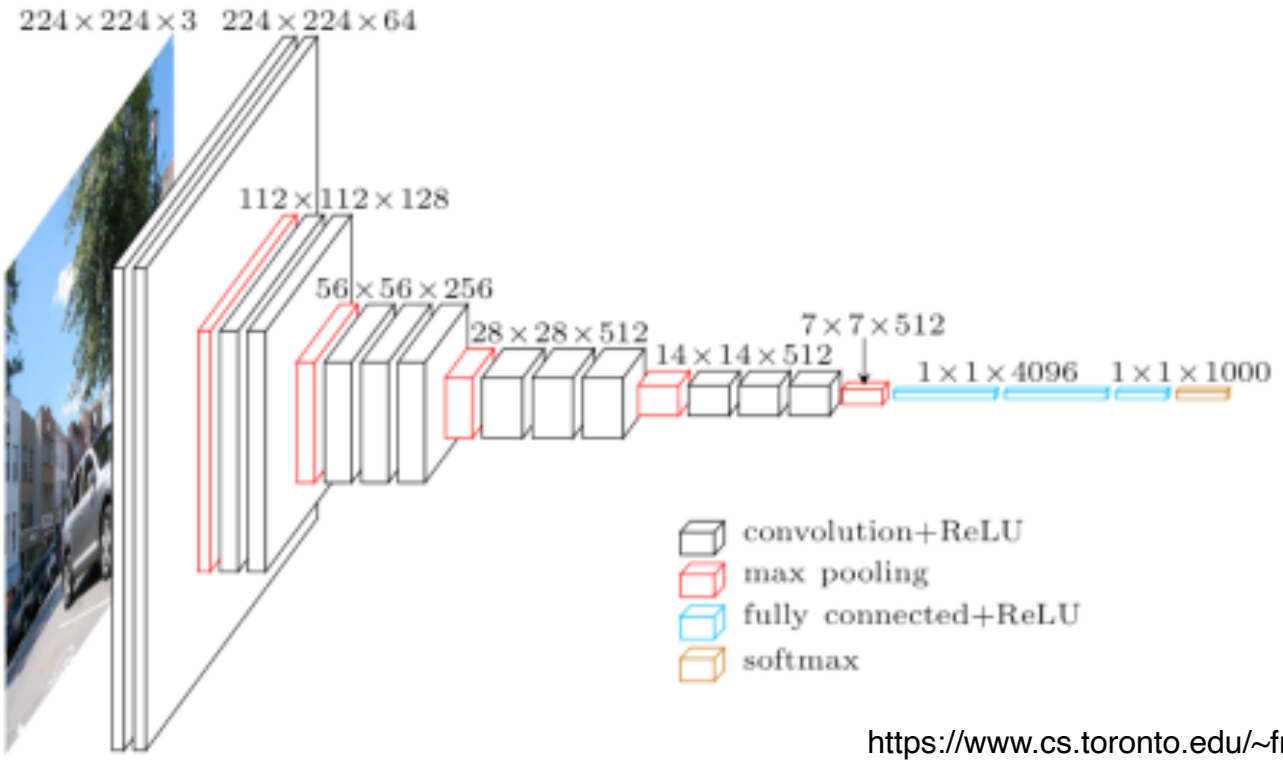
Deep Learning Architectures

Mai H. Nguyen, Ph.D.

Convolutional Neural Network (CNN)

