Elgammal, A., Liu, B., Elhoseiny, M., and Mazzone, M.

CAN: Creative Adversarial Networks Generating "Art" by Learning About Styles and Deviating from Style Norms.

Presented at the 8th International Conference on Computational Creativity.

Partially reproduced by Will Ashe

12/7/19

Motivation

- To expand upon Generative Adversarial Network (GAN) success in the creation of novel images that emulate distinct periods or styles
- To use deep learning to create non-derivative art from given examples

GAN Background

- 2014: GANs were first proposed by Goodfellow et al.
  - Reformulated known concepts into minimax-based network
- 2014 – present: GAN research is shown to have applications in Deep Vision for generating realistic images, but also becomes popular in adversarial defense
- DCGAN (Radford et al., ICLR 2016) introduced deep learning to GANS with great success on attribute arithmetic and sample-to-sample transition (and 4900 citations to date)
- Some argue that GANS produce the most representative samples among generative networks, but this is hard to quantify

CAN Background

- Computable Creativity researchers have started examining GANs to learn to control the networks and create unique, interesting images
- "Arousal potential" – properties of stimulus patterns that induce an excited state (D. E. Berlyne)
  - For aesthetics, significant properties are the "collative variables" (novelty, suprisingness, complexity, ambiguity, and puzzlingness)
- Progression of art is driven by playing the arousal potential
  - Update style enough to be different and new, but with the same understanding of technique that drove the artist to make good art

Related Work

– (2014) Goodfellow et al. demonstrated and formalized GAN concept

– (2016) DCGAN implements GAN using a Deep Convolutional Network

– (2016) Wu et al. use GAN to create 3-D objects from images

– (2017) Antipov et al. modify photos by aging human subjects

– (2019) Karras et al. build StyleGAN, allowing for detailed and controlled synthesis of faces

– (2019) Yu and Canales use Long Short-Term Memory GAN to write melody from provided lyrics

Claim/Target Task

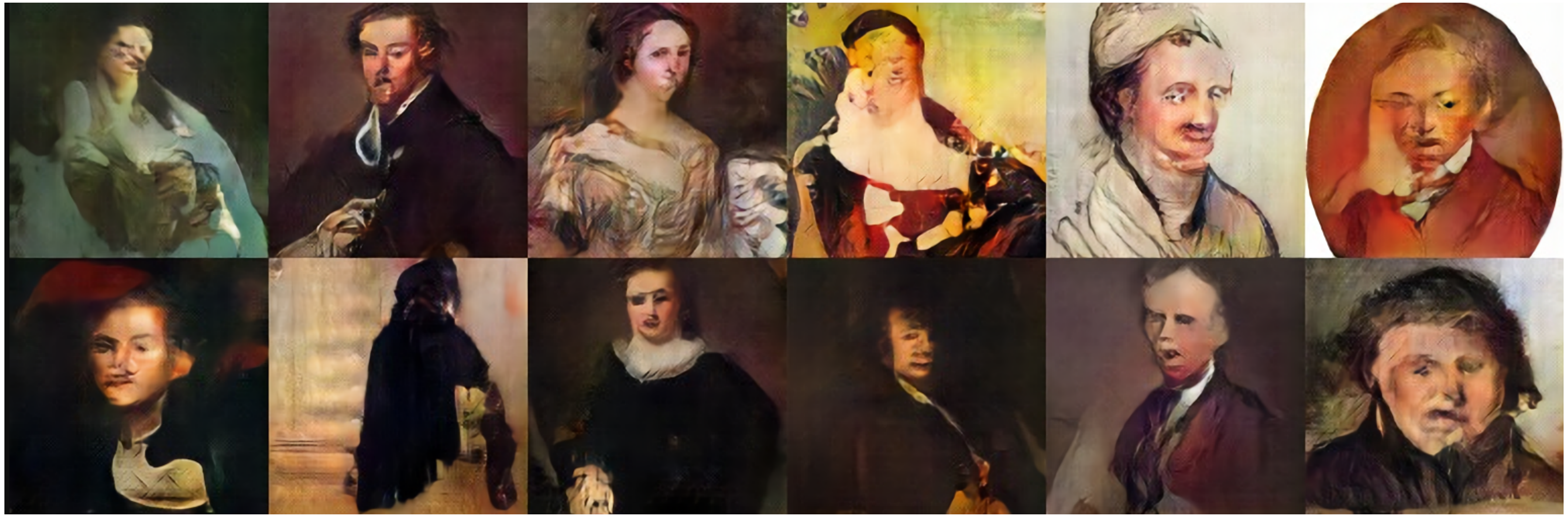
– To discriminate between real and fake artwork, and classify artwork by genre

