COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

GAN Image Detection: Up-Sampling Artifact & GAN Pipeline Emulator

CVPR Workshop on Media Forensics

Xu Zhang and Shih-Fu Chang

06/17/2019

Zhang, Xu, Svebor Karaman, and Shih-Fu Chang. "Detecting and Simulating Artifacts in GAN Fake Images." arXiv preprint arXiv:1907.06515 (2019)

COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science

Goals:

- Are there “artifacts” induced in the GAN image generation pipeline?
  - We explore a phenomenon and a theory related to **up-sampling artifact (checkerboard pattern)**.
- Are there ways to relax knowledge about the GAN models when training fake image classifier?
  - We propose a GAN pipeline emulator called AutoGAN.

Introduction

- 3 popular scenarios of image generation using GAN
  - Generating images from Noise
    - DCGAN [2016], ProgGAN [2017], StyleGAN [2018], BigGAN [2018]
    - Lack control of the generated content

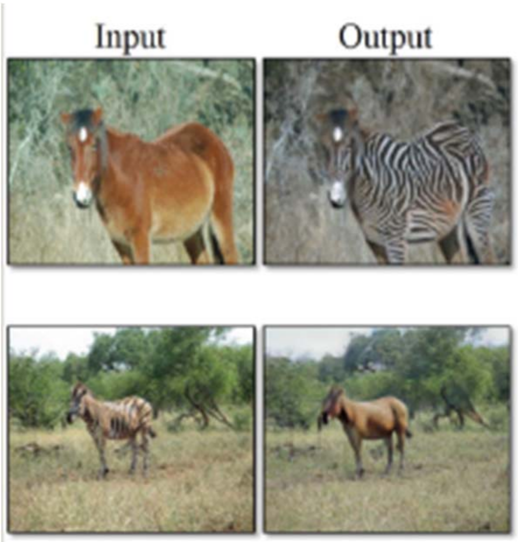
[Karras et. al, 2018a]
ProgressiveGAN

[Karras et. al, 2018b]
StyleGAN

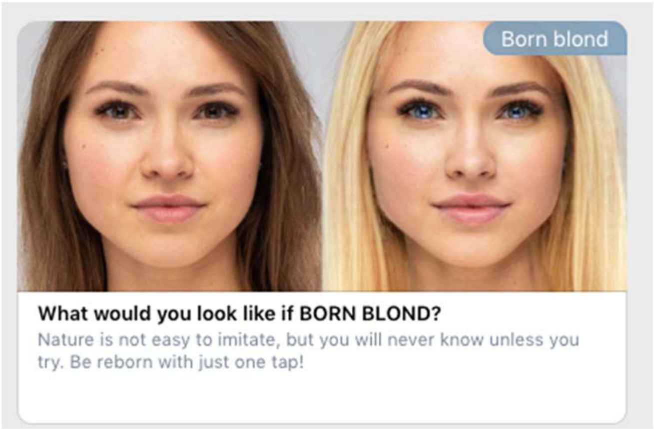
[Brock et. al, 2018]
BigGAN

Introduction

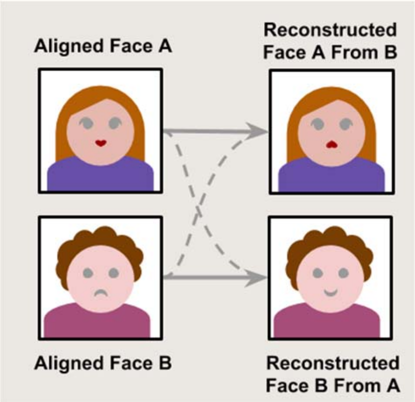
- 3 popular scenarios of image generation using GAN
  - Image to Image Translation: transfer images from one category/style to another
    - Pix2Pix [2016], CycleGAN [2017], StarGAN [2018], FaceSwap/DeepFake/FaceApp
    - Provide more control of the generated content



[Zhu et. al, 2017]
CycleGAN



FaceApp by Facebook



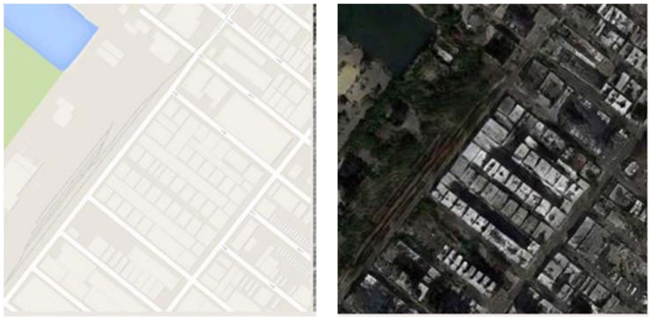
DeepFake

https://www.alanzucconi.com/2018/03/14/introduction-to-deepfakes/

Introduction

- 3 popular scenarios of image generation using GAN
  - Sketch to Image Translation
    - Pix2Pix [2017], CycleGAN [2017], GauGAN[2019]
    - Similar to image to image translation, but give even more controls to the generated content.

