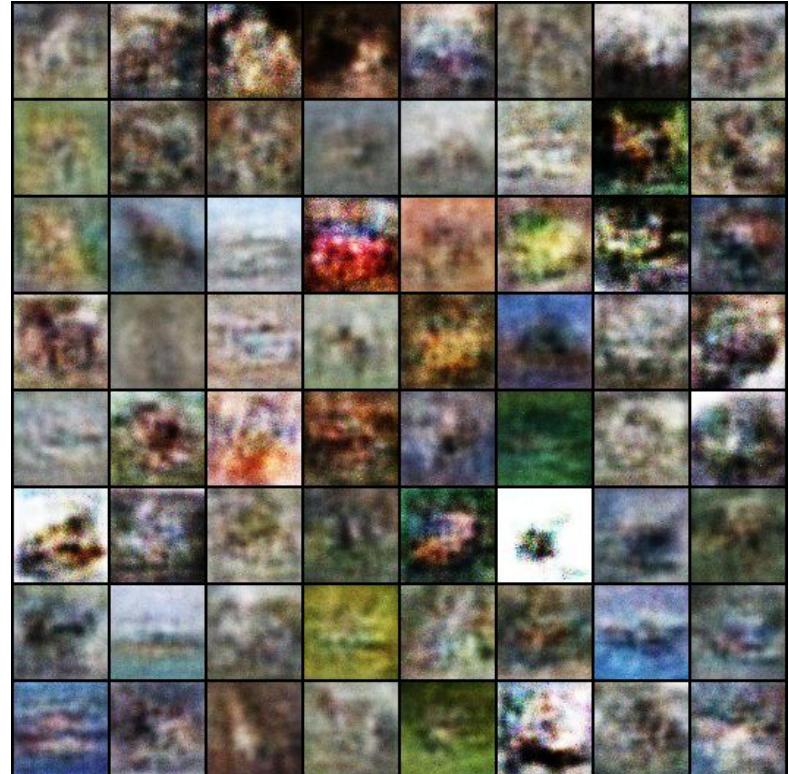
Insufficient time / processing power

```
class Generator(nn.Module):
    def __init__(self, nz, num_classes):
        super(Generator, self).__init__()
        self.label_emb = nn.Embedding(num_classes, num_classes)
        self.main = nn.Sequential(
            nn.Linear(nz + num_classes, 128),
            nn.ReLU(True),
            nn.Linear(128, 256),
            nn.BatchNorm1d(256),
            nn.ReLU(True),
            nn.Linear(256, 512),
            nn.BatchNorm1d(512),
            nn.ReLU(True),
            nn.Linear(512, 1024),
            nn.BatchNorm1d(1024),
            nn.ReLU(True),
            nn.Linear(1024, 3 * image_size * image_size),
            nn.Tanh()
        )

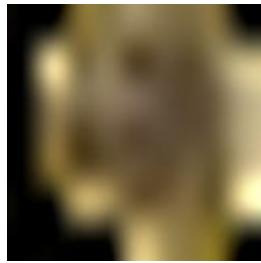
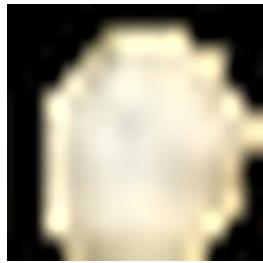
        def forward(self, z, labels):
            label_input = self.label_emb(labels)
            x = torch.cat([z, label_input], dim=1)
            return self.main(x).view(-1, 3, image_size, image_size)

class Discriminator(nn.Module):
    def __init__(self, num_classes):
        super(Discriminator, self).__init__()
        self.label_emb = nn.Embedding(num_classes, num_classes)
        self.main = nn.Sequential(
            nn.Linear(3 * image_size * image_size + num_classes, 512),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Linear(512, 256),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Linear(256, 1),
            nn.Sigmoid()
        )

        def forward(self, img, labels):
            label_input = self.label_emb(labels)
            img_flat = img.view(img.size(0), -1)
            x = torch.cat([img_flat, label_input], dim=1)
            return self.main(x)
```



# GAN Generated Buttons



Segmented from art

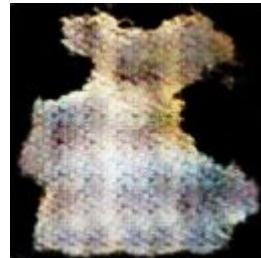


GAN generated

# GAN Generated Gowns



Segmented from art



GAN generated