Machine Learning

Walt Disney
ANIMATION STUDIOS

RAJESH SHARMA
Walt Disney Animation Studios

Thank you to ACM SIGGRAPH!

Pol Jeremias-Vila : SIGGRAPH 2021 Chair

Tomasz Bednarz: Frontiers Program Chair

Alex Bryant: Student Volunteers Chair

Tim Hendrickson: Digital Marketing Manager

Student Volunteers:

Rogelio, Trinity, Aurora, Emily, Hunter & Kendra

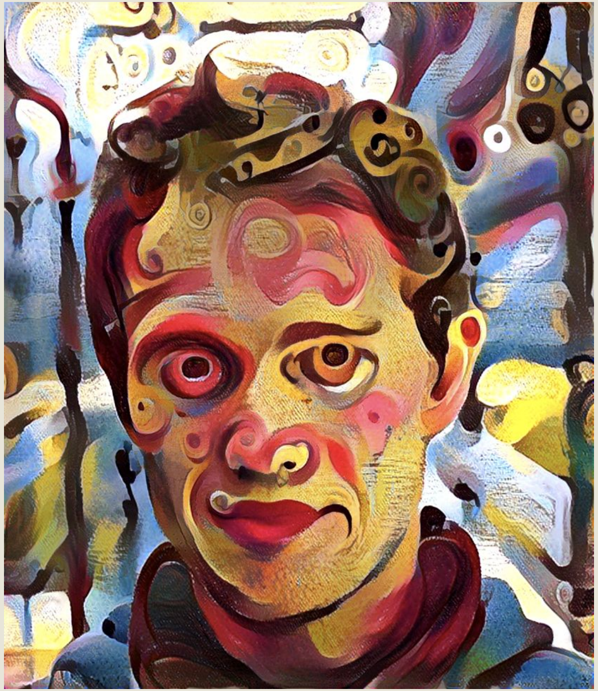
SIGGRAPH 2021

4

Machine Learning

Rajesh Sharma

Leon Gatys



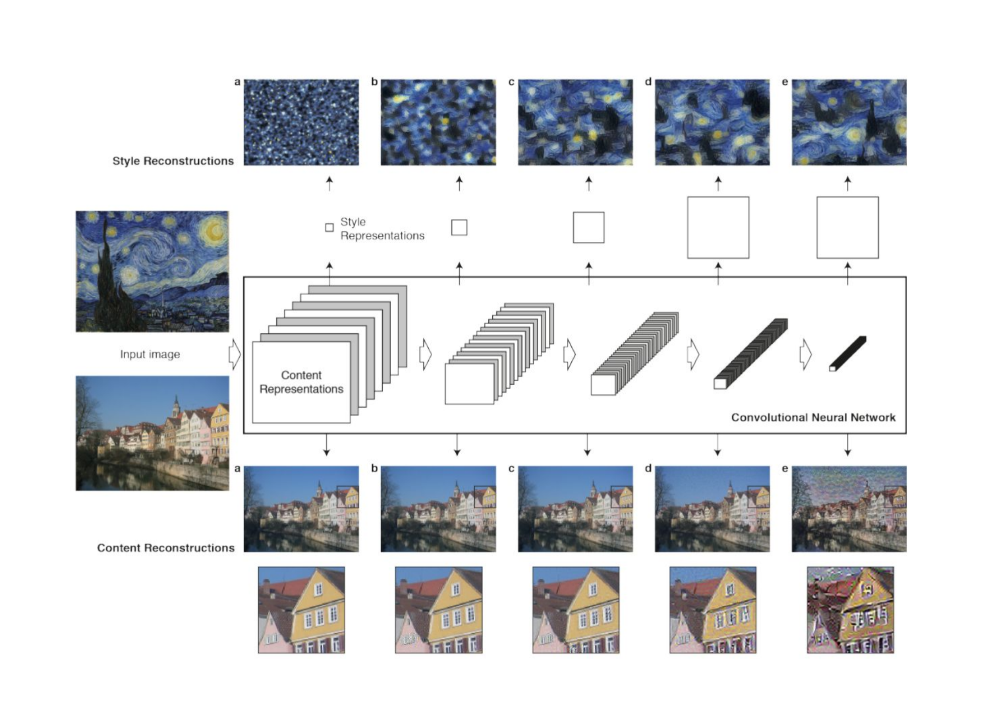
Research Scientist

Research Scientist

Apple

Leon Gatys is a Machine Learning researcher with a focus on modeling human experience. In his PhD thesis, he invented the popular Neural Style Transfer algorithm by using Deep Neural Networks to model Visual Perception. He is currently based in Seattle where he works as a founding member of Apple's Health AI team.