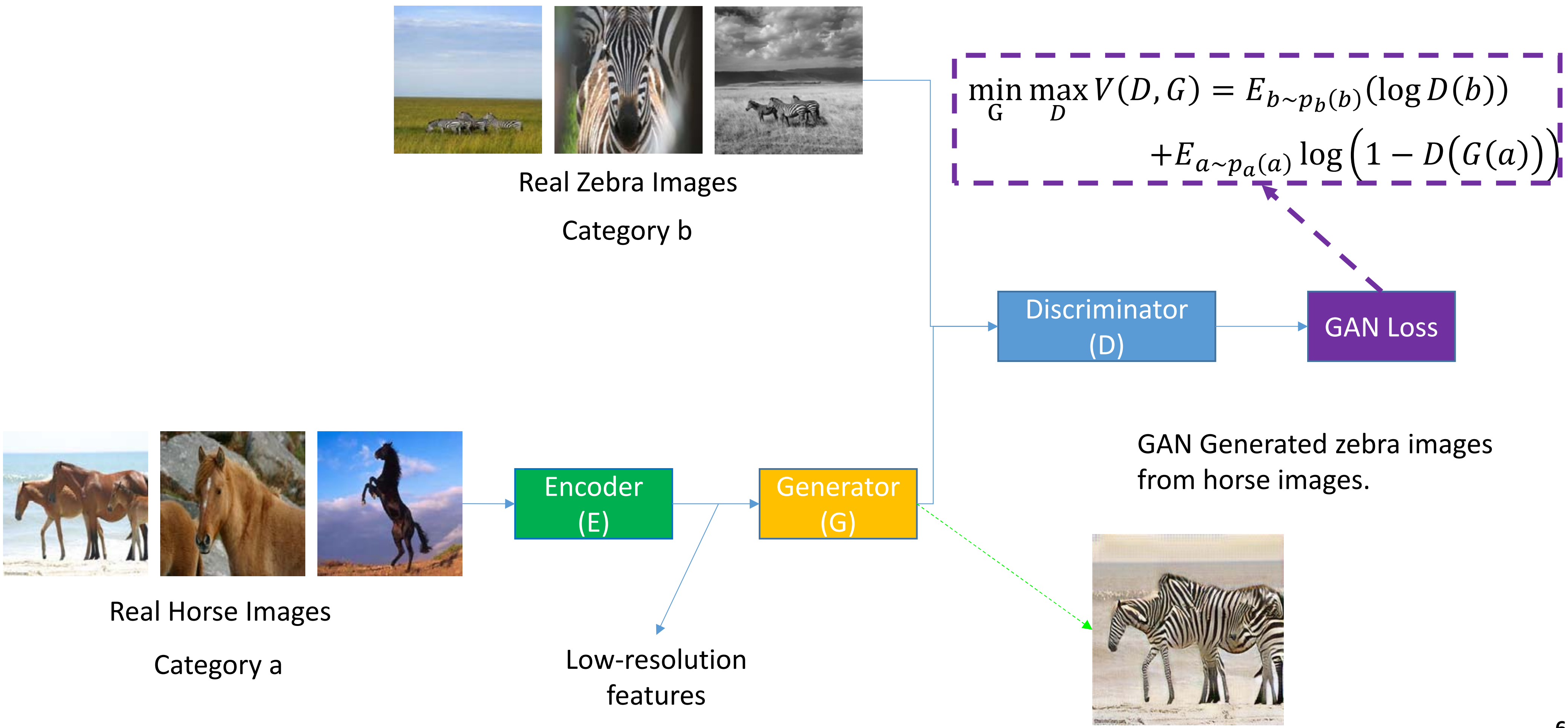
[Isola et. al, 2017]
Pix2Pix



[Zhu et. al, 2017]
CycleGAN

[Park et. al, 2019]
GauGAN

A Common Pipeline for Image2Image or Sketch2Image Transfer

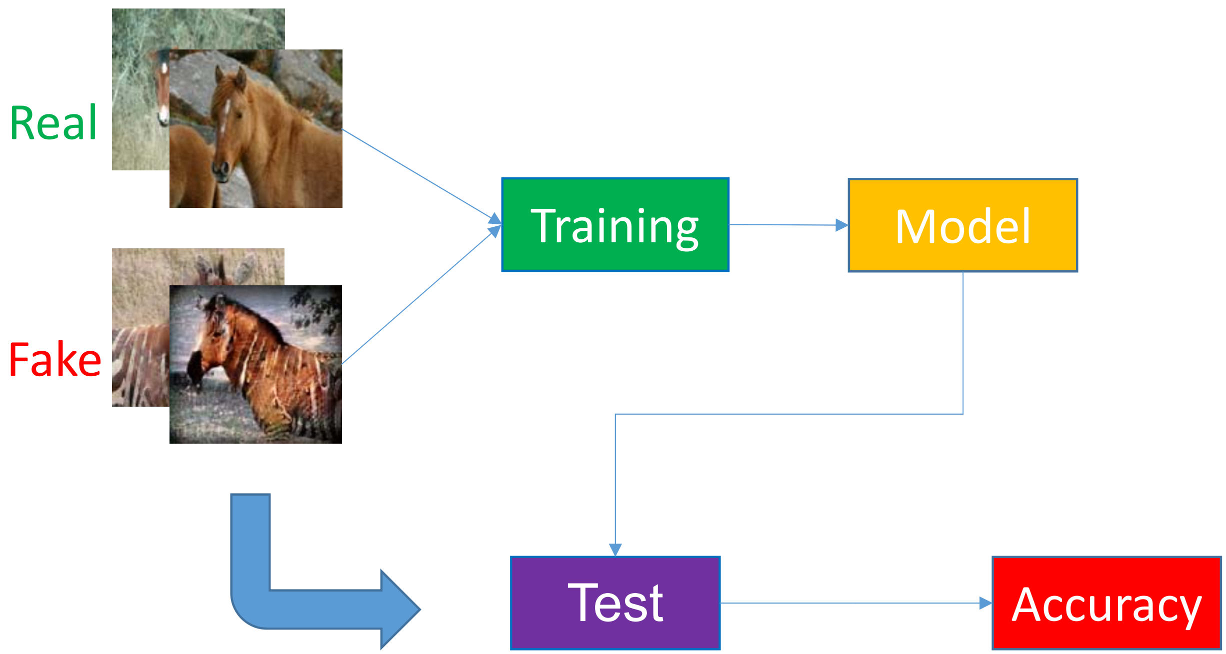


(An Incomplete) Review of Defense Tools

- Statistical Machine Learning + Feature Design
  - [Marra et. al 2018a] Use raw pixel and conventional forensics features. CNN, SVM, CycleGAN data
  - [Yu et. al 2018] Use raw pixel to detect noise2image GAN. CNN, ProGAN, SNGAN, and SAGAN
  - [Nataraj et. al 2019] Train with Co-Occurrence matrix. VGG-like, cycleGAN+StarGAN
  - [Marra et. al 2018b] Extract fingerprint from GAN. Correlation, cycleGAN+StarGAN
- Special Observations:
  - [McCloskey et. al 2018] GAN generated image doesn't have saturation region. SVM, NIST GAN challenge data
  - [Li et. al 2018] Deepfake video has no blinking eye. LSTM+VGG, Deep Fake
- Attribute Verification of Test Video against Real Video
  - [Agarwal et al 2019] Study the movement of the action unit of the leader from real video and see whether the generated video matches.

A Popular Baseline: Train a Fake/Real Image Classifier

- Design Issues
  - How to collect training samples?
  - What features to use?



Data Bias Pitfall

- In order to train a robust classifier we need, [Marra et. al 2018, Nataraj et. al 2019]
  - diverse training image content (avoid bias)
  - diverse generation models

Real

Horse

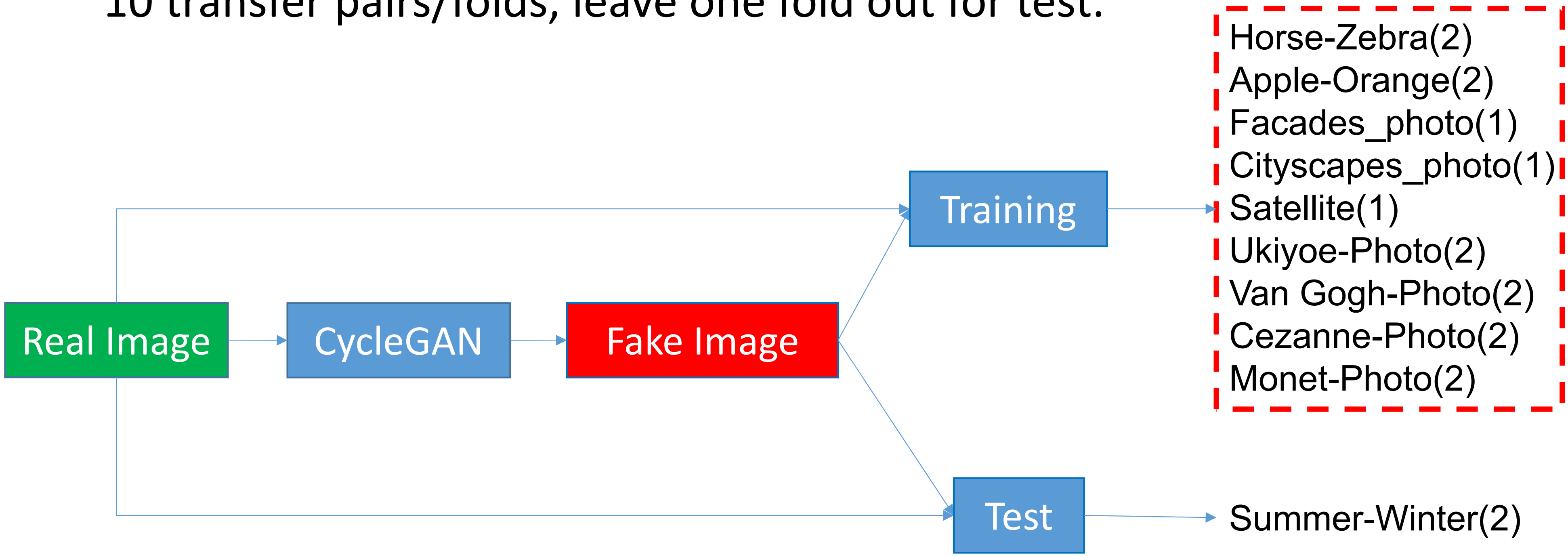
Fake

Generated Horse
From other category

Zebra

Leave-one-out strategy to avoid data bias

- Collecting real images and GAN generated image from a variety of sematic transfer pairs. [Marra et. al 2018, Nataraj et. al 2019]
- Train with leave-one-out strategy:
  10 transfer pairs/folds, leave one fold out for test.



Results (leave one out)

- Leave one out performs pretty well, but need training data from **diverse** sources.

Training	Test (Accuracy)										
	Horse-Zebra	Apple-Orange	Summer-Winter	Facades	CityScapes	Map	Ukiyo-e	Van Gogh	Cezanne	Monet	Avg.
DenseNet	79.1	95.8	67.7	99.0	93.8	78.3	99.5	97.7	99.9	89.8	90.1
Steganalysis feature	98.9	98.4	66.2	100.0	97.4	88.1	97.9	99.7	99.8	98.5	94.5
Cozzalino2017	99.9	100.0	61.2	99.9	97.3	99.6	100.0	99.9	100.0	99.2	95.7
XceptionNet	95.9	99.2	76.7	100.0	98.6	76.8	100.0	99.9	100.0	95.1	94.2
Nataraj2019	99.8	99.8	99.7	92.0	80.6	97.5	99.6	100.0	99.6	99.2	96.8

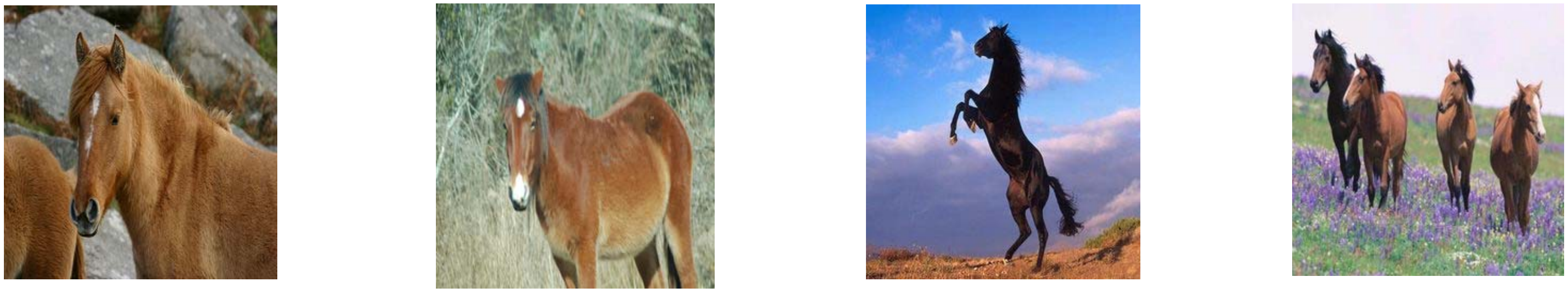
What if we train with one semantic class only?