– To produce images that qualify as art, given relevant training data

– To emulate the artistic intent without being stylistically derivative

– To achieve these metrics on a set of human judges

Figure showing WHY claim



Images created by art-DCGAN by Robbie Barrat – the images are in line with portrait techniques used 1600-1800 (with some variation), akin to mimicry as they lack the intentional response to contextualized understanding and emotion

Proposed Solution

- Modify the discriminator to perform a k-way loss classification for each of the k categories of art, in addition to the original GAN objective of determining the validity of the presented sample
- Modify the loss function of the network to prioritize the creative potential (specifically the "arousal potential")
  - Append "style classification" and "style ambiguity" loss

Implementation

- Similar to Deep Convolutional GANs (DCGANs)
- Generator CNN
  - Sampled (Gaussian) random noise is upsampled into a feature set
  - Six fractional-stride convolutional layers
  - Final reduction layer into image
  - Batch norm on non-Input/Output layers
  - Real/fake loss function of binary cross-entropy
    - Only half – see GAN formulation
  - Style Ambiguity loss used a categorical cross-entropy mixed with uniform distribution to produce tendency toward ambiguous class values

Implementation

- Similar to Deep Convolutional GANs (DCGANs)
- Discriminator CNN
  - Six Convolutional/LeakyReLU Layer Pairs create feature map
  - First head converts map to probability of image being real with a fully connected layer
  - Second head produces class probabilities from map with 3 fully connected layers
  - Real/fake loss function of binary cross-entropy
    - Only half – see GAN formulation
  - Classification loss using categorical cross-entropy (ish)

Implementation

$$
\begin{aligned}
&\min_G \max_D V(D,G) = \\
&\quad \mathbb{E}_{x,\hat{c}\sim p_{data}}[\log D_r(x) + \log D_c(c=\hat{c}|x)] + \\
&\quad \mathbb{E}_{z\sim p_z}[\log(1-D_r(G(z))) - \sum_{k=1}^{K}(\frac{1}{K} log(D_c(c_k|G(z)) + \\
&\qquad\qquad\qquad (1-\frac{1}{K}) log(1-D_c(c_k|G(z)))],
\end{aligned}
$$

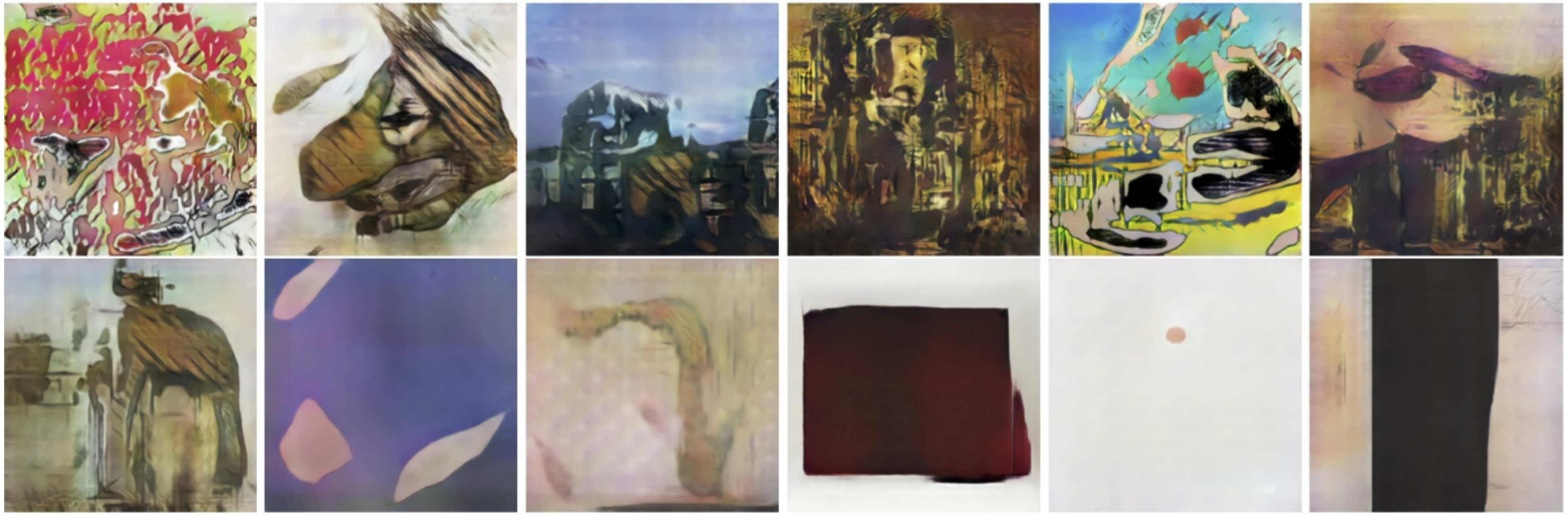
Data Summary

- Wikiart dataset
  - 81,449 paintings from 1,119 artists
  - Categorized into 25 genres
  - Freely available
- Used to train both proposed CAN and reference DCGAN networks