Artistic Style Transfer



$$L_{total}(\vec{c}, \vec{s}, \vec{x}) = \alpha L_{content}(\vec{c}, \vec{x}) + \beta L_{style}(\vec{s}, \vec{x})$$

Today

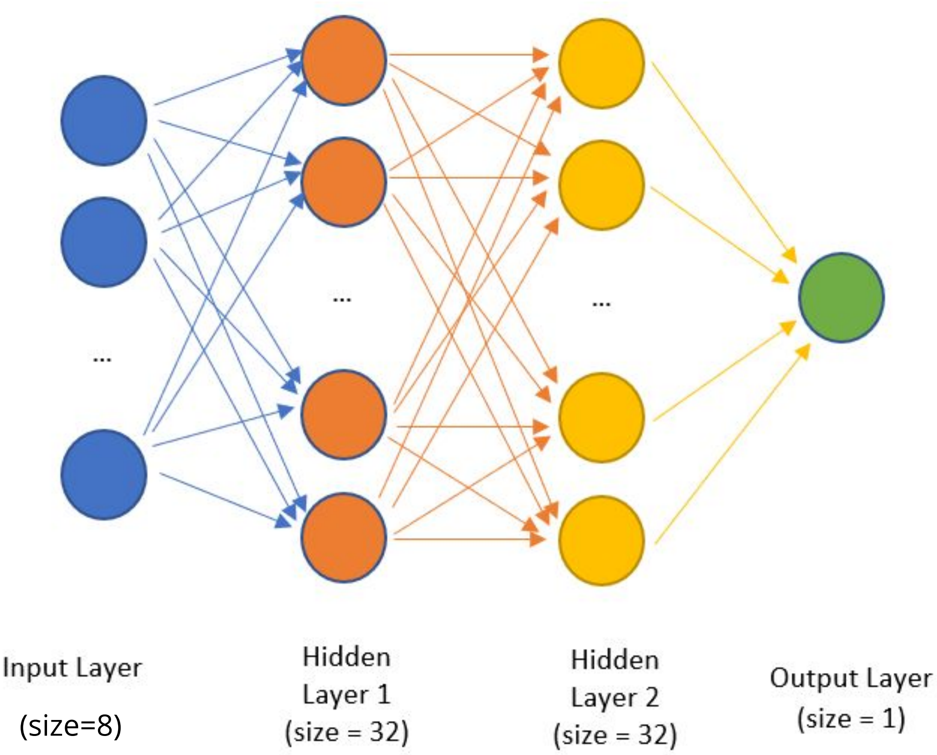
- Quick Recap: Distributions, Autoencoder
- Convolutional Neural Networks
- DataPipeline
- Denoising
- Transfer Learning / PreTrained Models

Hands-on

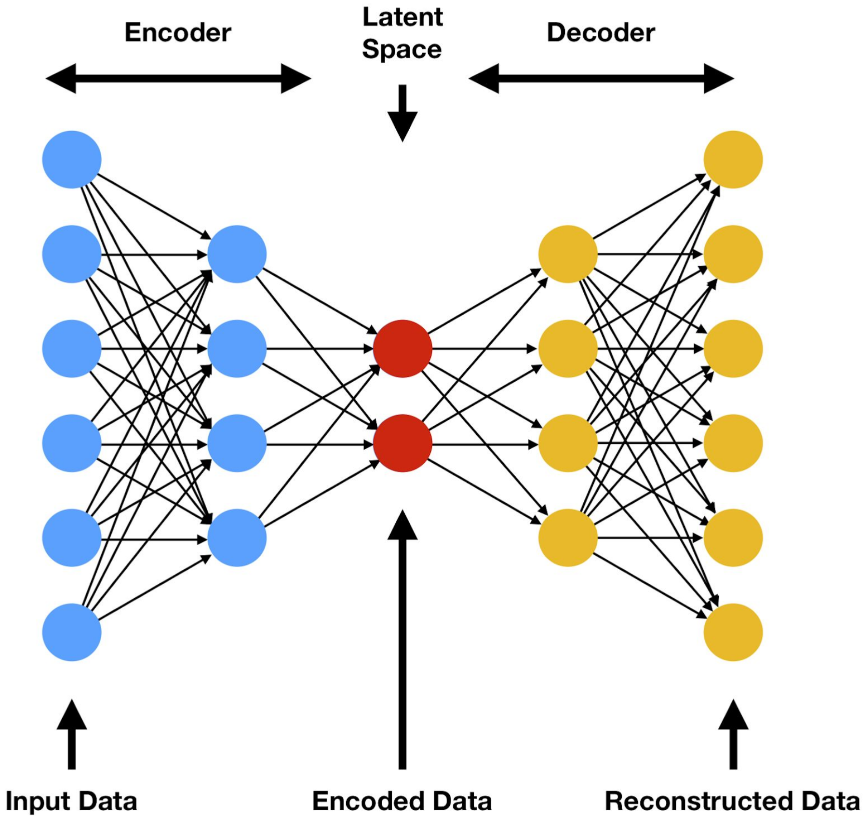
- Log in to your google drive
- Make a shortcut to: https://bit.ly/3oKCVCh
- Make a copy of:
    - Autoencoder.ipynb
    - dataPipeline.ipynb
    - denoiserCNN.ipynb
    - styleTransfer.ipynb
    - facialRecognition01.ipynb