- Performance downgrades significantly.
- Classifier does not generalize well to other categories

Training	Test														
	Horse	Zebra	Summer	Winter	Apple	Orange	Facades	CityScapes	Map	Ukiyoe	Van Gogh	Cezanne	Monet	Photo	Avg.
Horse	98.8	75.4	95.4	85.6	87.5	79.8	62.3	67.3	84.7	65.7	95.0	92.1	90.7	90.6	83.6
Zebra	87.7	98.8	95.4	92.1	57.2	57.8	50.5	53.9	50.1	66.3	89.7	64.5	89.2	90.3	74.5
Summer	88.8	87.3	98.7	99.8	76.1	76.3	50.9	59.5	77.0	94.5	91.9	93.7	90.5	94.3	84.2
Winter	84.6	82.7	98.2	98.9	74.7	69.6	50.0	50.4	88.5	96.7	82.5	93.3	87.3	92.9	82.2

Is It Recognizing Real vs. Fake images?

Real Horse Images



Generated Horse Images from Zebra Images

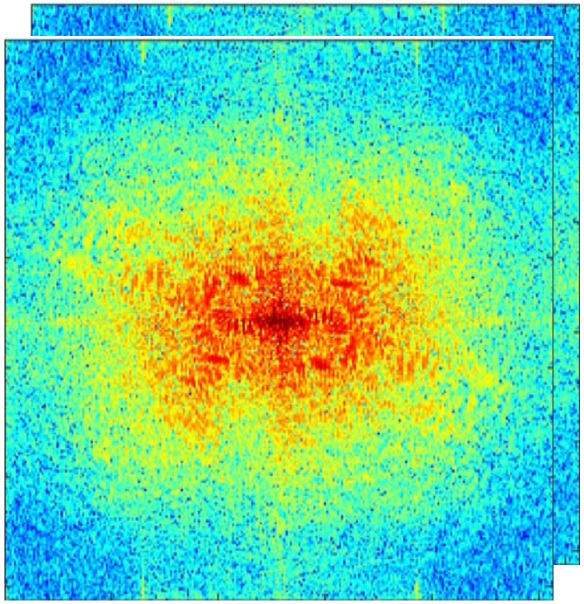
- Or is it recognizing other differences?
    - High-quality horse vs. low-quality horse
    - Horse habitats vs. zebra habitats

What if we change it to the frequency domain?

- Use frequency-domain data as input to the classifier
- Convert 3 RGB channels to the spectrum of each channel as input.

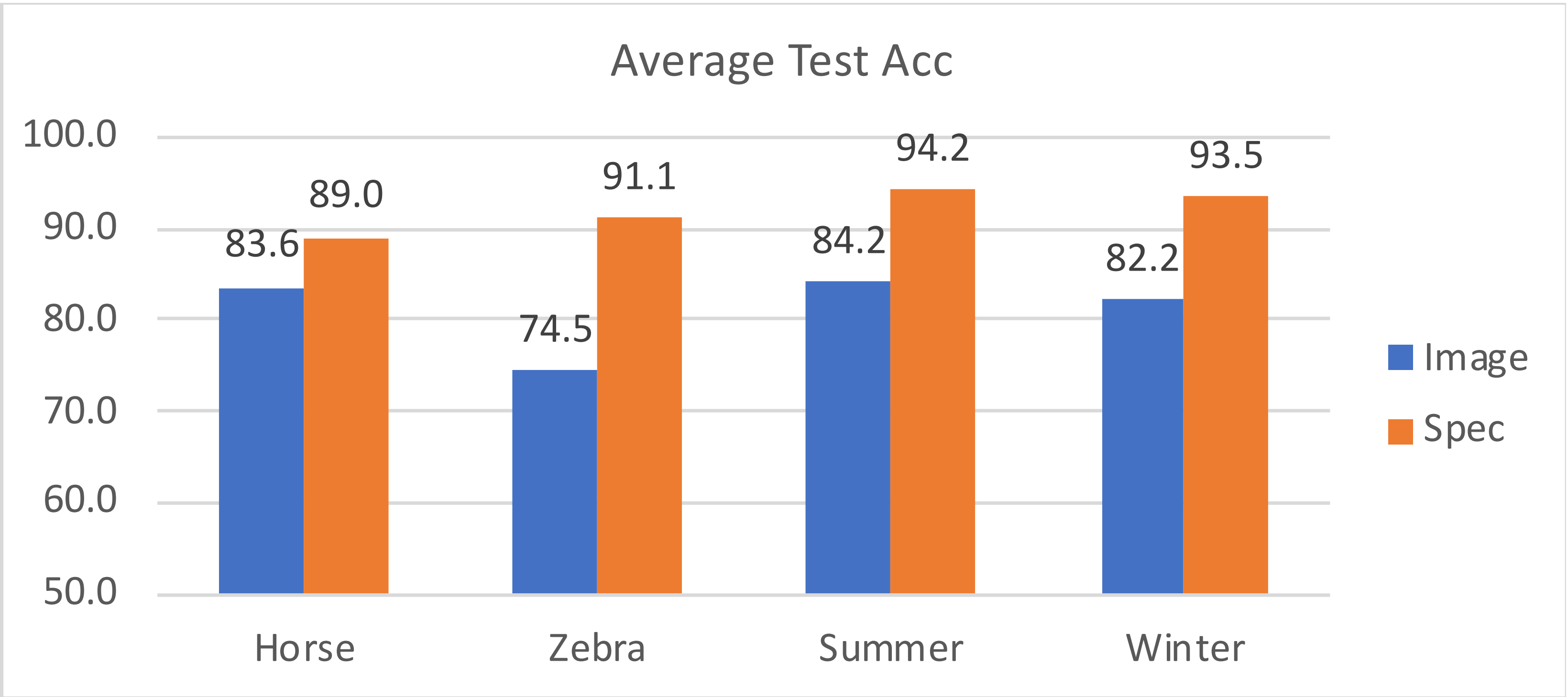
Generated Image

Spectrum of the Generated Image
(3 channels RGB)



Directly Train with DFT Spectrum, using one class only

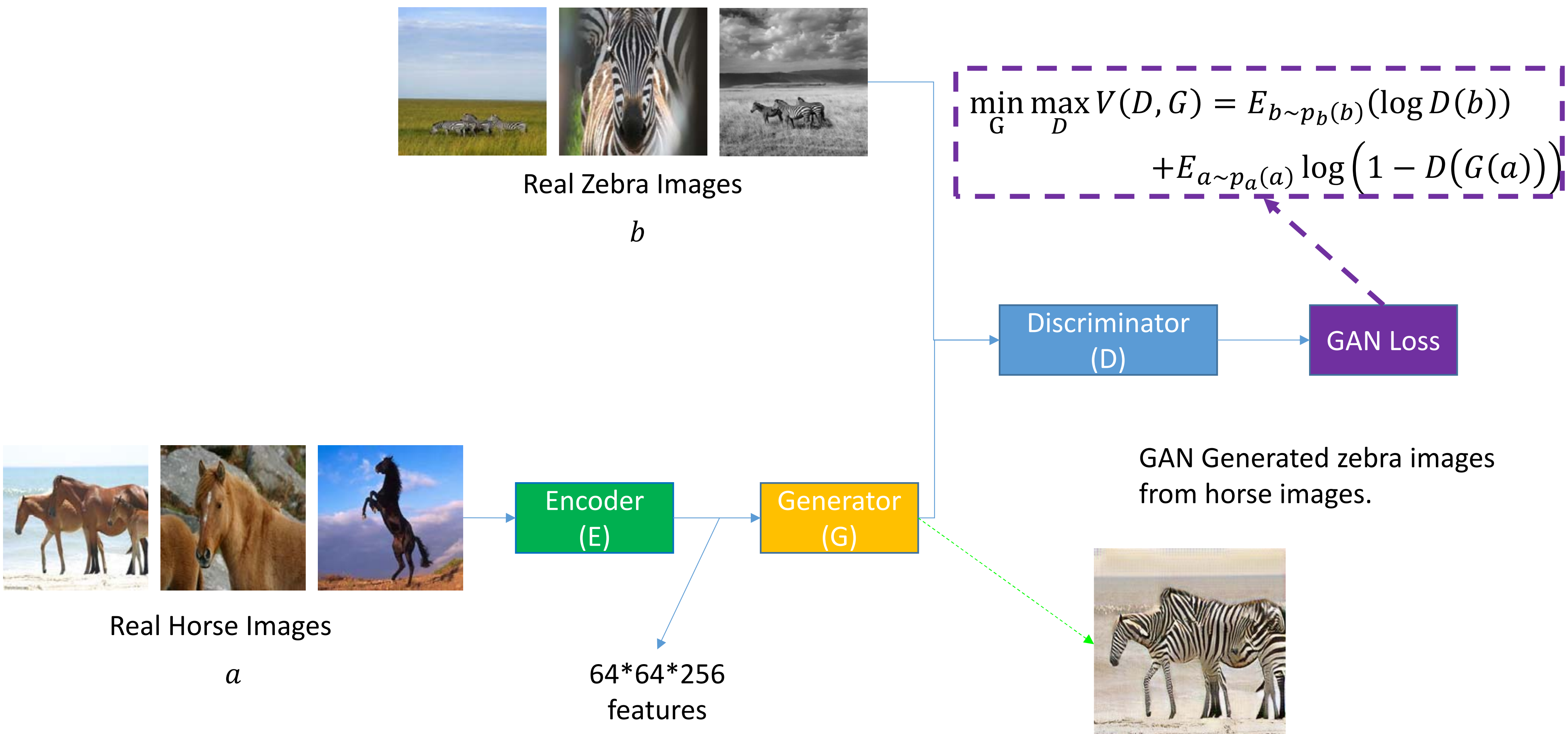
- Performance is significantly improved
- The generalization ability is much better than training with RGB images



