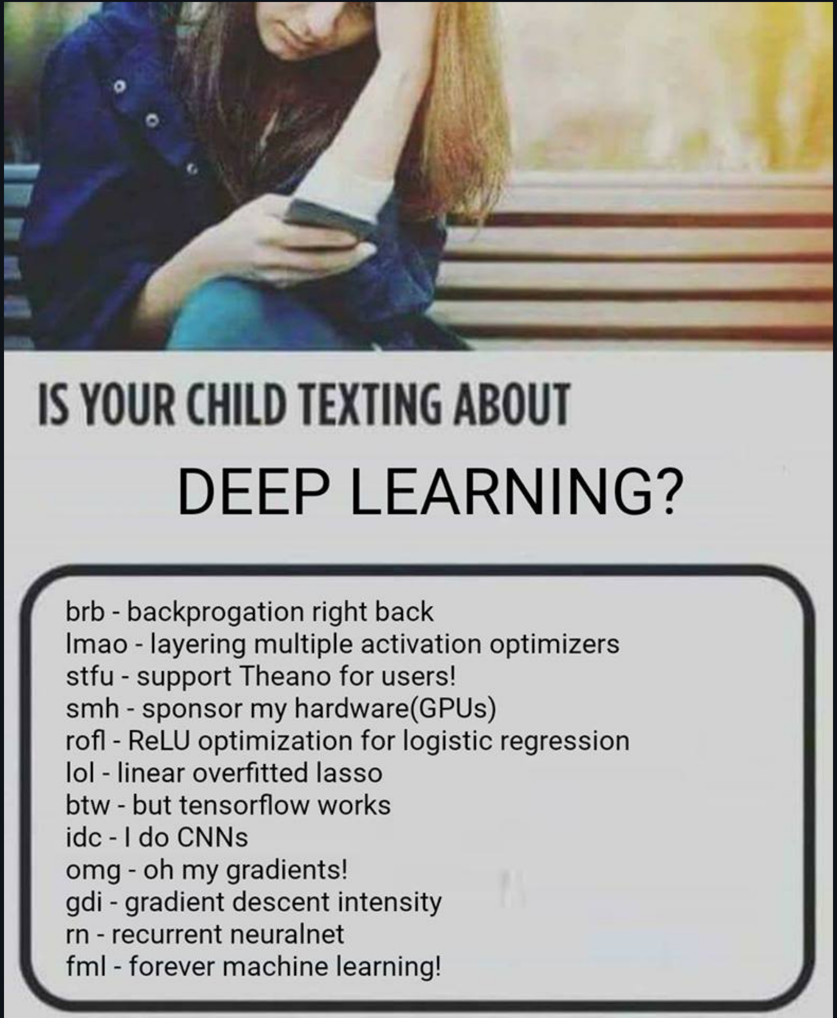


Image Source:
facebook.com/convolutionalmemes

Building & Training an Image-to-Image Generative Model

Felipe Ducau (Sophos / Paperspace)
Dillon Erb (Paperspace)
Lindy marcel (Paperspace)

SOPHOS

Paperspace

Agenda

1. Introduction
2. Image to Image translation - Problem Definition
3. Generative Adversarial Networks (GANs)
4. Deep Convolutional GAN (DCGAN)
5. Conditional GAN (cGAN)
6. Results demo and discussion
7. Closing remarks & further possible applications

Image to Image Translation - How does it look like?

1)

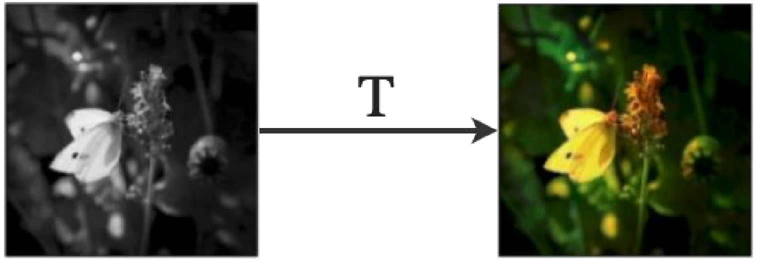
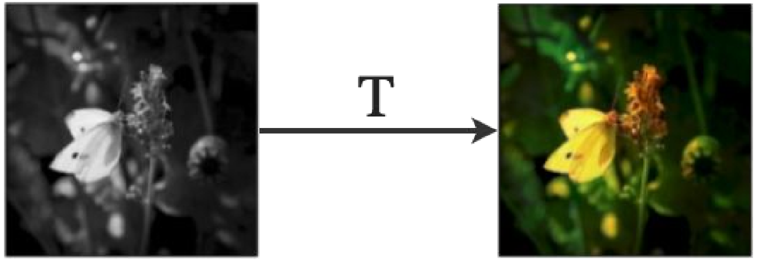


Image credits: Isola, et al. 2016

Image to Image Translation - How does it look like?

1)



2)

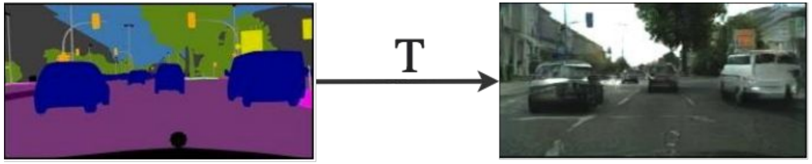
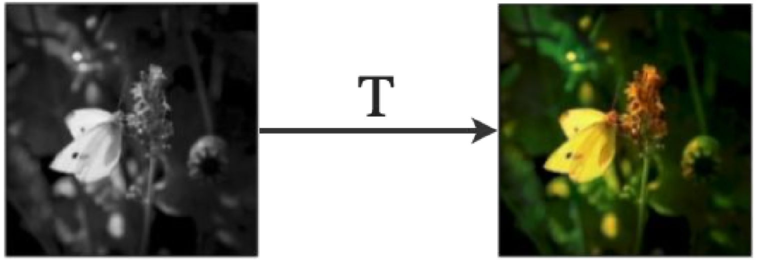


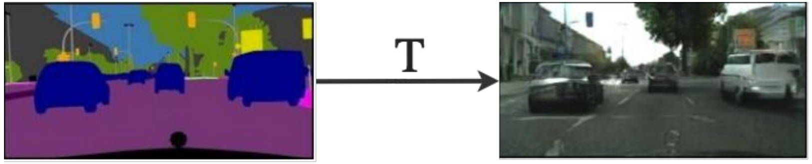
Image credits: Isola, et al. 2016

Image to Image Translation - How does it look like?

1)



2)



3)

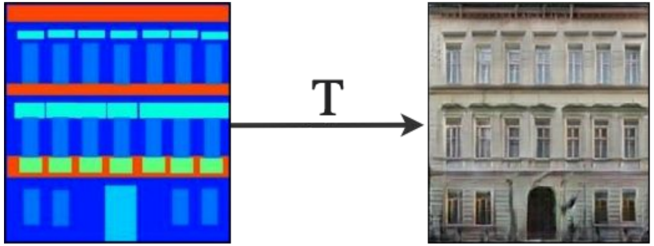
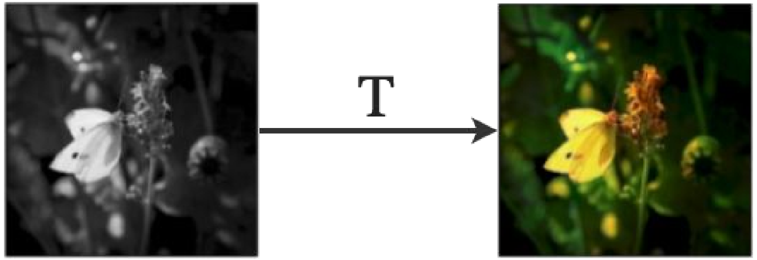


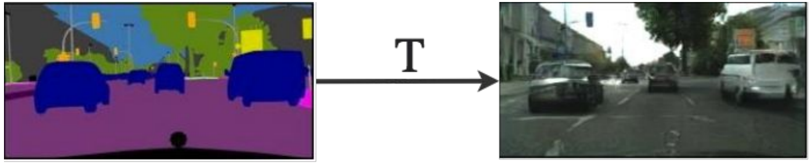
Image credits: Isola, et al. 2016

Image to Image Translation - How does it look like?