Autoencoder

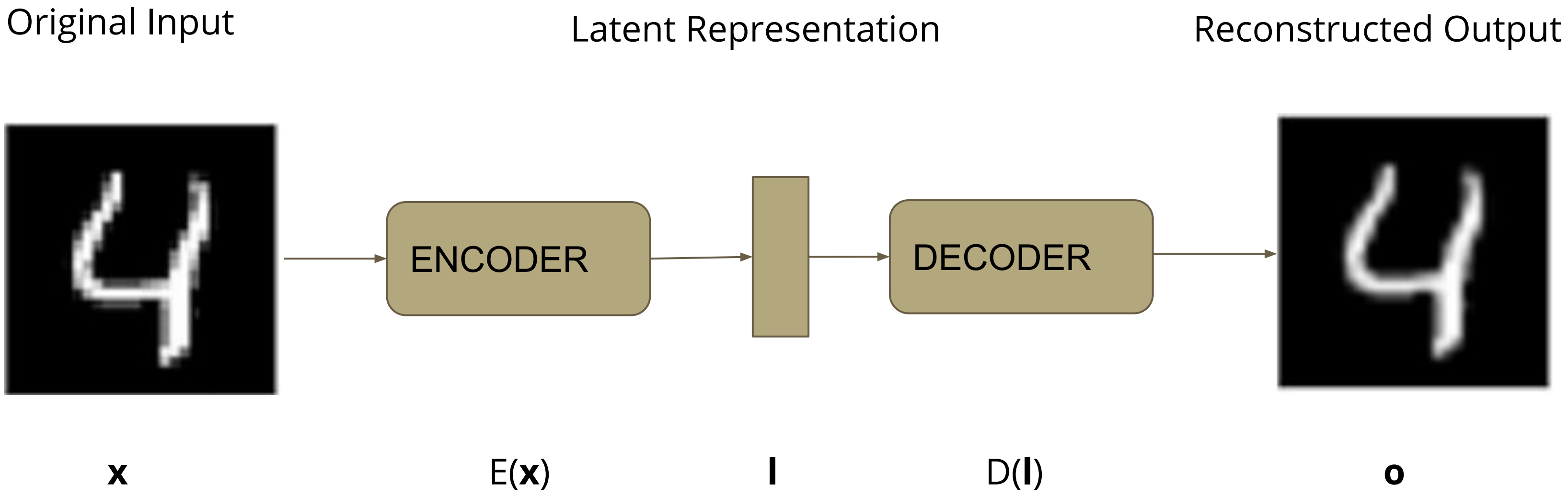
For regression, we had a fully-connected network, output layer size=1