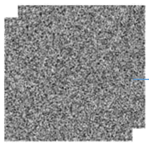
Explaining the Success of Spectrum Input

- Explore the signal processing model underlying the GAN synthesis pipeline

Revisit the Pipeline in Image2Image Transfer



Inside the GAN Generator



Feature low-resolution 64x64x256

Convolution layer 1 → Up-sample layer 1 → … → Convolution layer N → Up-sample layer N → Convolution layer N+1 →

Output Image High Resolution

Up-sample layer

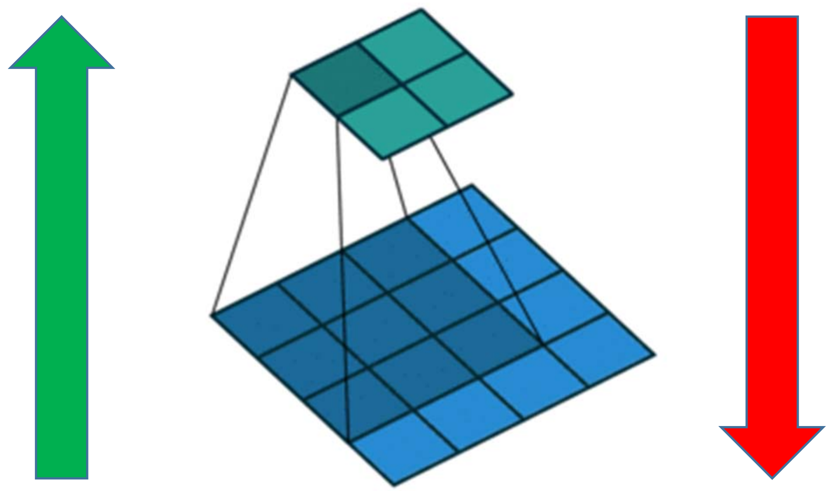
- Transposed Convolution [Zhu et. al, 2017] [Choi et. al, 2018]
- Nearest Neighbor Sampling [Karras et. al, 2018a]
- Bilinear Sampling [Karras et. al, 2018b]

Convolution vs. Transposed Convolution (Deconvolution)

Output

Convolution

Input



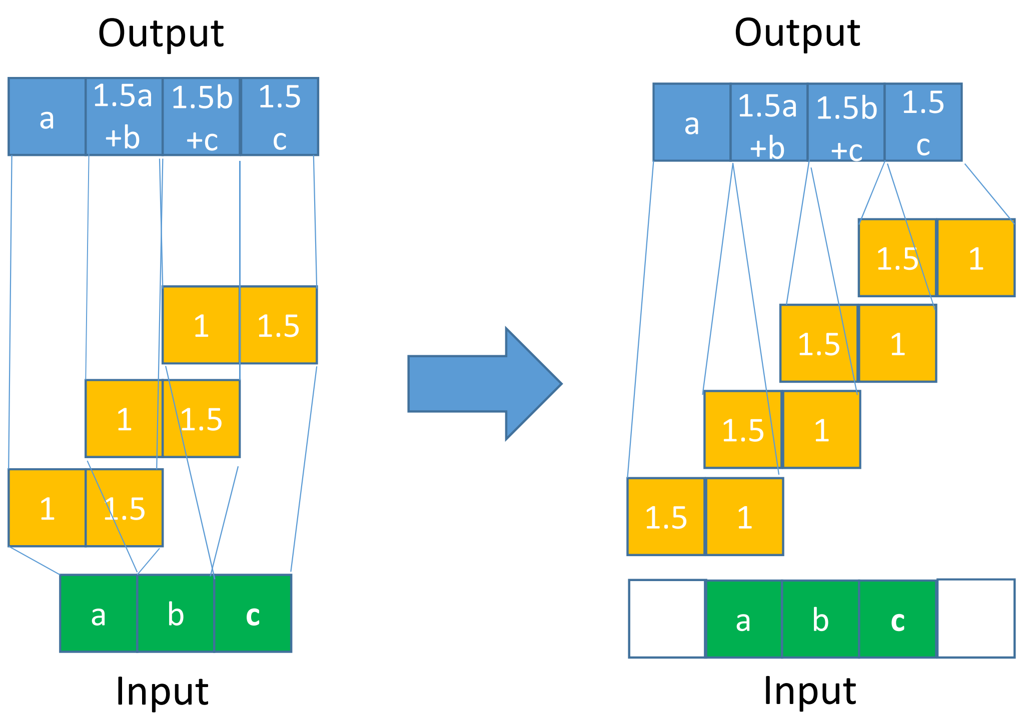
Input

Transposed Convolution

Output

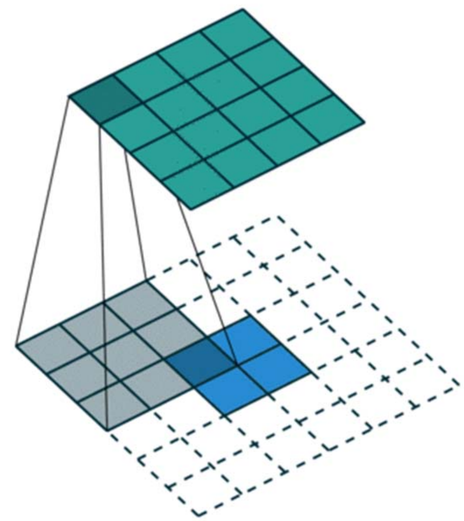
http://deeplearning.net/software/theano_versions/dev/tutorial/conv_arithmetic.html

Transposed Convolution = Zero Padding Convolution



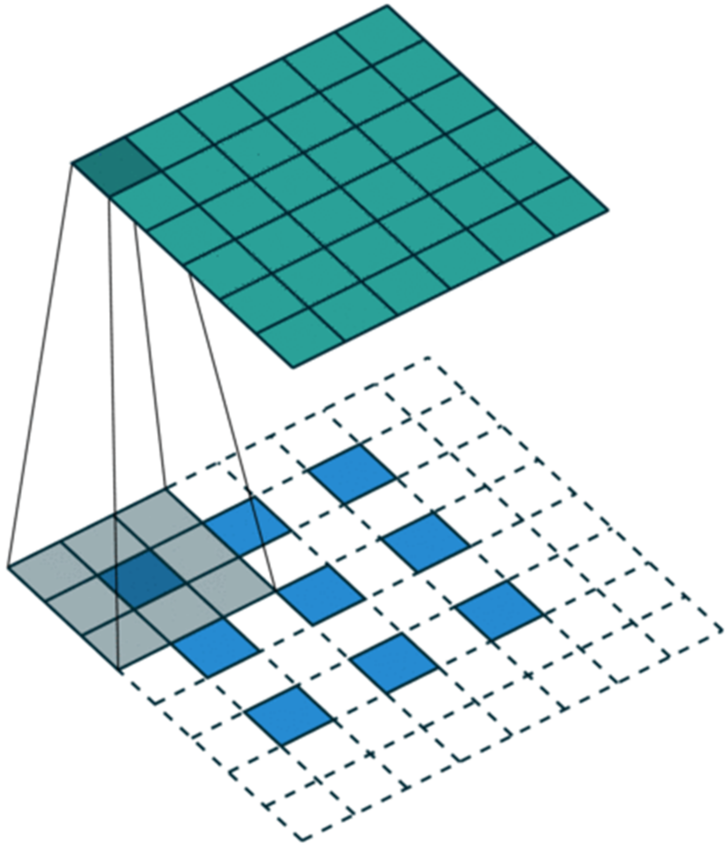
Transposed Convolution

Zero Padding Convolution



http://deeplearning.net/software/theano_versions/dev/tutorial/conv_arithmetic.html