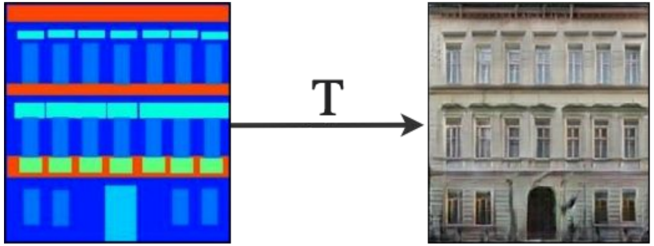
1)



2)



3)



4)

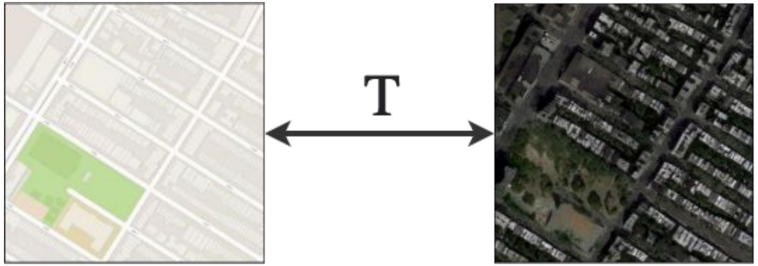
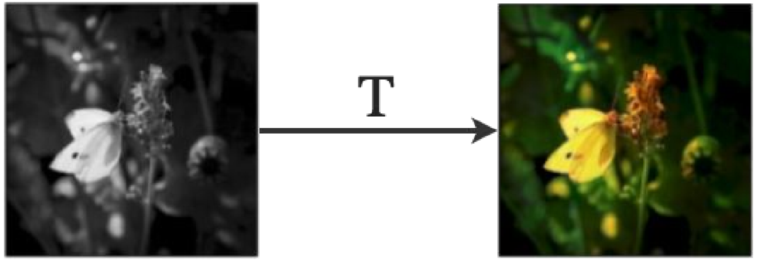


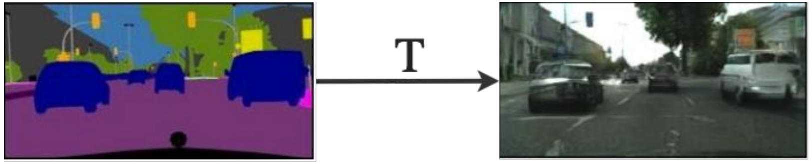
Image credits: Isola, et al. 2016

Examples of Image to Image Translation

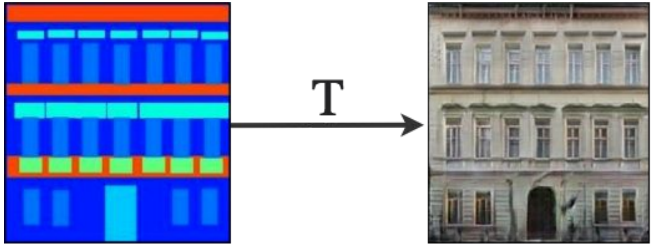
1)



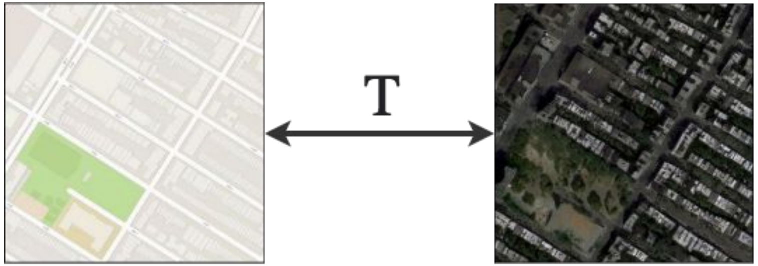
2)



3)



4)



5)

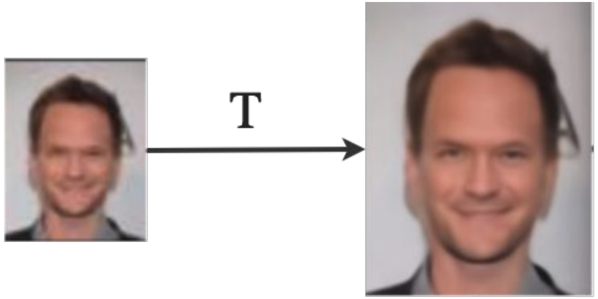
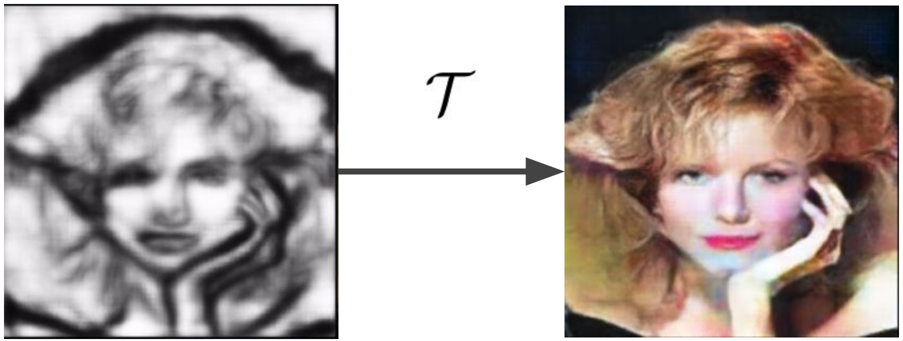


Image credits: Isola, et al. 2016

Image to Image Translation

Find a **translation** between one possible representation of a scene into another representation given sufficient data.

Ill posed task: the solution (translation) is underdetermined, i.e. multiple possible outputs for a given input.



Input distribution
Image edges

Output distribution
Real Image

Image to Image Translation

Naive Approach

Train a Deep Neural Network **G** to learn the translation from input to output distributions using a parallel corpus and L_1 or L_2 loss.