Autoencoder



Autoencoder



Original Input

Latent Representation

Reconstructed Output

ENCODER

DECODER

x

E(**x**)

l

D(**l**)

o

Autoencoder - Model

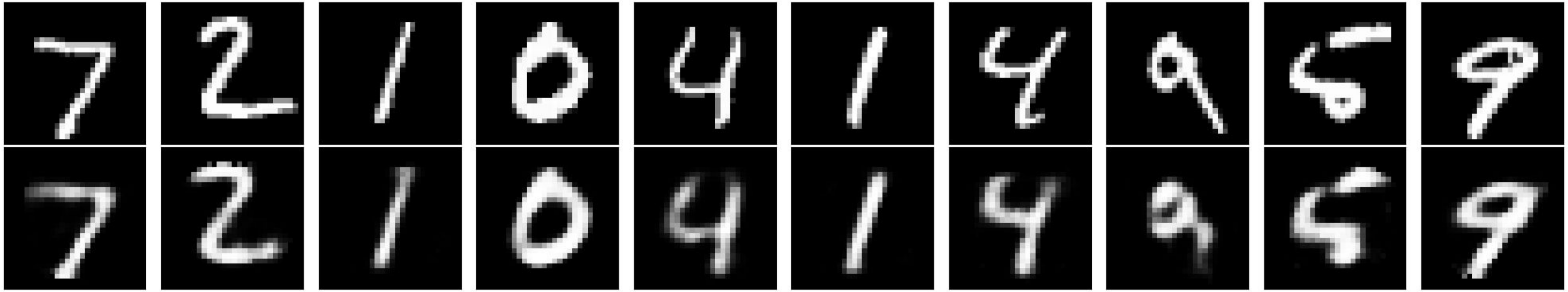
```
# build an autoencoder
model = tf.keras.models.Sequential([
    tf.keras.layers.InputLayer(IMG_SHAPE),
    tf.keras.layers.Flatten(),
    # encoder
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(32, activation='relu'),
    # decoder
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(784, activation='sigmoid'),
    tf.keras.layers.Reshape(IMG_SHAPE)
    ])

# compile
model.compile(optimizer='adamax', loss='mse')

# fit
model.fit(x_train, x_train, epochs=17,batch_size=256, shuffle=True, validation_data=(x_test, x_test))

# predict
decoded_imgs = model.predict(x_test)
```