Stride 2 Transposed Convolution for Up-sampling



Input: 3*3
Output: 6*6
Kernel: 3*3
Stride: 2

http://deeplearning.net/software/theano_versions/dev/tutorial/conv_arithmetic.html

Zero insertion → spectrum artifact

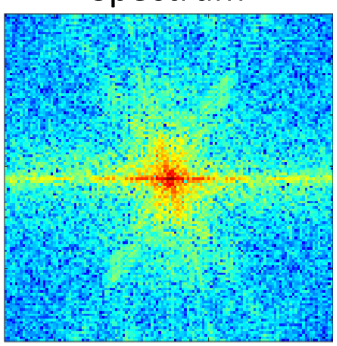
Low-resolution image

128

128

Spectrum

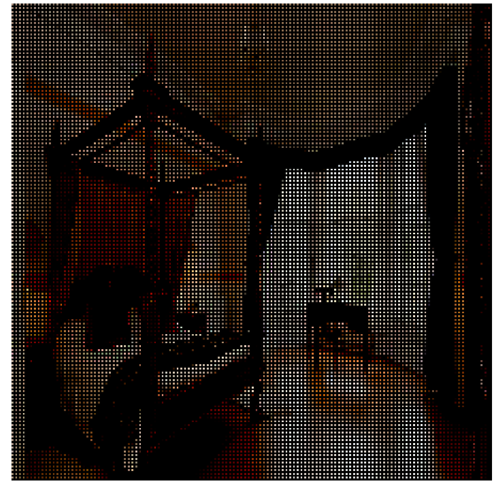
128



128

Zero Inserted image

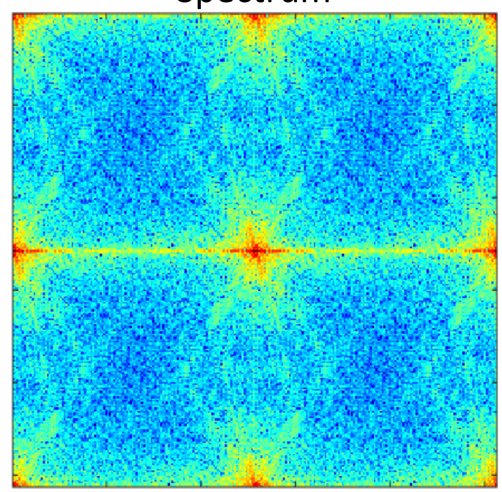
256



256

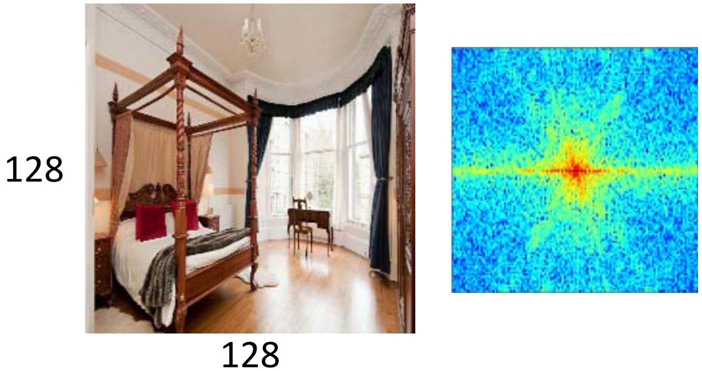
Spectrum

256

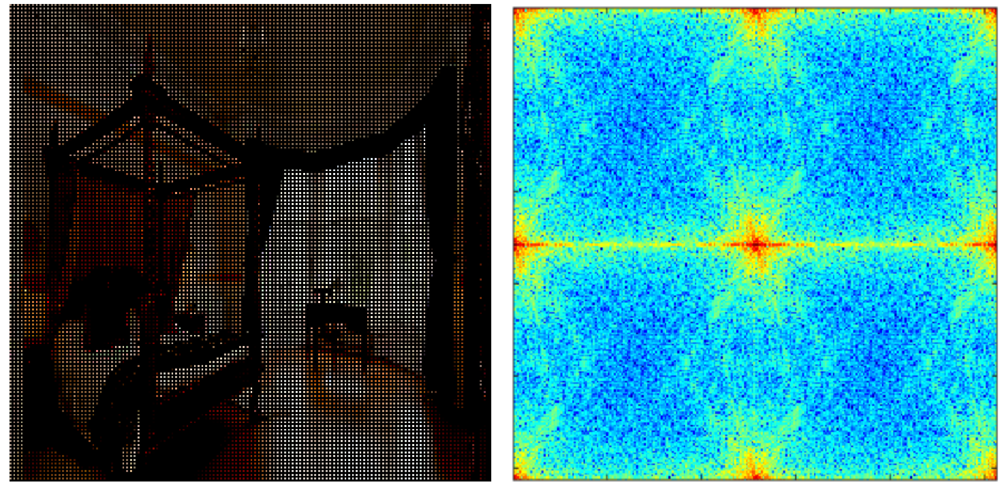


256

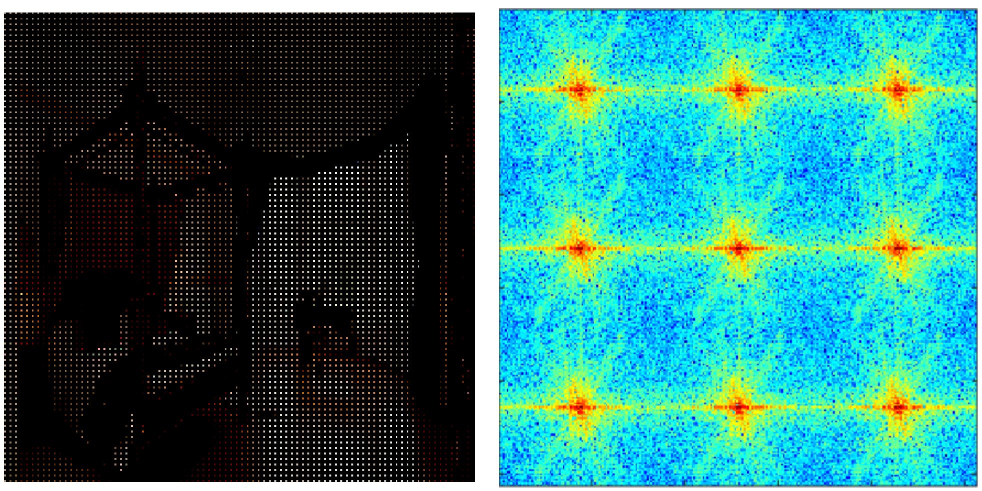
Low-resolution image



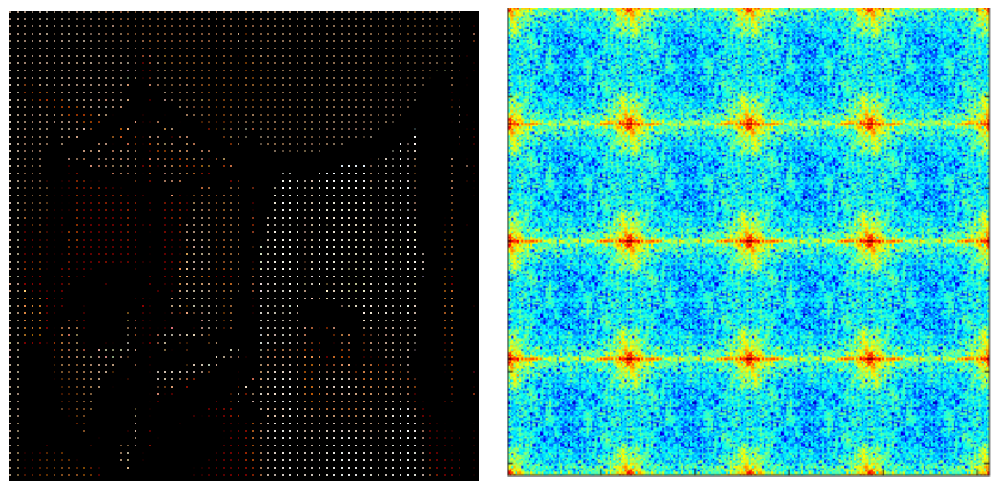
Stride 2



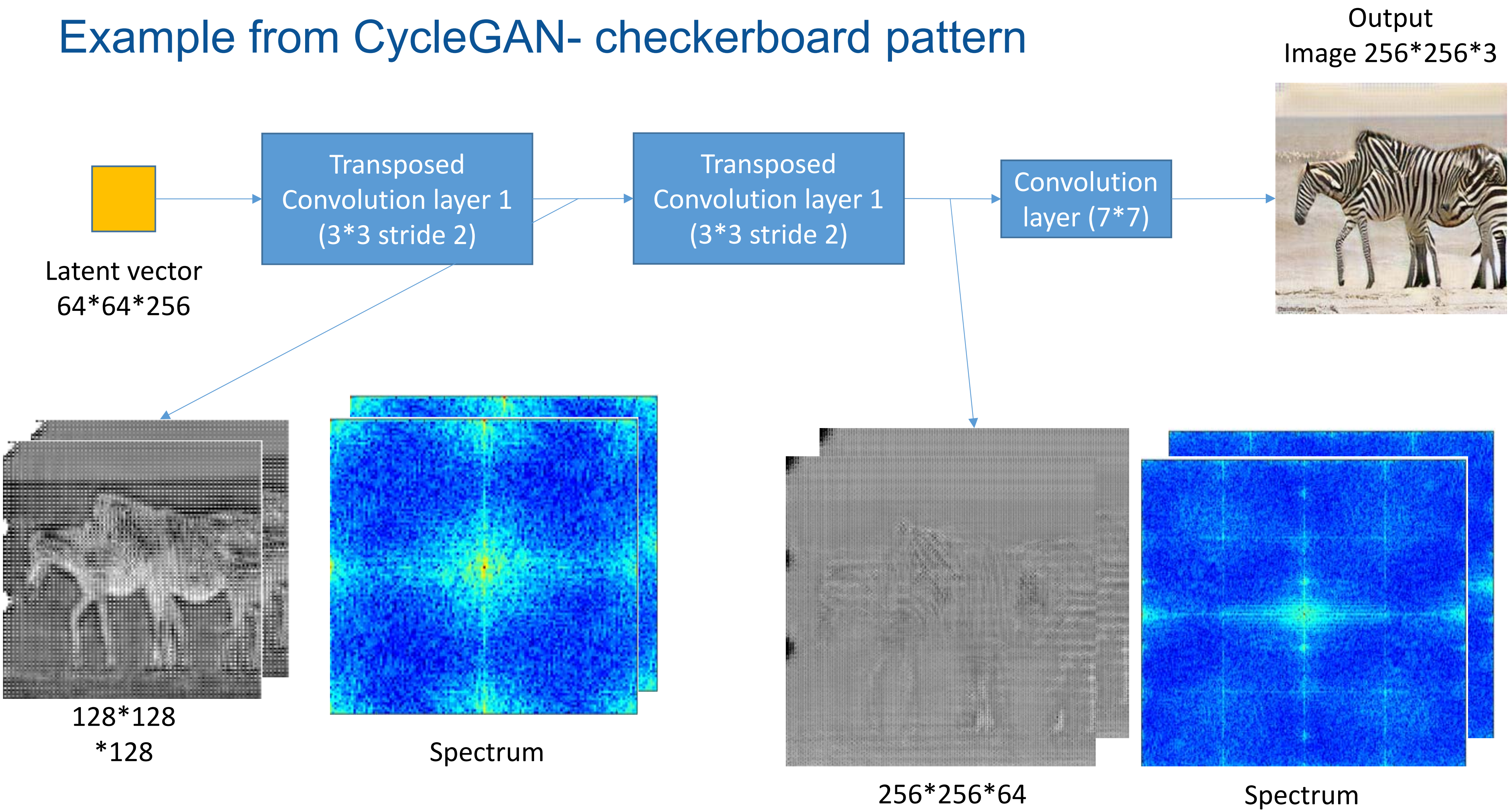
Stride 3



Stride 4



Example from CycleGAN- checkerboard pattern

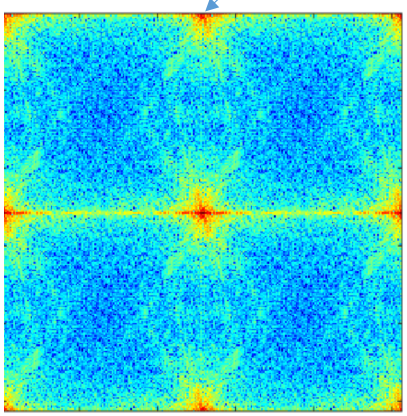