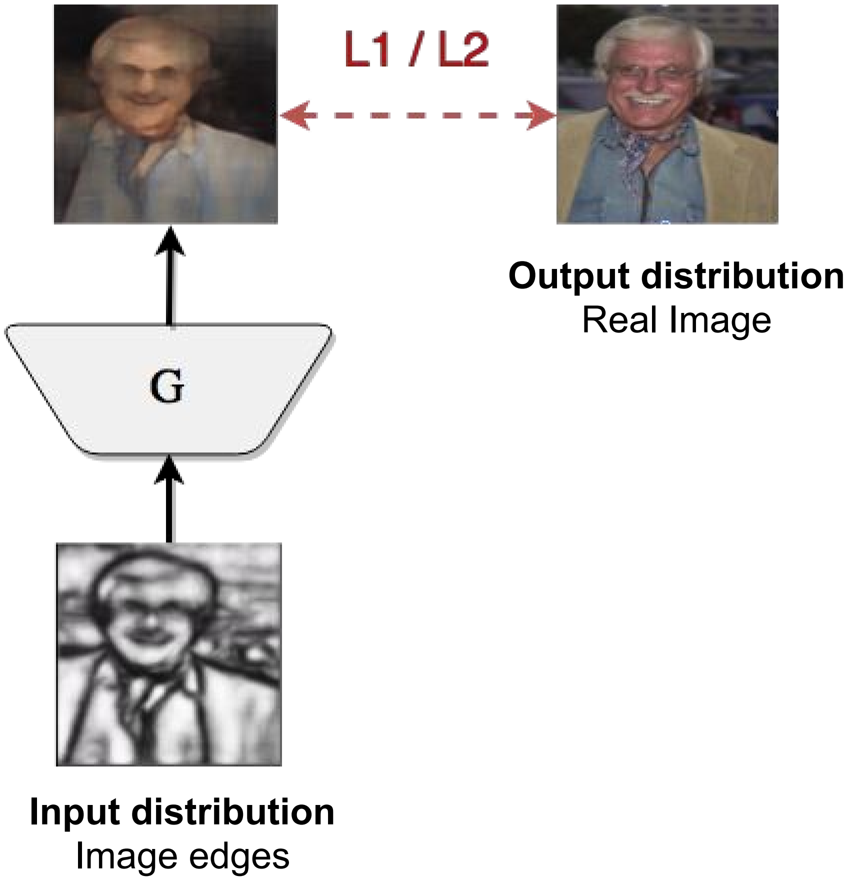


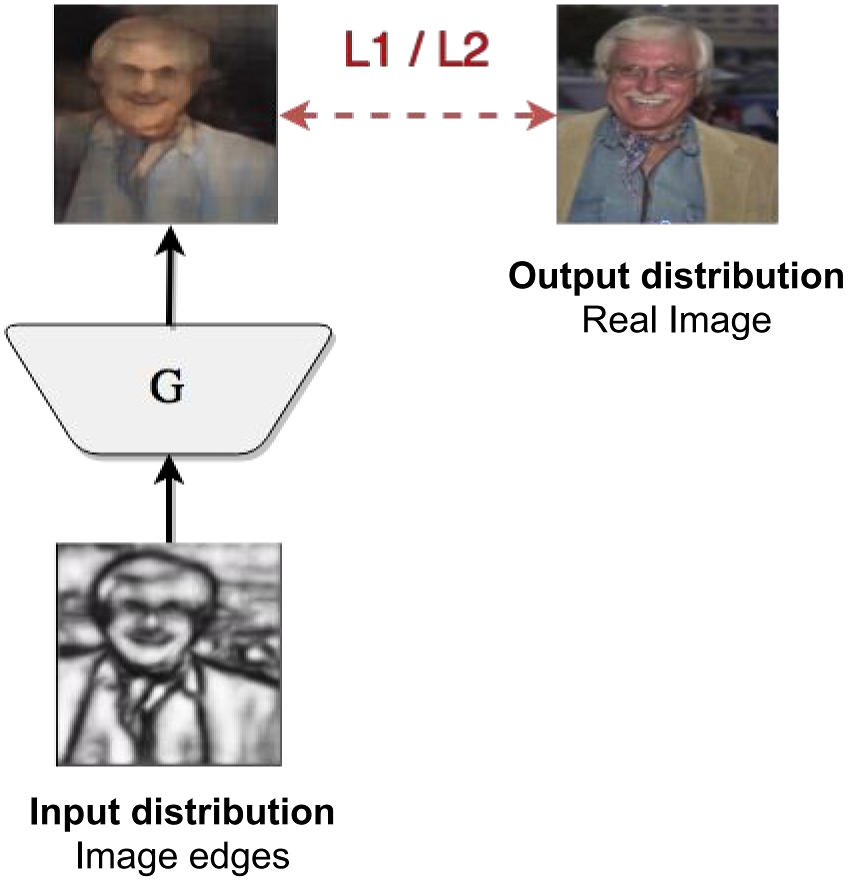
Image to Image Translation

Naive Approach

Train a Deep Neural Network **G** to learn the translation from input to output distributions using a parallel corpus and L_1 or L_2 loss.

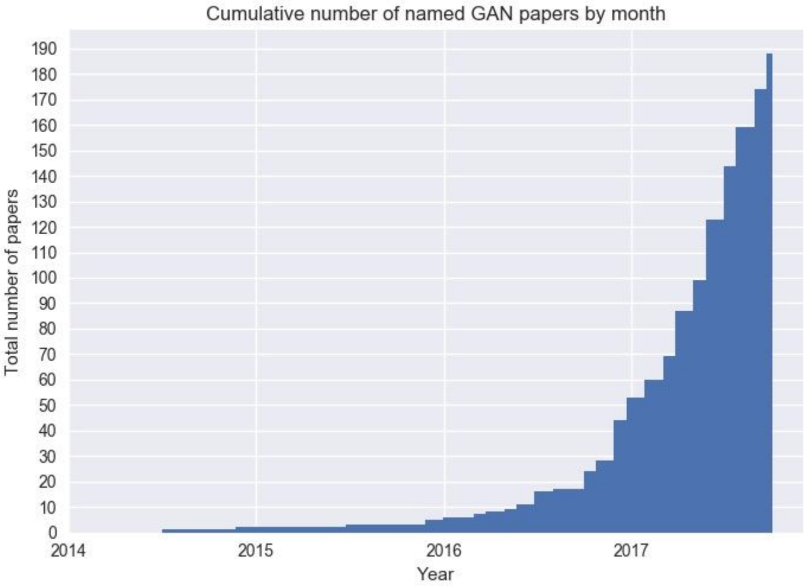
Problem - Multiple Possible Outputs

Euclidean distance is minimized by averaging all plausible outputs, which causes blurring.



Generative Adversarial Networks (Goodfellow et al., 2014) (GANs)

Generative model that allow us to generate more realistic images.



"Adversarial training is the coolest thing since sliced bread."

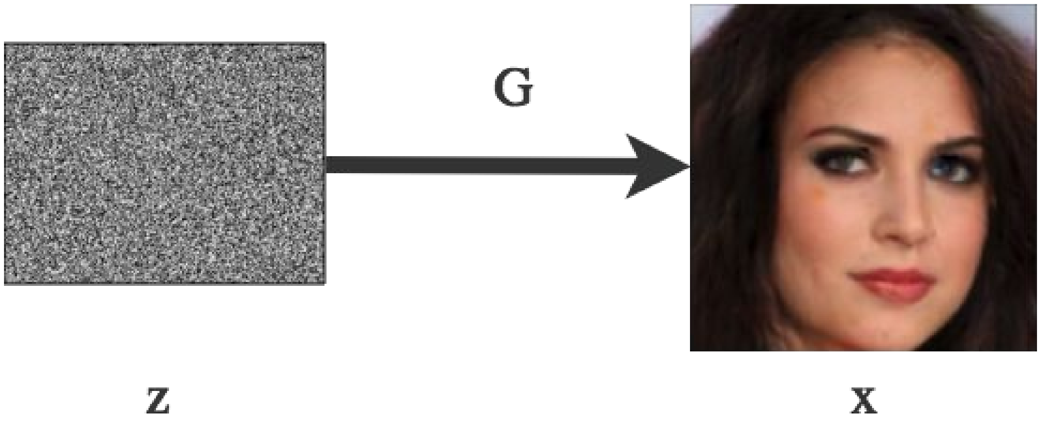
-Yann LeCun

github.com/hindupuravinash/the-gan-zoo

GANs

Generative model that learns a mapping from a noise vector **z** to an output image **x**.

$G : \mathbf{z} \rightarrow \mathbf{x}$



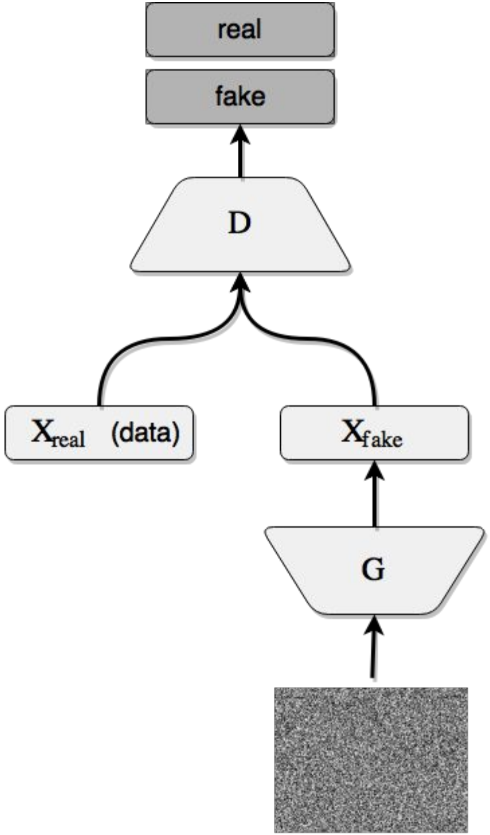
GANs

Generator (G)

trained to produce outputs that cannot be distinguished from “real” images.