Autoencoder - results



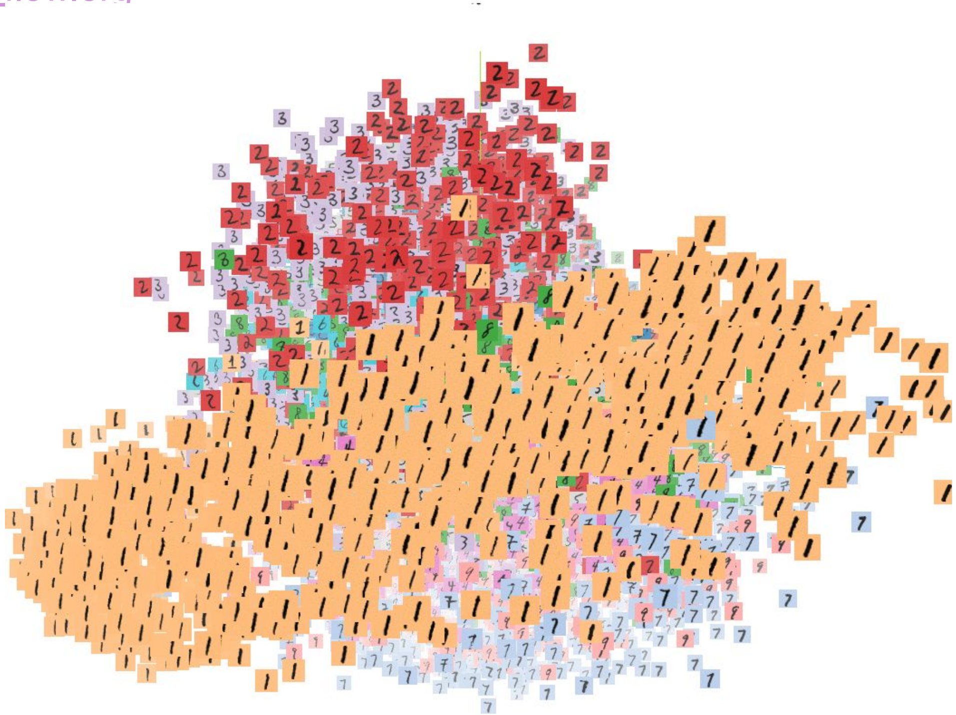
Compression Factor: 28x28/32 ~ 25X

Hands-on

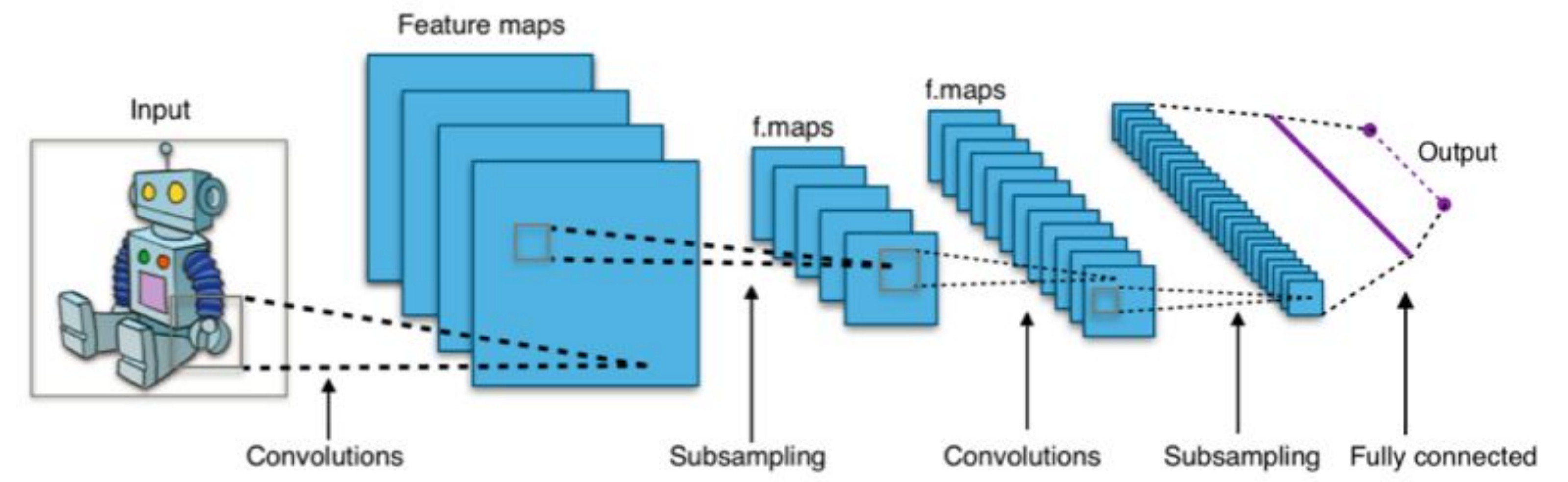
- *Log in to your google drive*
- Find the shared folder
- Make a copy of:
  - AutoEncoder.ipynb

Latent Spaces and Embeddings

https://projector.tensorflow.org



Convolutional Neural Network (CNN)



Convolution (Extract High-Level Features)