Experimental Results

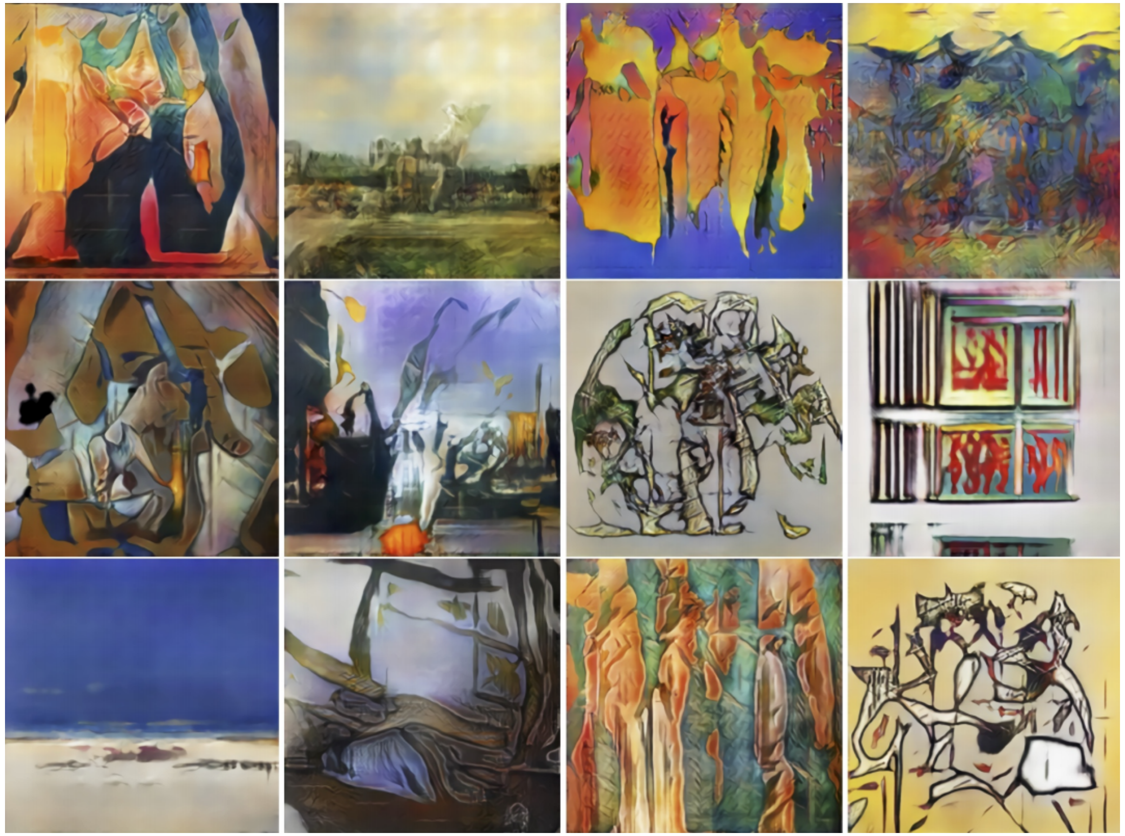
64x64 Art-Trained DCGAN

Experimental Results



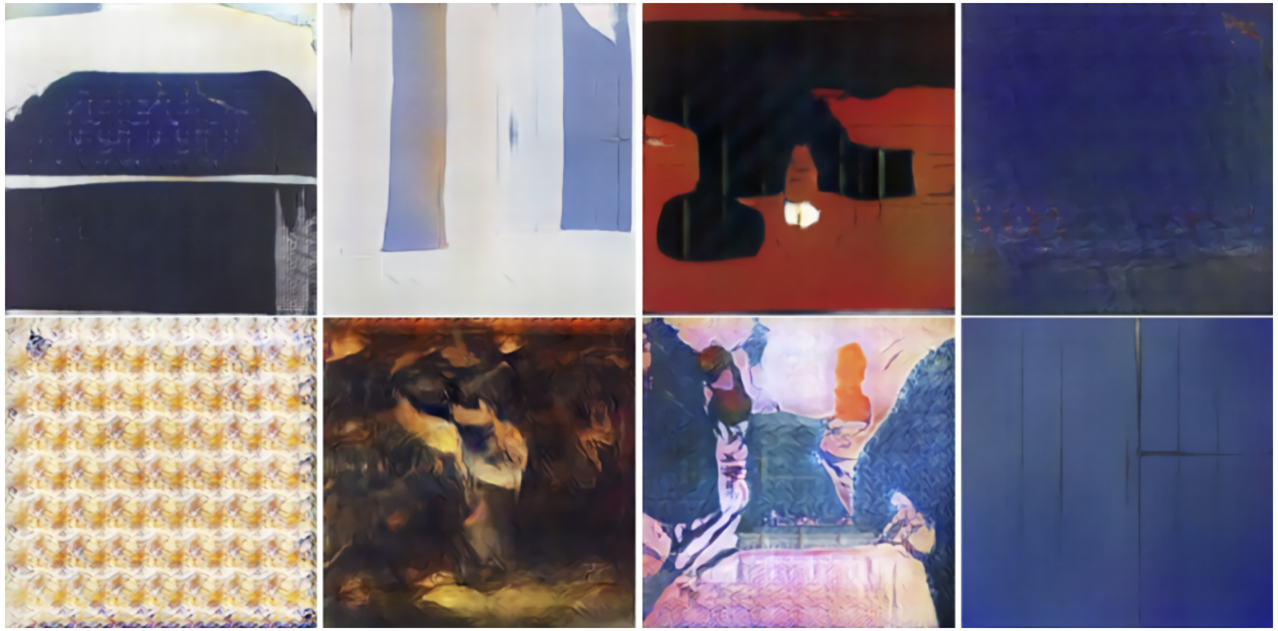
256x256 Art-Trained DCGAN

Experimental Results



CAN-produced images: Top ranked by humans

Experimental Results



CAN-produced images: Lowest ranked by humans

Experimental Analysis

- Survey experiments to determine:
  - Computer vs. Human created
  - Artistic quality
  - Novelty and aesthetic appeal
- Found:
  - CAN outperforms DCGAN in “human or computer” (75% vs 65% think human-made)
  - CAN found to have metrics of artistic merit on par with human-made art
  - Addition of style ambiguity loss to the generator improves the novelty and aesthetic appeal
    - 59% think CAN is more novel
    - 60% think CAN is more aesthetically appealing

On the Reproduction of Results

- Implementing a GAN is difficult
  - Building two networks where one encapsulates the other, then alternately training just the inner model (discriminator) or the super-model (generator and discriminator combined)
- Implementing a CAN is harder
  - Custom Loss Functions
  - Multi-class dataset
- Keras can only do so much to hide Tensorflow
  - spent multiple days resolving Tensorflow errors with code based on Keras-supplied examples
  - required switching versions of Tensorflow twice

GAN Building

- Batch Norm
  - Lots of opinions on worth, lots of results
  - Seems like having is better than not
- Layer Specs
  - Tanh generator output
  - LeakyReLu works well, ReLu can work for generator inner layers
  - Fractional convolutions for upsampling, convolutions for downsampling
  - No pooling layers
- Training
  - ADAM
  - Soft labels
  - Separate batches of training and testing data
  - Non-saturating game

For more: https://github.com/soumith/ganhacks, Goodfellow NIPS 2016 Tutorial, DCGAN Paper