Discriminator (D)

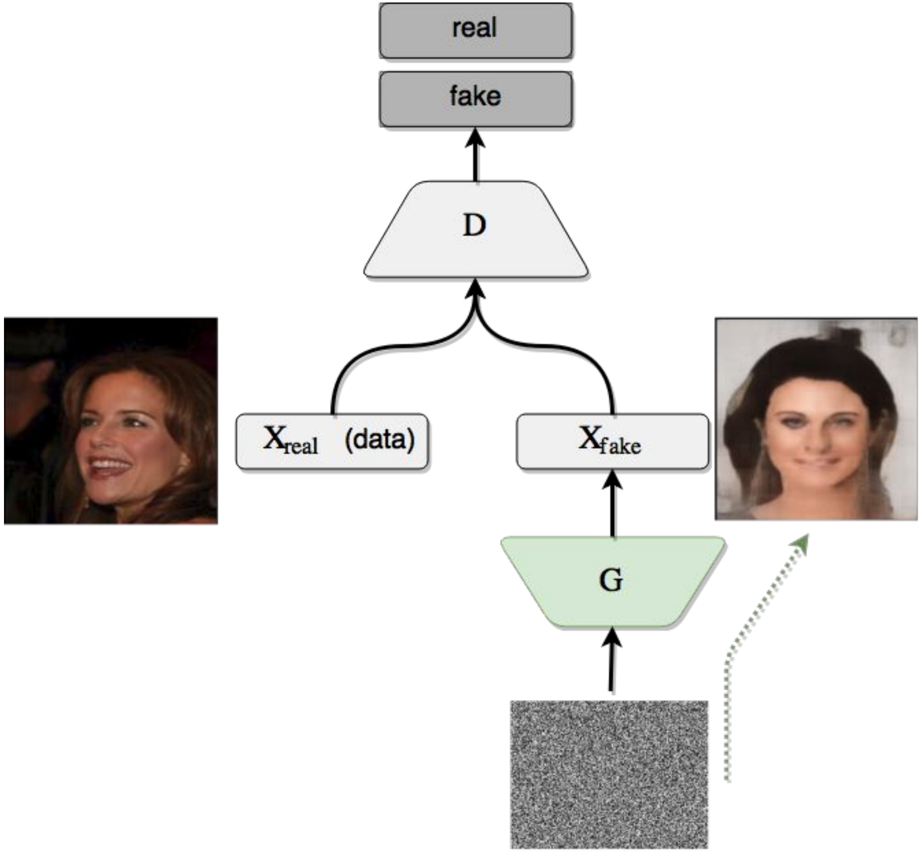
trained to differentiate between “fake” images produced by the generator and “real” images



GANs

Forward path - Generator

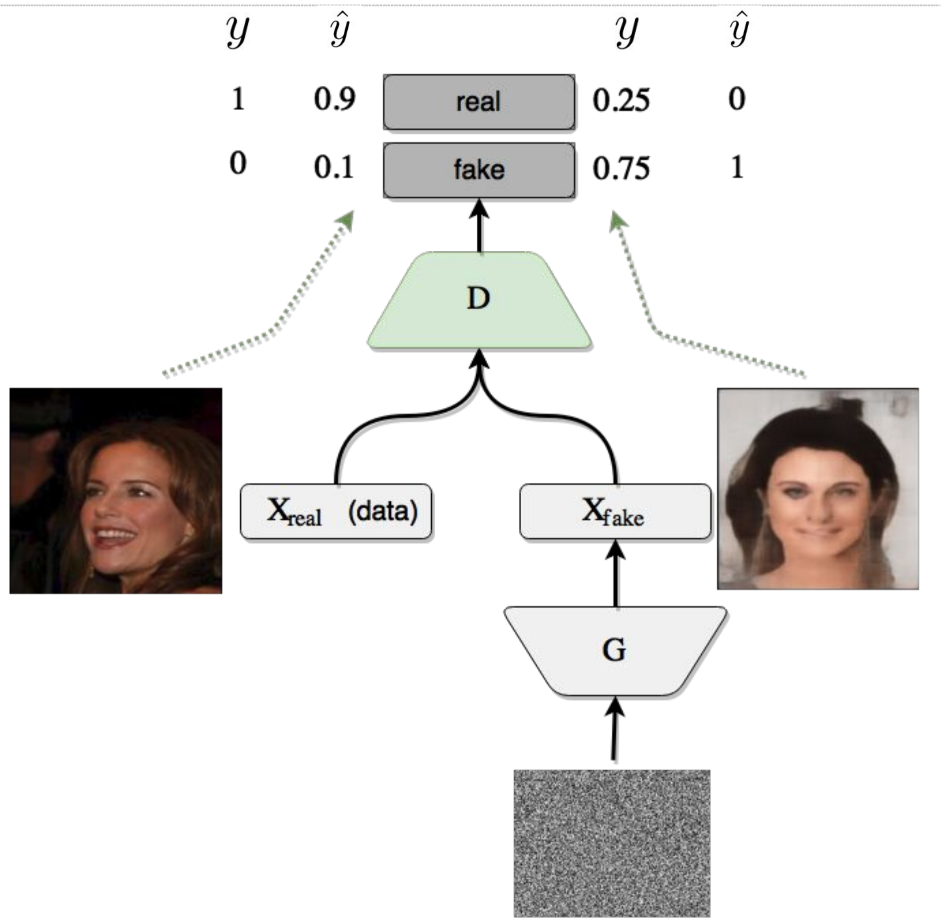
$$z \sim p(z)$$
$$x_{fake} = G(z)$$
$$x_{real} \sim p_{data}(x)$$



GANs

Forward path - Discriminator

$$P(x_{fake} \sim p_{data}(x)) = D(x_{fake})$$
$$P(x_{real} \sim p_{data}(x)) = D(x_{real})$$