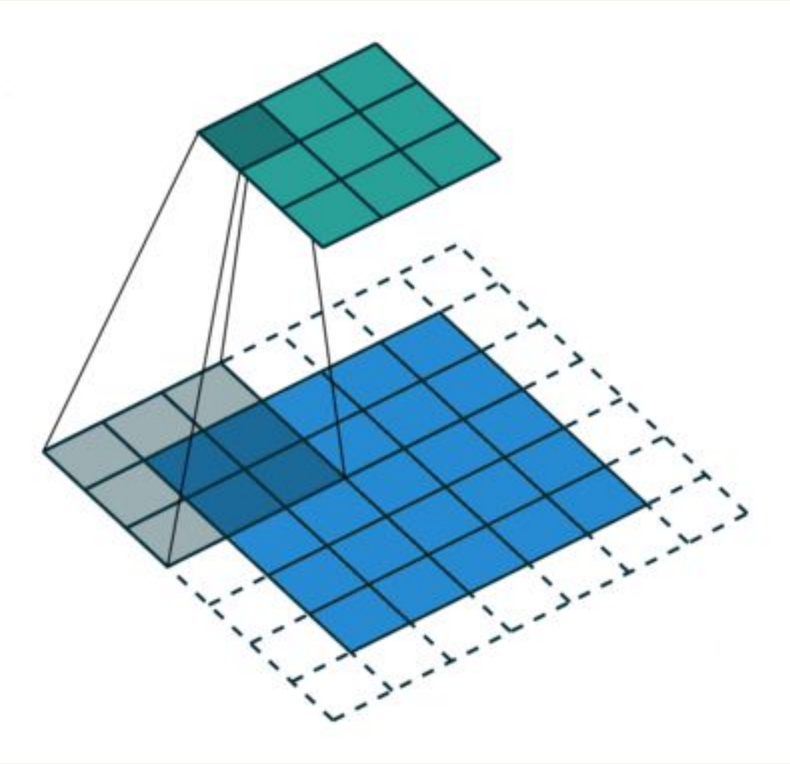
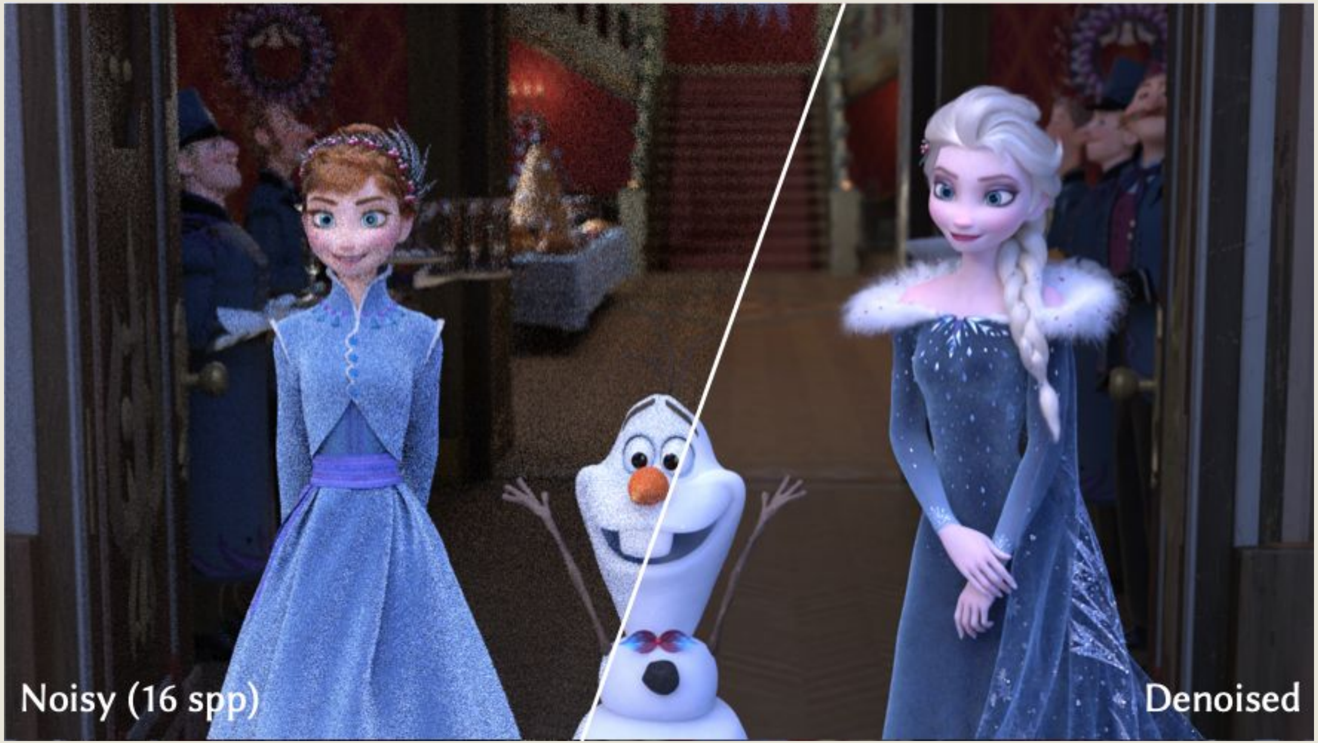
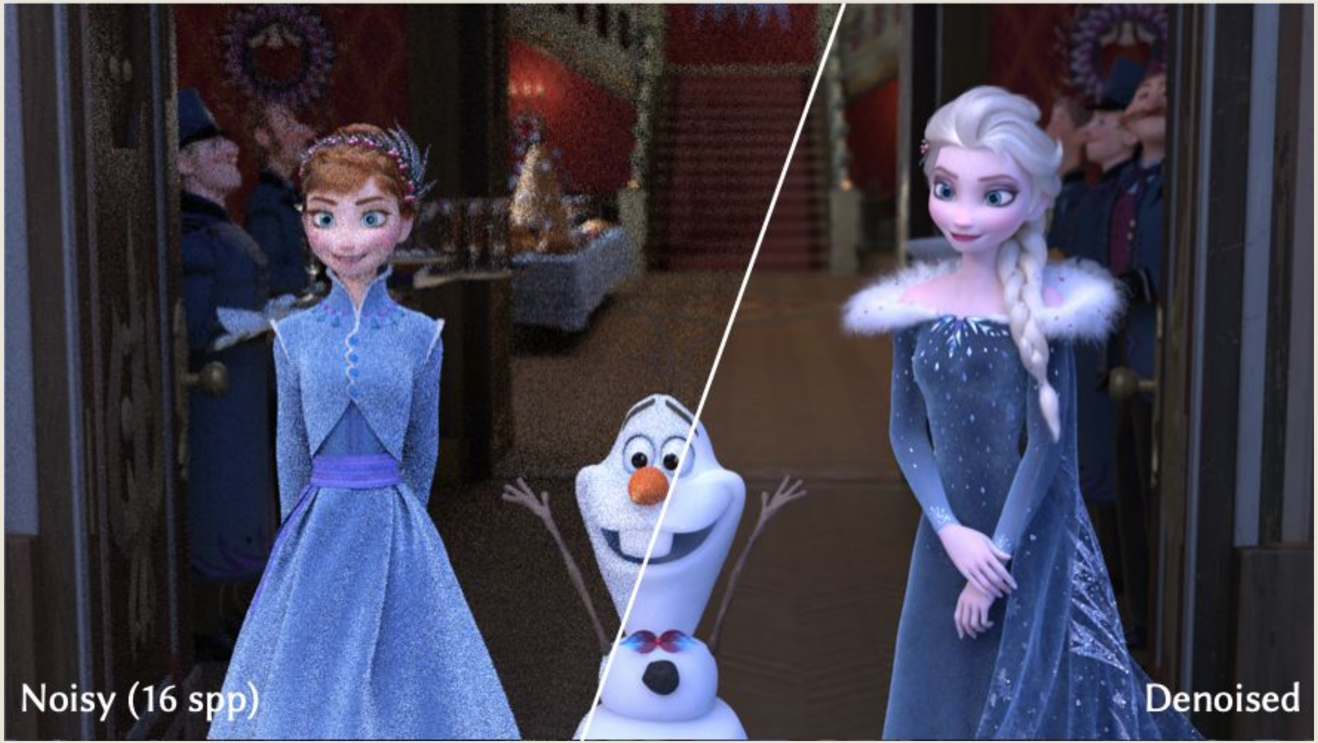