Effect of the Convolution Kernel

Low Resolution Image

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

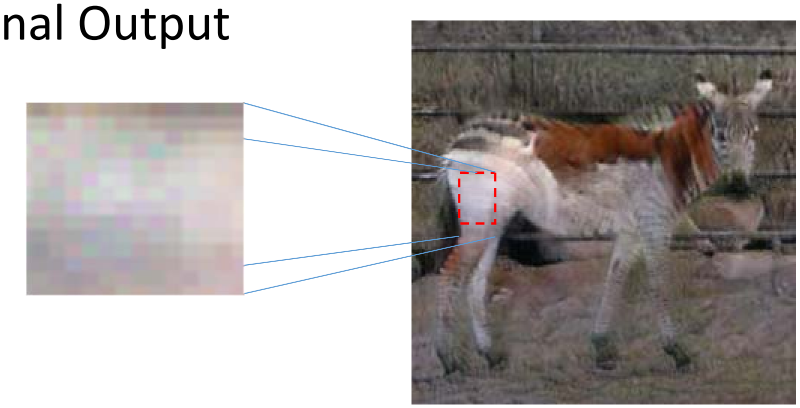
$$\underbrace{\begin{bmatrix} 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ 3 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\text{Zero Inserted Image}} \circledast \underbrace{\begin{bmatrix} ? & ? & ? \\ ? & ? & ? \\ ? & ? & ? \end{bmatrix}}_{\text{Learned Kernel}} = \underbrace{\begin{bmatrix} ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \\ ? & ? & ? & ? \end{bmatrix}}_{\text{High Resolution Image}}$$



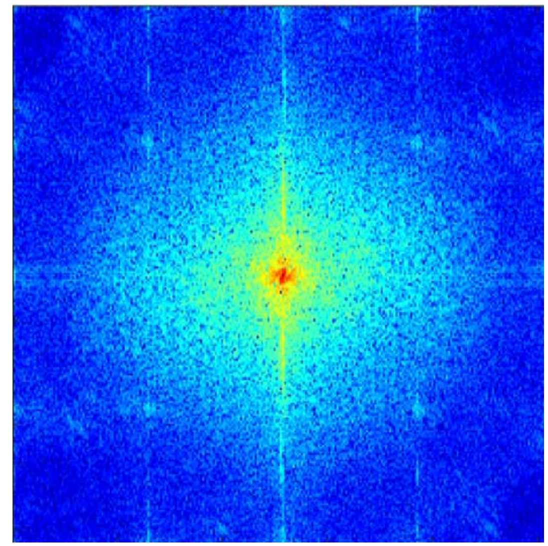
1. It has to be low-pass to remove the up-sampling artifacts
2. It can’t cut too much high frequency,
   otherwise the detail of the image will disappear.