GANs

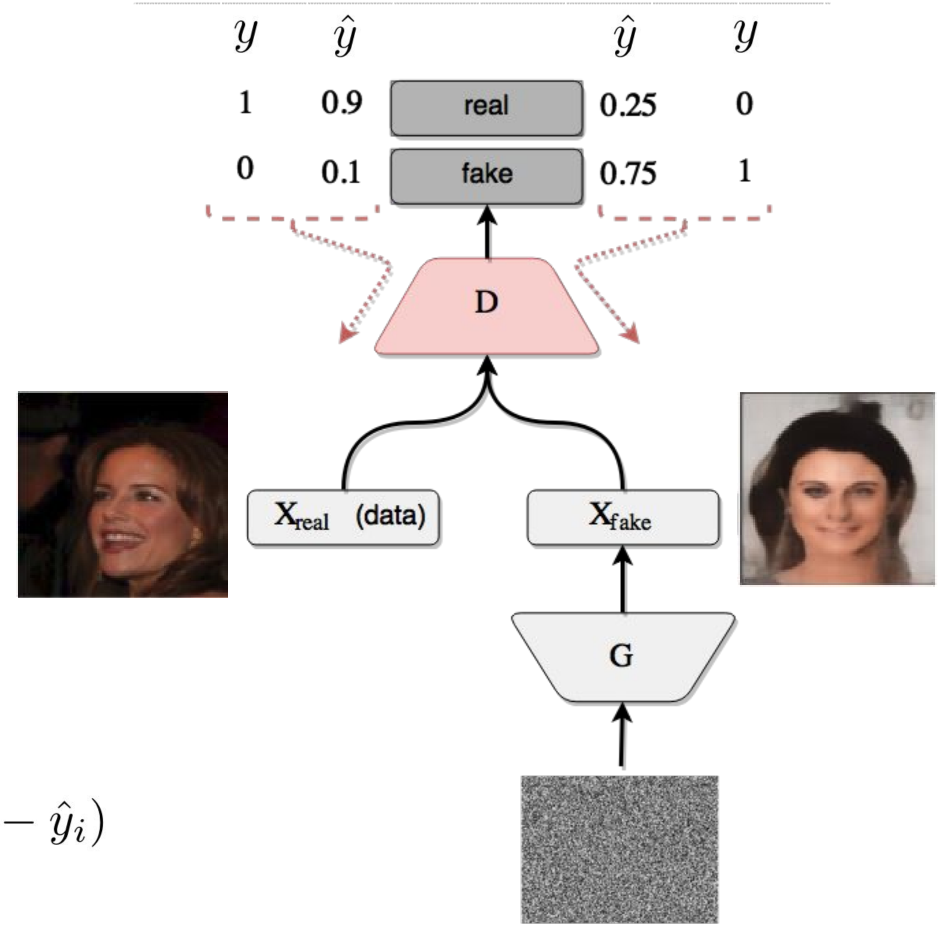
Backward path - Discriminator loss

True Label: y

Prediction: $\hat{y} = D(x)$

$$\mathcal{L}_D = \text{BinaryCrossEntropy}(\hat{y}, y)$$

$$= -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)$$



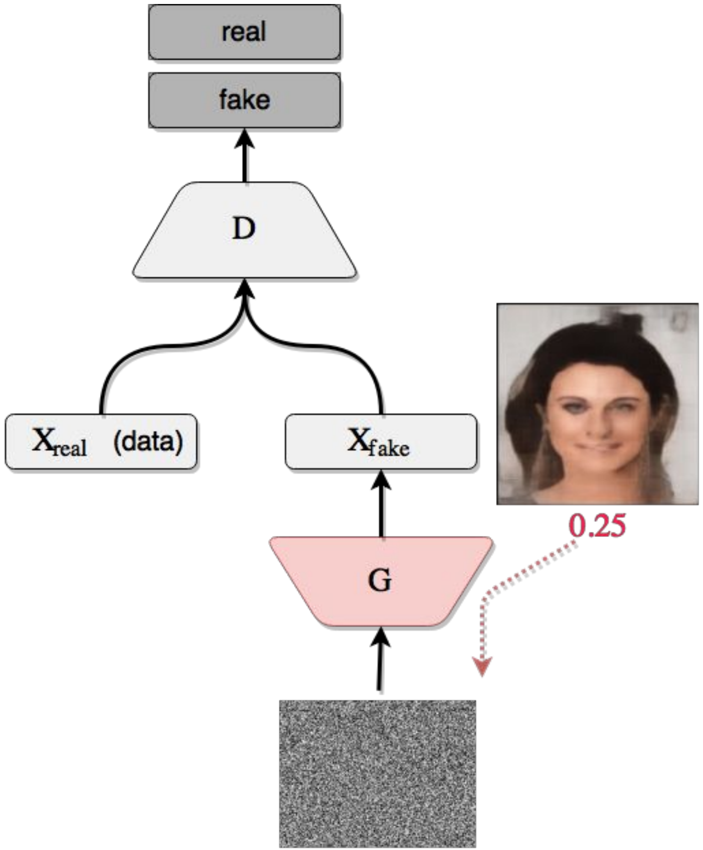
GANs

Backward path - Generator Loss

$$\text{Error}_{G_{Adv}} = (1 - D(x_{fake}))$$

$$\mathcal{L}_{G_{Adv}} = -\frac{1}{N}\sum_{i=1}^{N}\log(1 - D(x_{fake_i}))$$

$$= -\frac{1}{N}\sum_{i=1}^{N}\log(1 - D(G(z_i)))$$



GANs - General Formulation

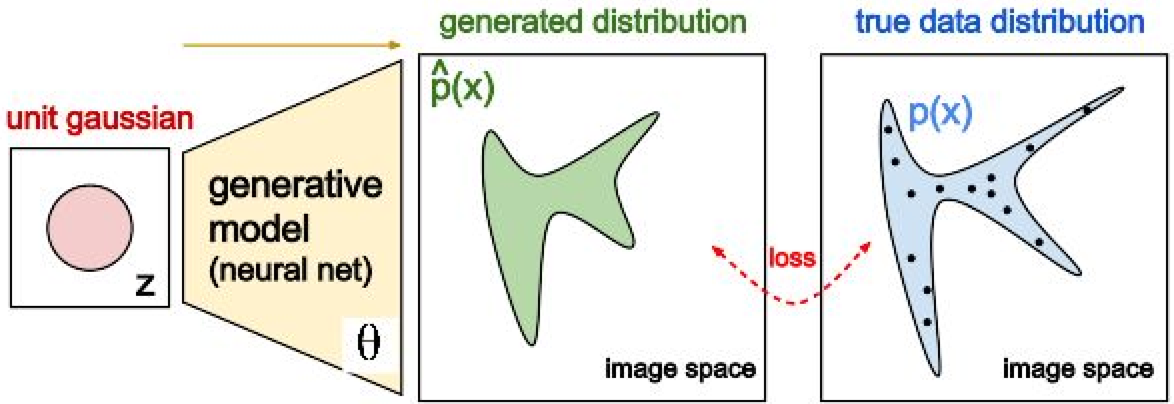
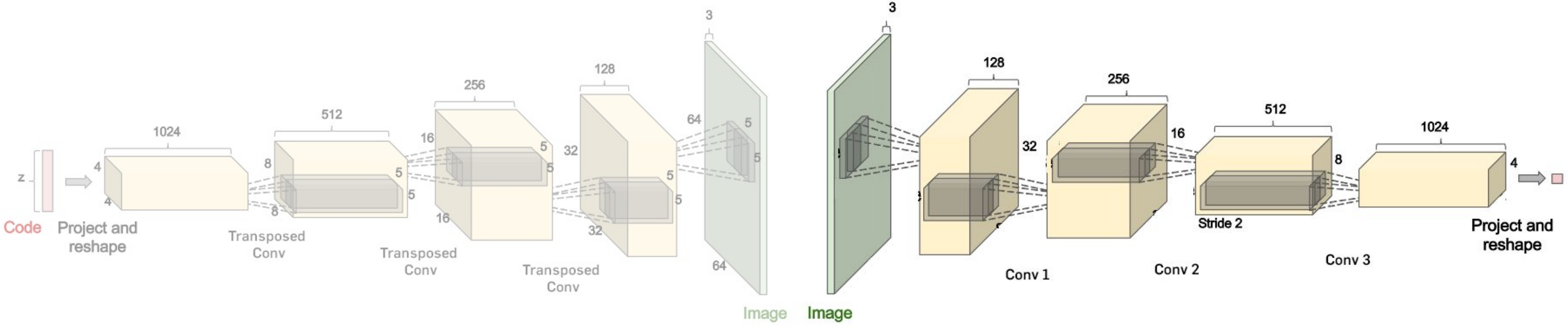


Image source: blog.openai.com/generative-models

DCGAN (Radford et al. 2016)

Generator

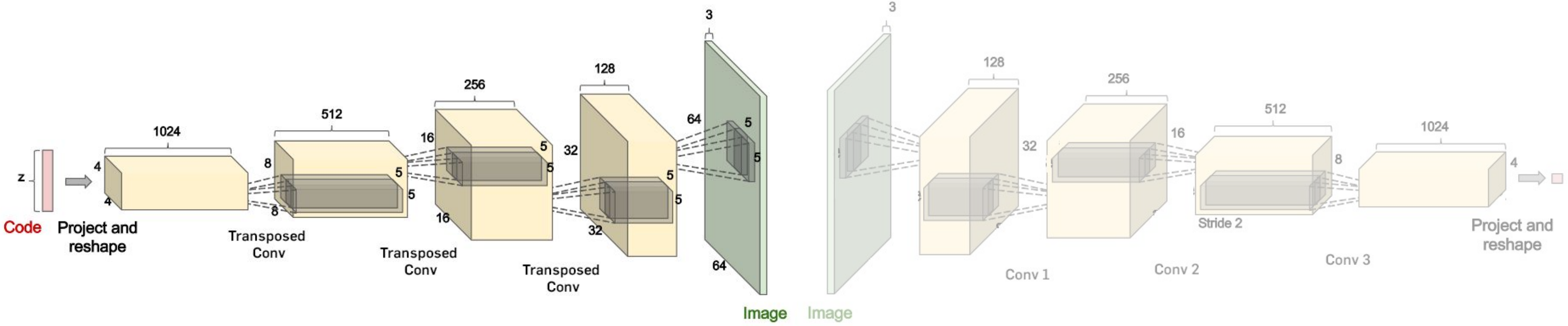
Discriminator



DCGAN (Radford et al. 2016)