Image Denoising



Denoising with Kernel Prediction and Asymmetric Loss Functions SIGGRAPH 2018, Vogels et al

Noisy image.....<similar image>.....Clean image



Denoising with Kernel Prediction and Asymmetric Loss Functions SIGGRAPH 2018, Vogels et al

Noisy image.....<similar image>.....Clean image

- If we have a set of noisy images and, a set of corresponding clean images,

- We can train our network to recover
  - Clean images from noisy images

- How
  - By setting Clean image as the ground truth,
  - the Noisy image as input and,
  - the loss function as the difference btwn the two

Don’t have a noisy version?

- Take a clean image
- Add synthetic noise to it (Data Augmentation)

But first, we need some more Engineering!

- Take a look at `dataPipeline.ipynb`
  - --tensorflow data sets and pipeline
  - --addNoise
  - --extractPatches

Noisy image...<similar image>...Clean image

- Take a look at denoiserCNN.ipynb
  - Make a CNN

~~Noisy~~ image.....<similar image>.....Clean image

degraded

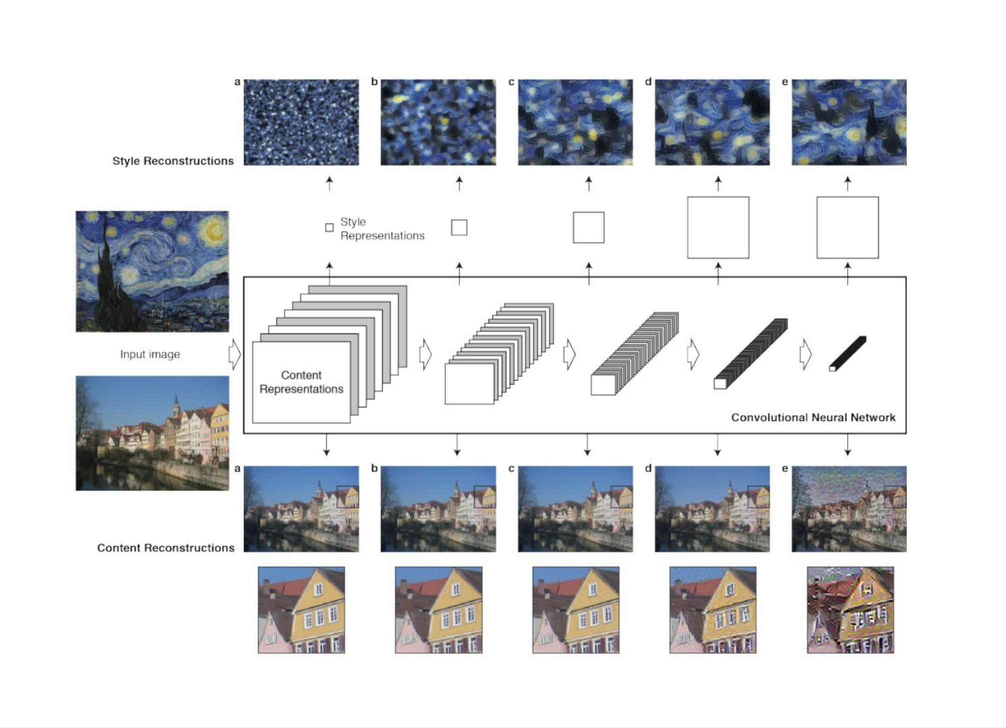
What else can we do?