Spectrum of the Fake Image

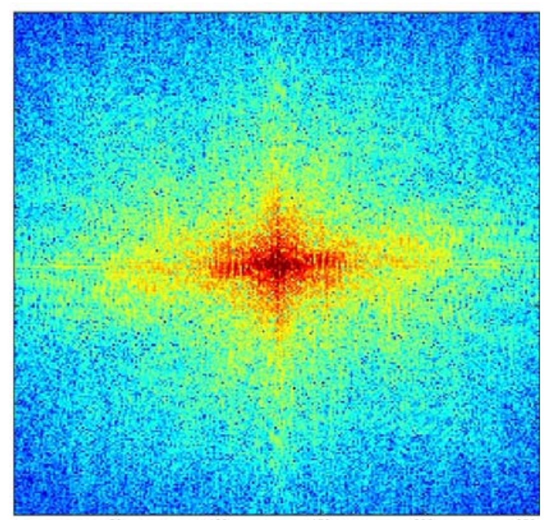
- Final Output



Fake



Real

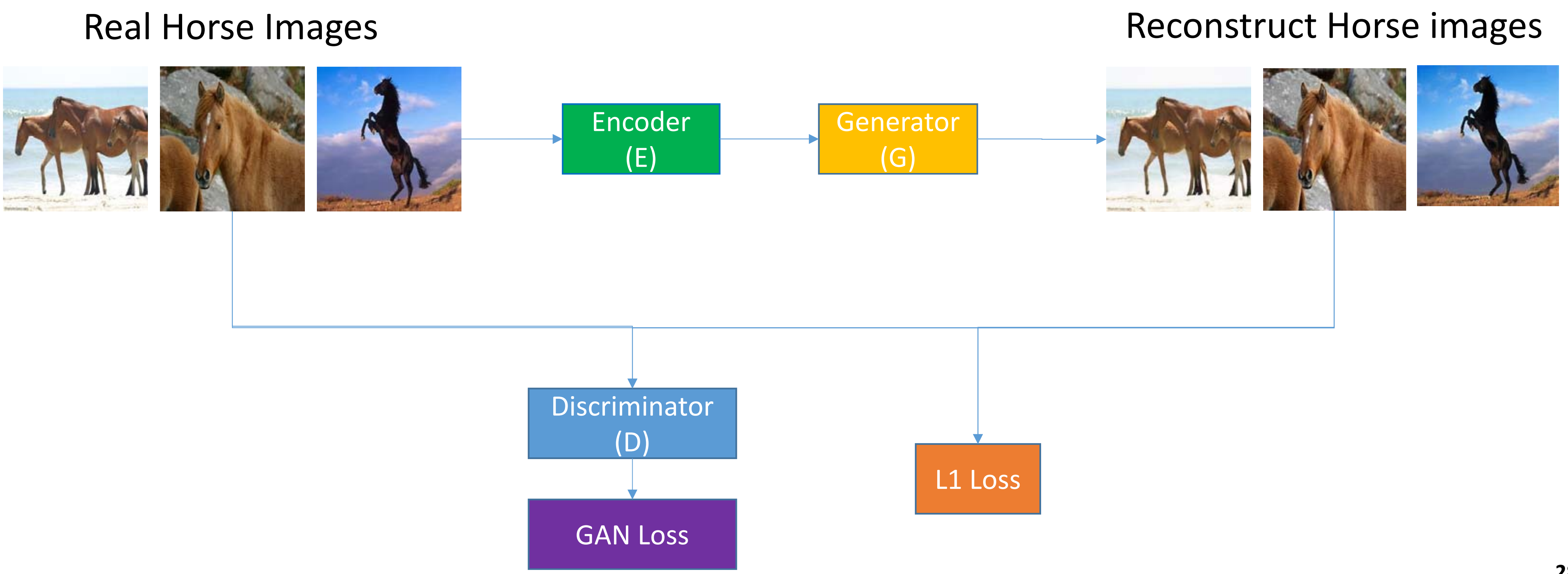


Goals:

- Are there "artifacts" induced in GAN image generation pipeline?
  - We explore a phenomenon and a theory related to **up-sampling artifact**.
- Are there ways to relax knowledge about the GAN models when training fake image classifier?
  - We propose a GAN pipeline emulator called AutoGAN.

AutoGAN – a GAN emulator for generating training samples

- Inspired by CycleGAN, we propose AutoGAN, which emulates the pipeline used in most GAN generation processes



AutoGAN

Real Horse Image

AutoGAN Reconstructed Horse image

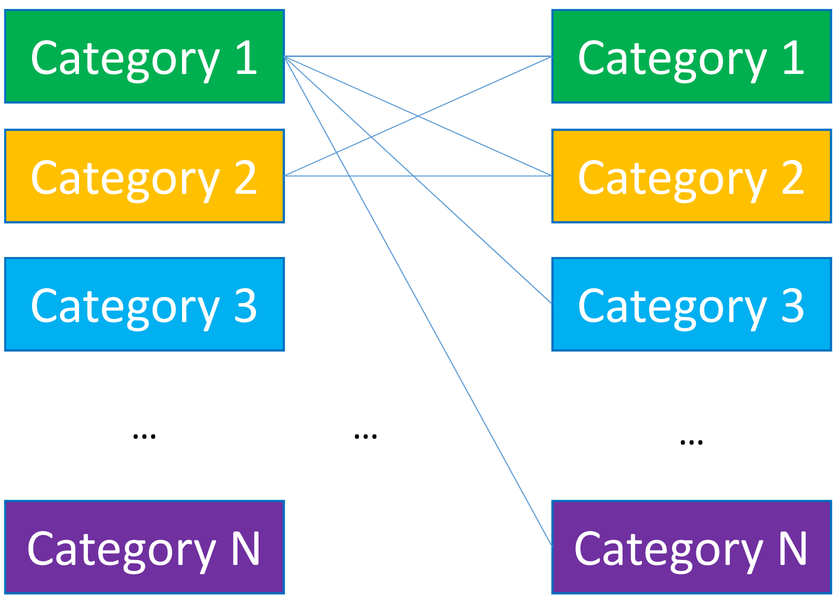
Benefits of GAN Pipeline Emulator

- High output image quality
- Different components can be easily incorporated (e.g., different up-samplers)
- Can be applied to any semantic class

Pairwise Training vs. AutoGAN

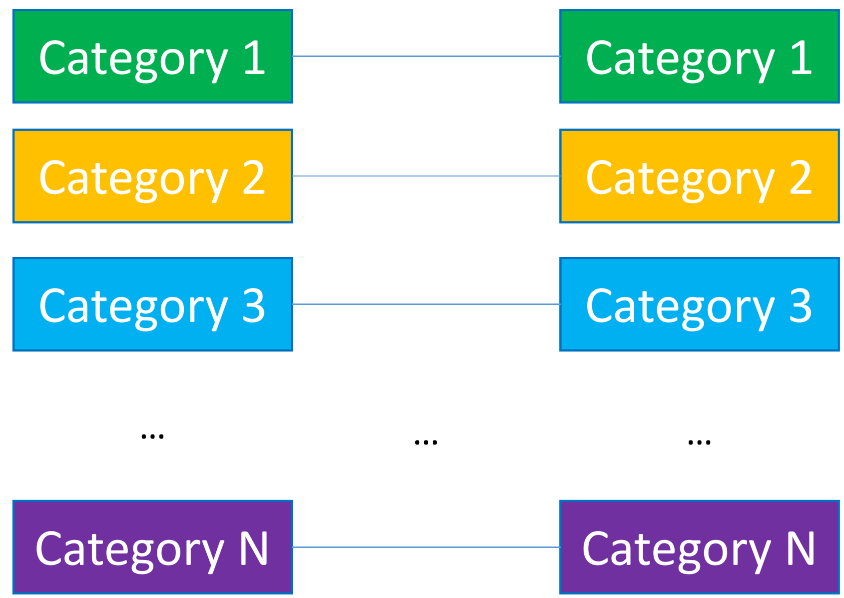
- AutoGAN does not need category transfer pairs and does not require access to the pre-trained model.

Pairwise Im2Im Transfer



Needs to consider all possible pairs (infeasible) or smartly chosen pairs

AutoGAN



Can be applied to any category

Leave One Out Performance

- Result
  - Leave one out performs pretty well, but need a huge number of training data from diverse sources.

Training	Test										
	Horse-Zebra	Apple-Orange	Summer-Winter	Facades	Cityscapes	Map	Ukiyo-e	Van Gogh	Cezanne	Monet	Avg.
image	95.2	90.0	97.9	76.4	85.1	92.7	98.0	91.6	96.3	97.0	92.0
Spectrum	99.6	99.4	99.7	100.0	100.0	50.0	100.0	98.0	100.0	99.4	**94.6**
Auto	92.7	67.4	98.4	94.8	50.6	51.8	68.7	97.1	57.4	92.5	77.1
Auto Spectrum	98.7	99.3	99.9	100.0	100.0	79.1	100.0	99.7	97.8	98.7	**97.3**

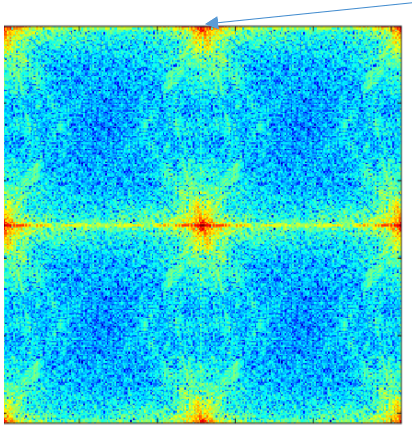
Trained with One Semantic Class Only

- Train with spectrum and AutoGAN works well for selective classes
- Conjecture: need classes that have sufficient spectrum coverage

Training	Test														
	Horse	Zebra	Summer	Winter	Apple	Orange	Facades	Cityscapes	Map	Ukiyo-e	Van Gogh	Cezanne	Monet	Photo	Avg
Horse Auto Spec	99.2	99.6	100.0	98.7	98.8	98.2	100.0	99.9	59.1	100.0	99.0	99.9	97.6	99.5	**96.4**
Zebra Auto Spec	61.9	91.5	94.0	54.3	61.7	53.1	100.0	86.0	51.5	94.3	52.8	91.8	51.1	97.1	74.4
Summer Auto Spec	96.9	96.9	99.8	95.6	95.1	95.5	100.0	99.6	50.1	99.4	99.1	99.6	97.5	99.0	**94.6**
Winter Auto Spec	47.3	69.2	94.1	82.1	58.8	57.2	53.8	52.1	50.8	99.2	95.0	84.7	90.9	95.4	73.6
COCO Auto Spec	93.8	88.5	85.9	85.6	89.9	92.0	100.0	100.0	89.7	97.2	74.0	98.0	83.3	79.7	**89.8**

Does It Work for Different Up-sample Modules?

- Nearest neighbor interpolation
  - Widely used nowadays for up-sampling (Prog-GAN, GauGAN)
  - Can be viewed as zero inserting + low pass filter
  - Suffers less from checkerboard patterns [Odena, et al., 2016].