Generator

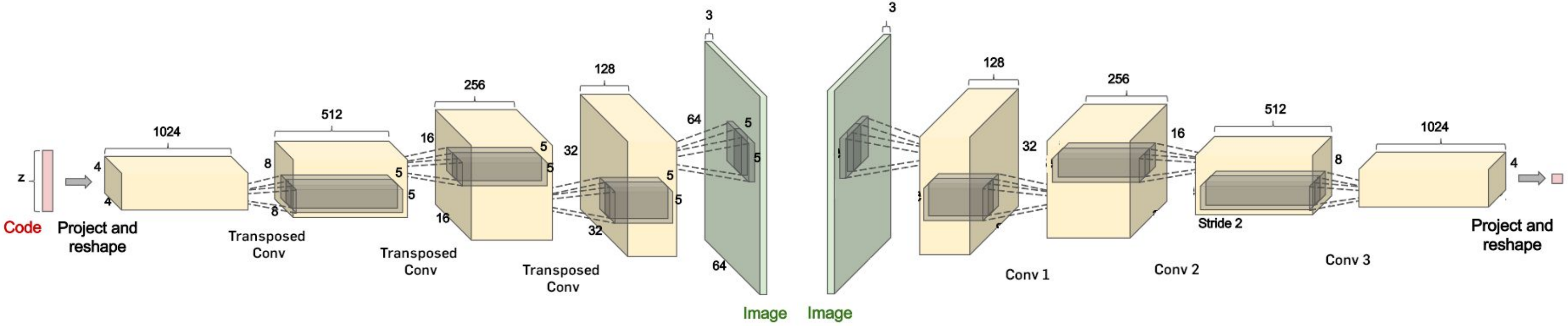
Discriminator



DCGAN (Radford et al. 2016)

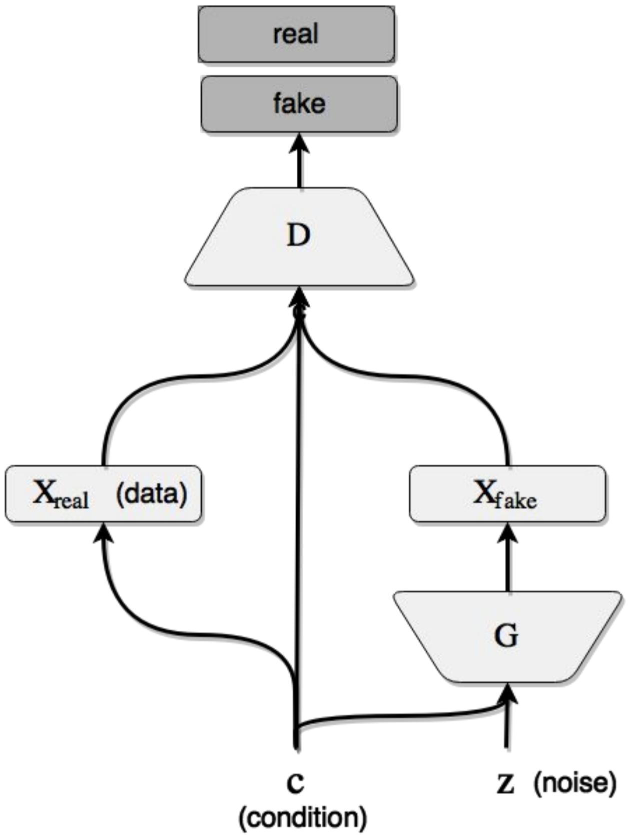
Generator

Discriminator



Conditional GAN (cGAN)

G : {**c**, **z**} ➜ **x**



Conditional GAN (cGAN)

G : {**c**, **z**} ➔ **x**

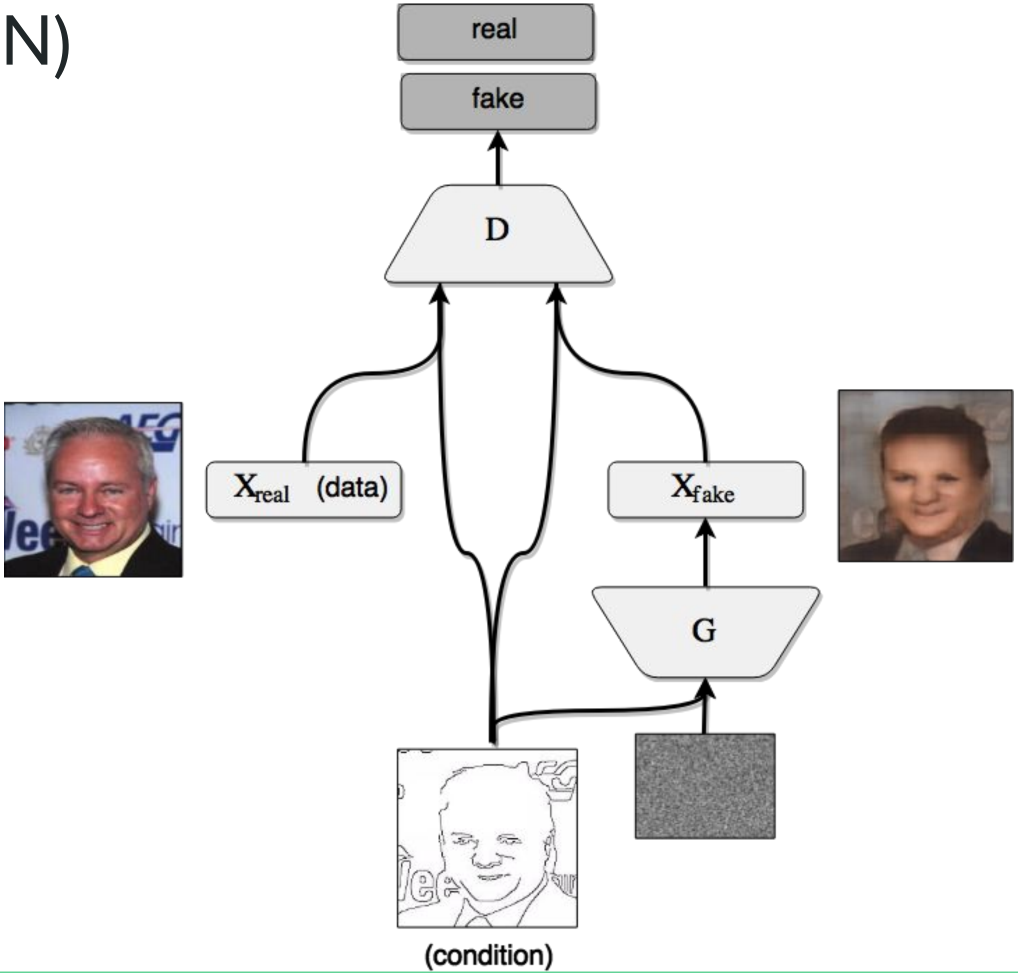


Image2Image with cGANs

Generator

U-Net architecture

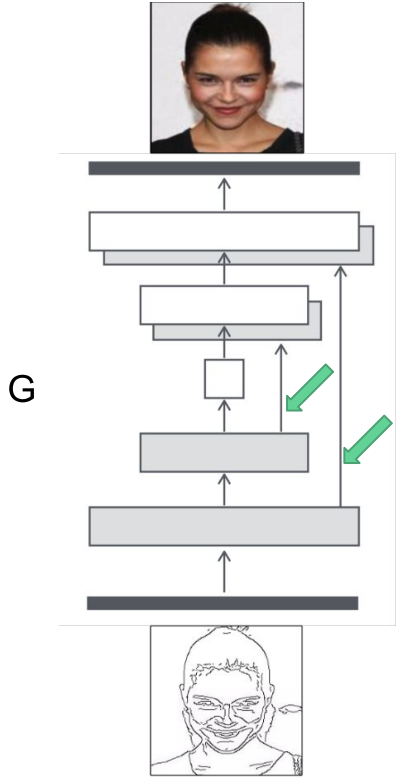


Image2Image with cGANs

Discriminator

PatchGAN

(Markovian Discriminator)

The patch size becomes an hyperparameter of the model.