Tint removal:	Image with tint.....<similar image>.....Clean image
In-painting:	Image with holes....<similar image>...Clean image
Dirt-removal:	Image with speckle....<similar image>...Clean image
Colorization:	Grayscale Image....<similar image>....Color image
Up-resing:	Lowres Image....<similar image>....Hires image
Inbetweening:	Image1, Image3....<similar image>.... Image2

Using off-the-shelf pretrained models

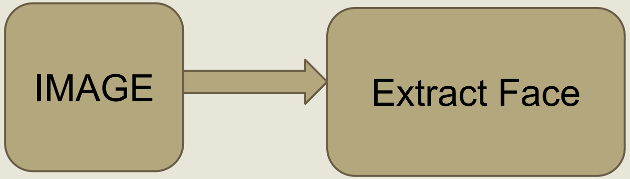
- Style Transfer
- MT-CNN

Artistic Style Transfer



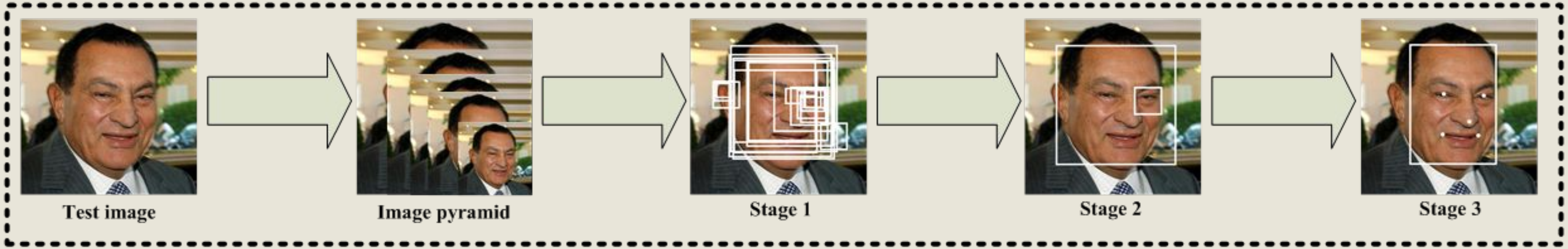
$$L_{total}(\vec{c}, \vec{s}, \vec{x}) = \alpha L_{content}(\vec{c}, \vec{x}) + \beta L_{style}(\vec{s}, \vec{x})$$

Extracting Faces -- MT-CNN



IMAGE

Extract Face



Homework:

Colorization: `tf.image.adjust_saturation`

Up-resing:

```
tf.image.resize(image, size=[256,256], method=tf.image.ResizeMethod.NEAREST_NEIGHBOR)
```

In-Painting:

```
mask = np.ones((PATCH_WIDTH, PATCH_HEIGHT), dtype=np.float32)
scale = 0.25
low, upper = int(PATCH_WIDTH * scale), int(PATCH_HEIGHT * (1.0 - scale))
mask[:, low:upper, low:upper] = 0.
tf.multiply(patch, mask)
```

Frame interpolation:

```
stacked = tf.concat([frame1, frame3], axis=-1)
```

Next Class

- Generative Neural Networks:
  - Variational AutoEncoder
  - Generative Adversarial Networks
- Homework:
  - Do other kinds of ‘denoising’
- @xarmalarma, #siggraph2021