Lessons Learned

- A lot about Keras and Tensorflow
- For unfamiliar structures, start with a simpler, working model before getting complex
  - i.e. basic GAN -> style class GAN -> CAN
  - Get the guts working, then build out features
  - Code-a-bit, test-a-bit for model building
- How to clean and implement a public dataset
  - Used the cs-chen/ArtGAN's Github labeled repo of varying image sizes
  - Cropped and resized to 256x256 input size
- How to batch the input
  - Necessary for GAN training
  - I also ran out of memory – size of dataset necessitated batched testing too

CAN: Conclusion and Future Work

- CAN effectively classifies style while creating artistic images that deviate from any one movement
- System has concept of style, but lacks the depth of semantic understanding of art and misses the meaning behind artistic movements and human subject matter
- Still, CAN outperforms contemporary art datasets on some of the survey metrics
  - Leaves open for more research into creating more meaningful works and examining the perceived qualities of these products

References

Elgammal et al. CAN: Creative Adversarial Networks Generating “Art” by Learning About Styles and Deviating from Style Norms (ICCC 2017).

Goodfellow et al. Generative Adversarial Networks (NeurIPS 2014)

Radford et al. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks (ICLR 2016).

Wu et al. Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling (NeurIPS 2016).

Antipov et al. Face Aging With Conditional Generative Adversarial Networks (ICIP 2017).

Akhtar and Mian. Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey (2017).

Karras et al. A Style-Based Generator Architecture for Generative Adversarial Networks (2019).

Yu and Canales. Conditional LSTM-GAN for Melody Generation from Lyrics (2019).