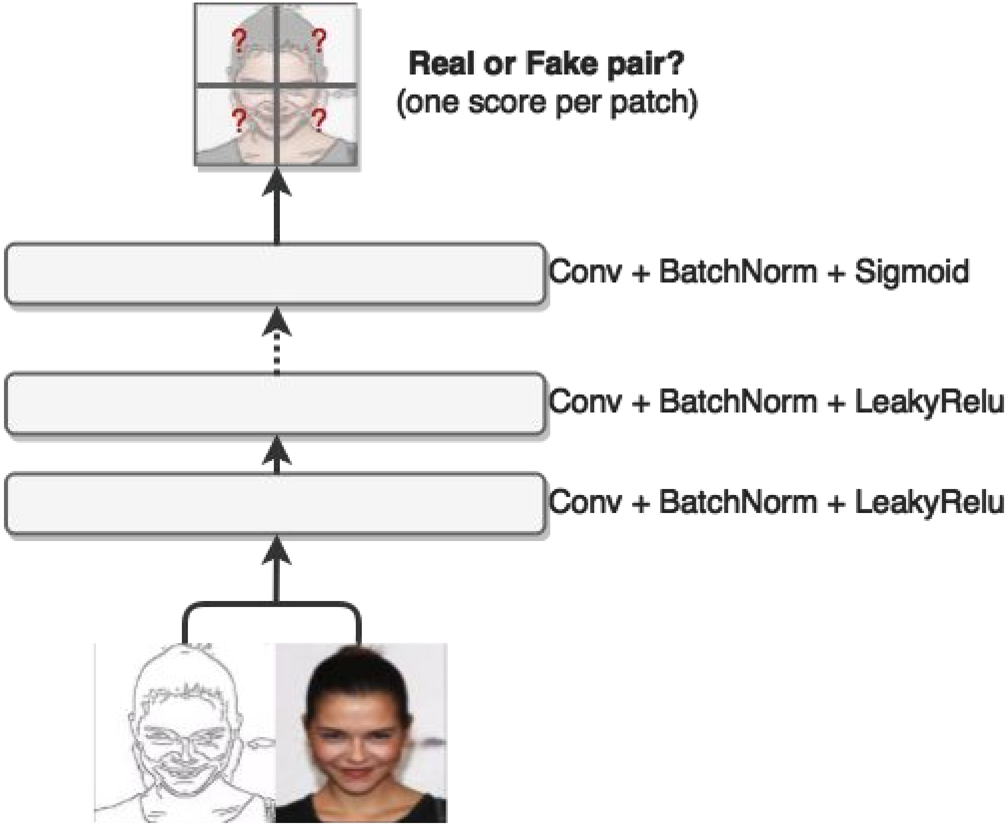


Image-to-Image with cGANs

Architecture

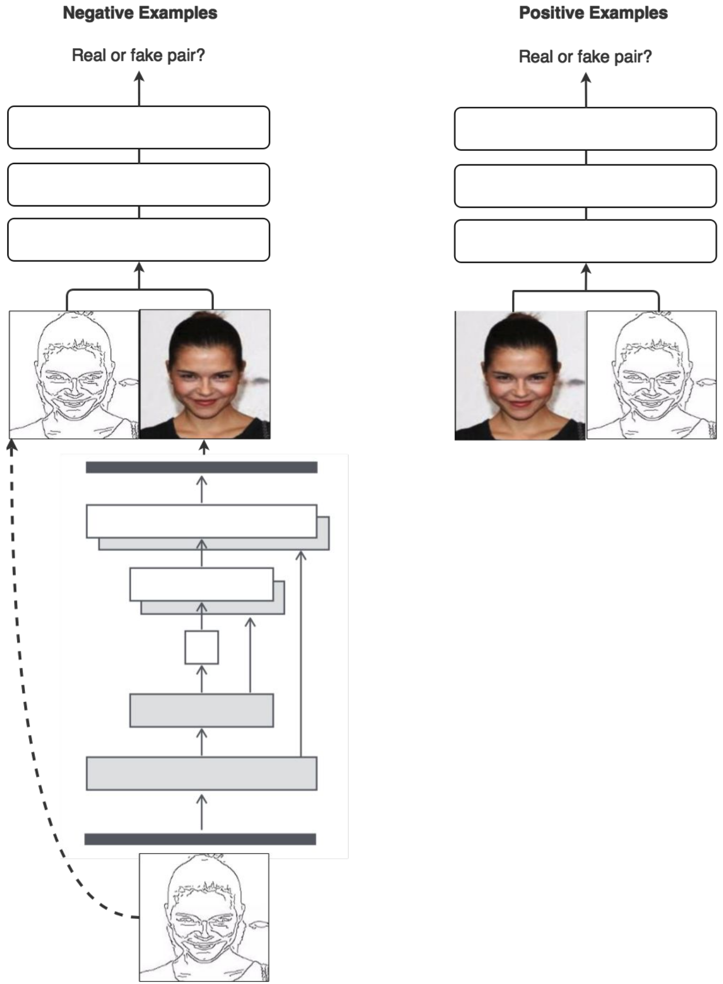


Image2Image with cGANs

Generator loss:

$$\mathcal{L}_G = ||x_{fake} - x_{real}||_2 + \lambda \cdot \mathcal{L}_{G_{Adv}}(G(c, z))$$

λ **is another hyperparameter of the model.**

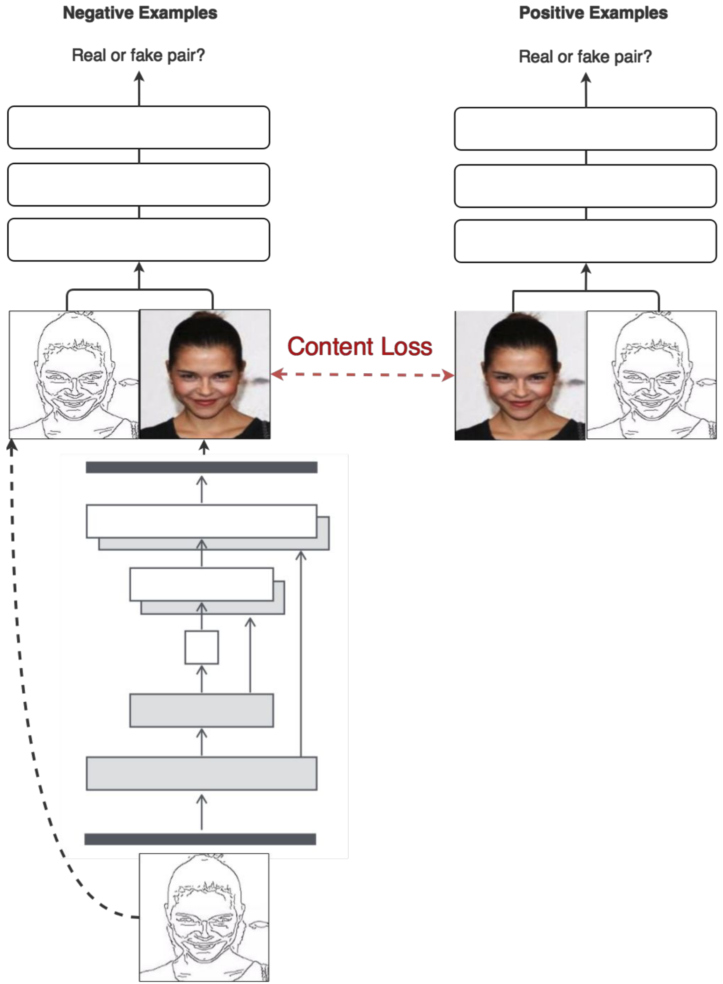


Image2Image with cGANs

Generator loss:

$$\mathcal{L}_G = ||x_{fake} - x_{real}||_2 + \lambda \cdot \mathcal{L}_{G_{Adv}}(G(c, z))$$