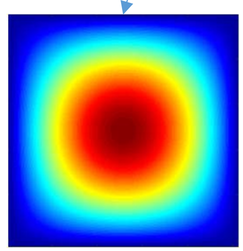


1	0	2	0
0	0	0	0
3	0	4	0
0	0	0	0

Zero Inserted Image

$\circledast$

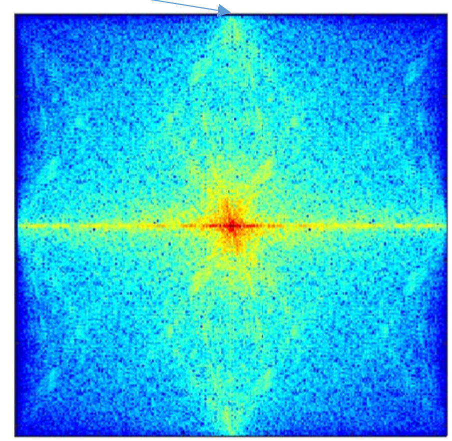
1	1	0
1	1	0
0	0	0



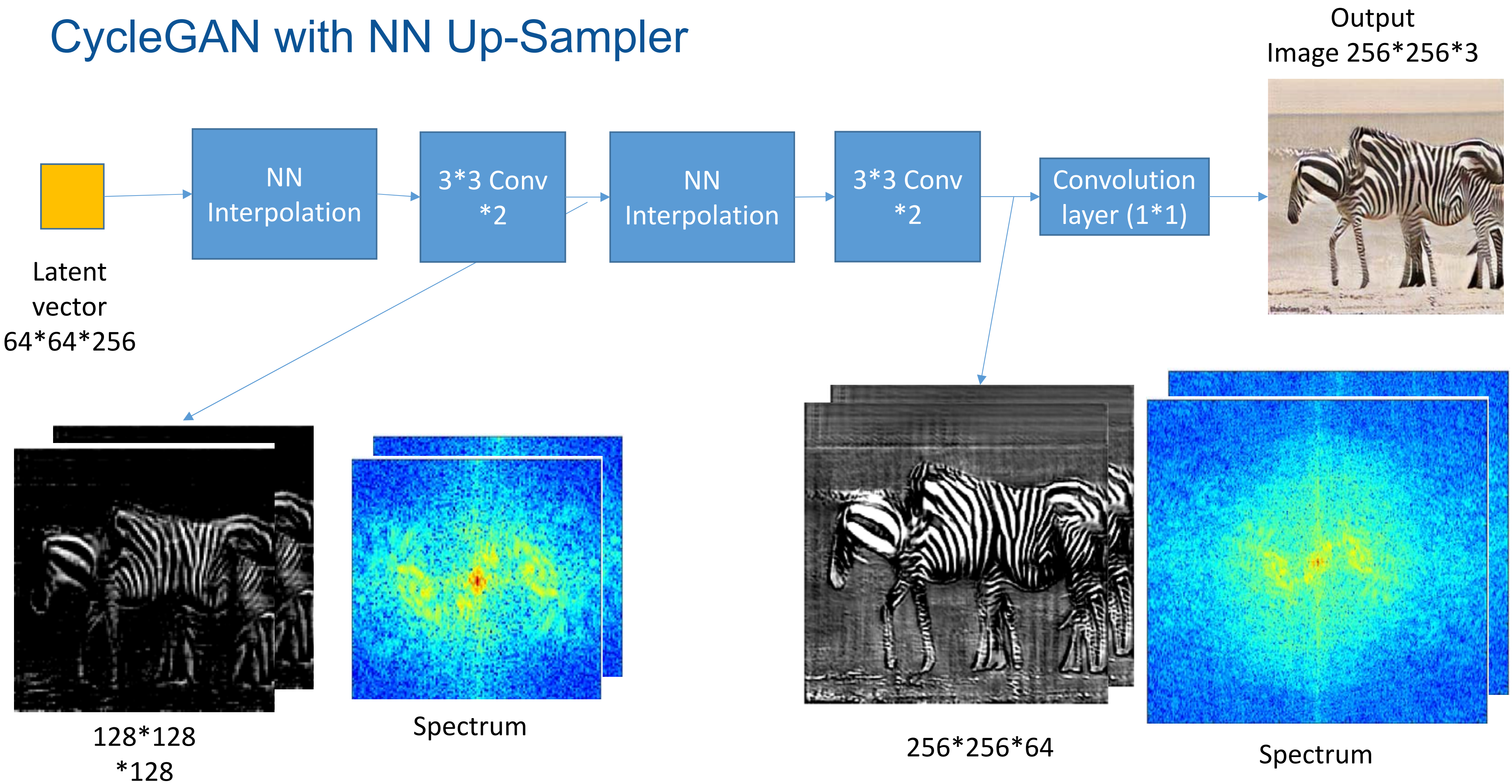
=

Interpolated Image

1	1	2	2
1	1	2	2
3	3	4	4
3	3	4	4



CycleGAN with NN Up-Sampler



Up-sample Module Comparison

- Nearest neighbor interpolation causes less checkerboard effect

Real Horse Image

Reconstructed Zebra image
with TRANS convolution

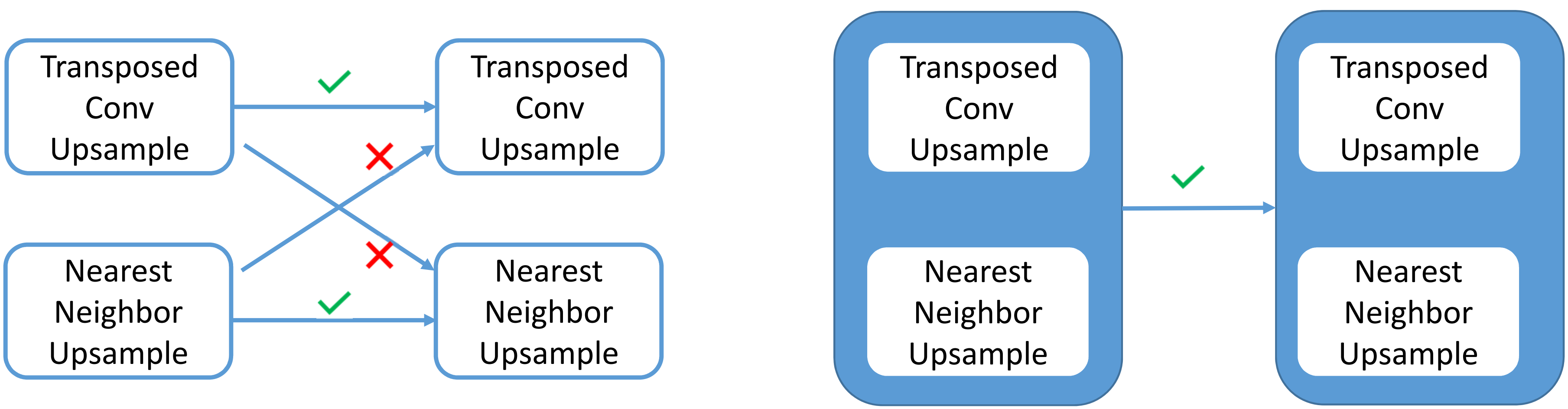
Reconstructed Zebra image
with NN interpolation

Train with NN up-sampler and Test with NN up-sampler, One Class

- Spectrum based models still work well for NN up-sample, even if trained on one class only

Training	Test						
	Horse NN	Zebra NN	Summer NN	Winter NN	Apple NN	Orange NN	Avg
Horse_NN_Spec	99.6	96.9	86.7	97.1	96.1	93.2	94.9
Zebra_NN_Spec	100.0	99.6	96.5	99.3	92.2	90.9	96.4
Summer_NN_Spec	96.2	91.2	99.6	99.8	87.2	85.0	93.2
Winter_NN_Spec	96.2	96.5	100.0	100.0	93.4	90.3	96.1

Generalization: GANs of different upsamplers



Generalization across different models

- [Nataraj et. al 2019] showed model trained with CycleGAN works well for StarGAN
- StarGAN and CycleGAN share the similar generator structure
- But model learned with cycleGAN (2 up-sampling modules) does not generalize well to GauGAN (5 up-sampling modules)

Test	Method Train with CycleGAN			
	Image	Spectrum	Auto	Auto_Spectrum
StarGAN	65.06	100.0	92.49	98.68

Conclusions

- Typical up-sampling modules in GAN leave **up-sampling artifacts** in the generated images.

- **Spectrum-based detectors** seem to be able to reveal the artifacts
    - Training with spectrum input generalizes well even if trained with one calss only.

- We also propose **GAN pipeline emulator AutoGAN**, while emulates the up-sampling artifacts in GAN generated image.
    - Relax knowledge about GAN model
    - Does not need access to the GAN model or generated images

Conclusions

- Model trained with one up-sampling module does not generalize well to different up-sampling modules
    - But models trained with multiple modules work

- Model learned with similar up-sampling architectures works (CycleGAN vs. StarGAN), but not distinct models (e.g., GauGAN)