λ **is another hyperparameter of the model.**

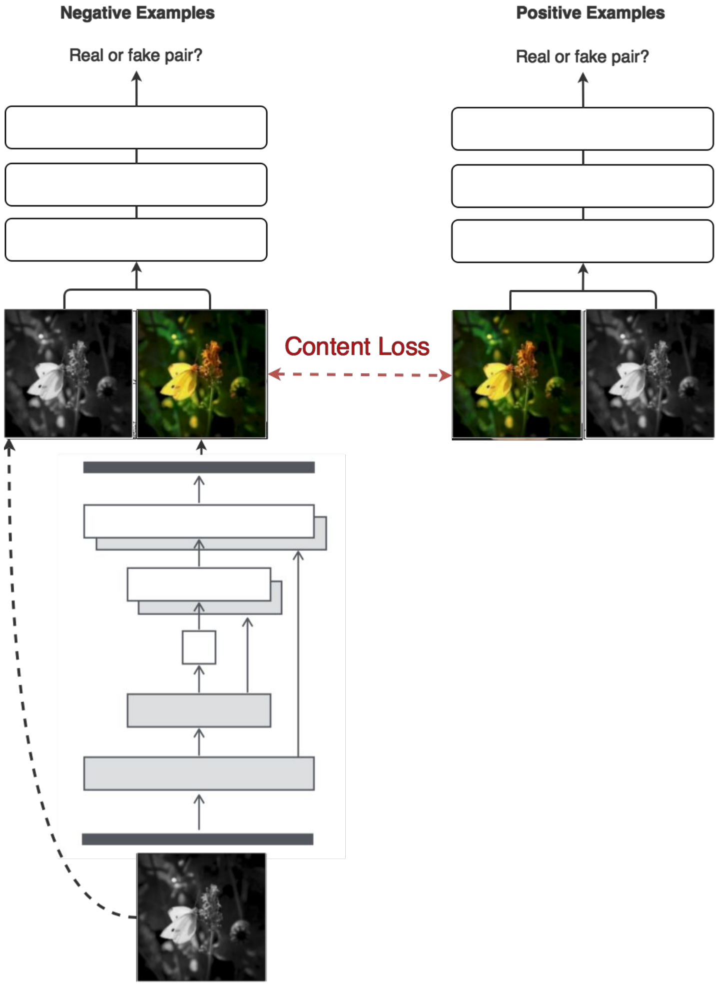
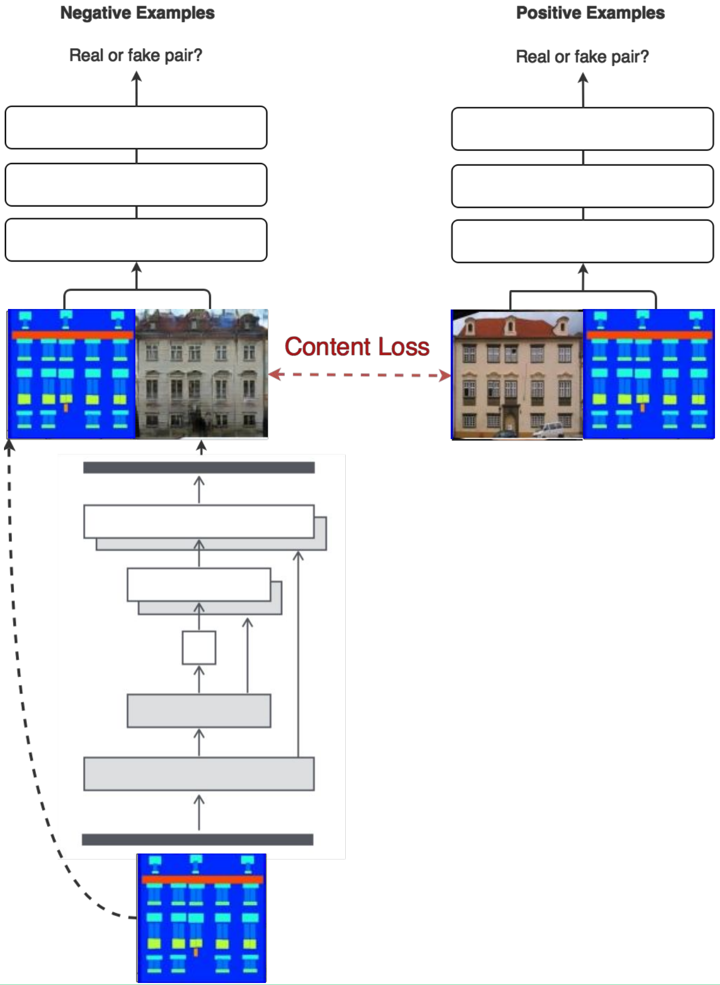


Image2Image with cGANs

Generator loss:

$$\mathcal{L}_G = ||x_{fake} - x_{real}||_2 + \lambda \cdot \mathcal{L}_{G_{Adv}}(G(c, z))$$

λ **is another hyperparameter of the model.**



Demo

Source code: goo.gl/hsVCHx

Data: goo.gl/HrVebr

Demo

Ground truth images 256x256 (Total: 202,599)

- CelebA - cropped and centered
- CelbA - faces in the wild

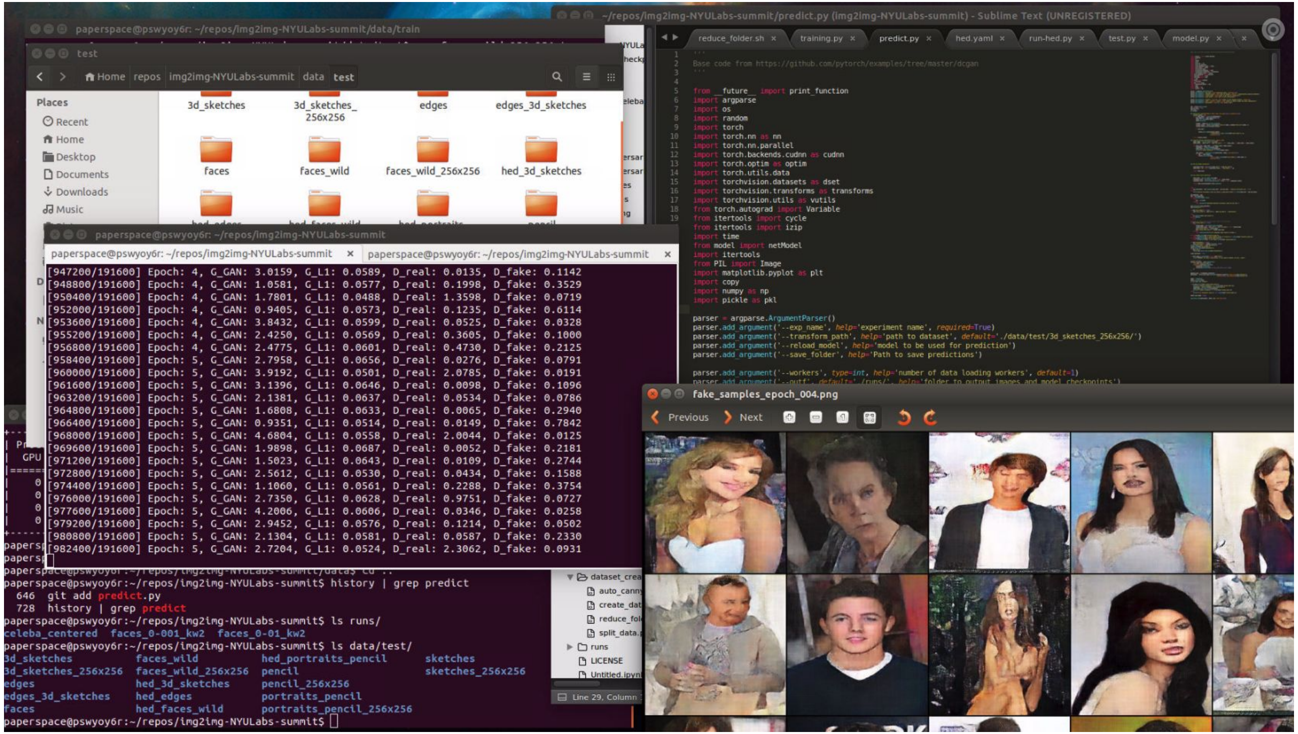
Condition images - Edges

- Canny edges
- HED Edges
    - Holistically-Nested Edge Detection, Xie et al., 2015
    - github.com/harsimrat-eyeem/holy-edge

Demo

Repo: goo.gl/hsVCHx

Data: goo.gl/HrVebr



To think about...

We are not constrained to images

Similar underlying principles can be applied to video, audio, etc.

The network is “dreaming” the details

We cannot say that it is reconstructing the original image **but** one plausible reconstruction. Identikit example.

What do you think you could do with this?

Thank you!

Bibliography

[1] Isola, et al. Image-to-Image Translation with Conditional Adversarial Networks, 2016. arXiv 1611.07004.

[2] Radford, et al. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 2015. arXiv 1511.06434.

[3] Mirza, et al. Conditional Generative Adversarial Nets, 2014. arXiv 1411.1784.

[4] Liu, et al. Deep Learning Face Attributes in the Wild, 2015. ICCV.