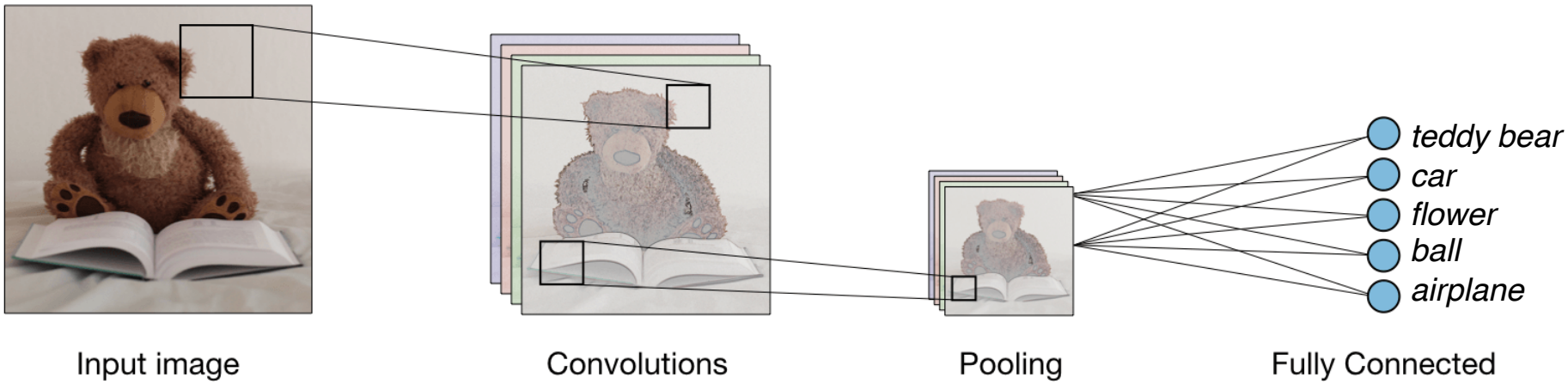
- Model consists of several repeating sets of layers called ‘blocks’
- Input volume is image of size width X height X # of channels
- Output is vector of numbers representing class probabilities



https://www.cs.toronto.edu/~frossard/post/vgg16/

CNN

- **General CNN Architecture**
  - Has sequence of layers
  - Each layer transforms its input to generate an output through nonlinear function
  - Has different types of layers

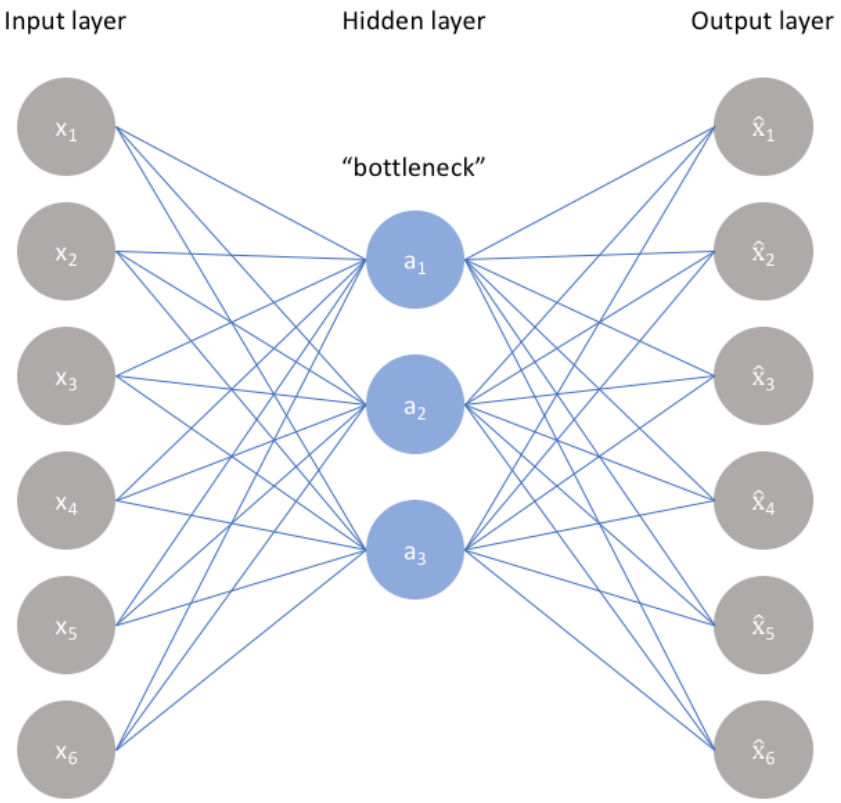


https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks#style-transfer

CNN Applications

- **Image Analysis**
  - Object classification, localization, detection
  - Face recognition
  - Text classification
- **Natural Language Processing**
  - Topic modeling
  - Part-of-speech tagging
- **Others**
  - Drug design
  - Crime hot spots identification
  - House price prediction

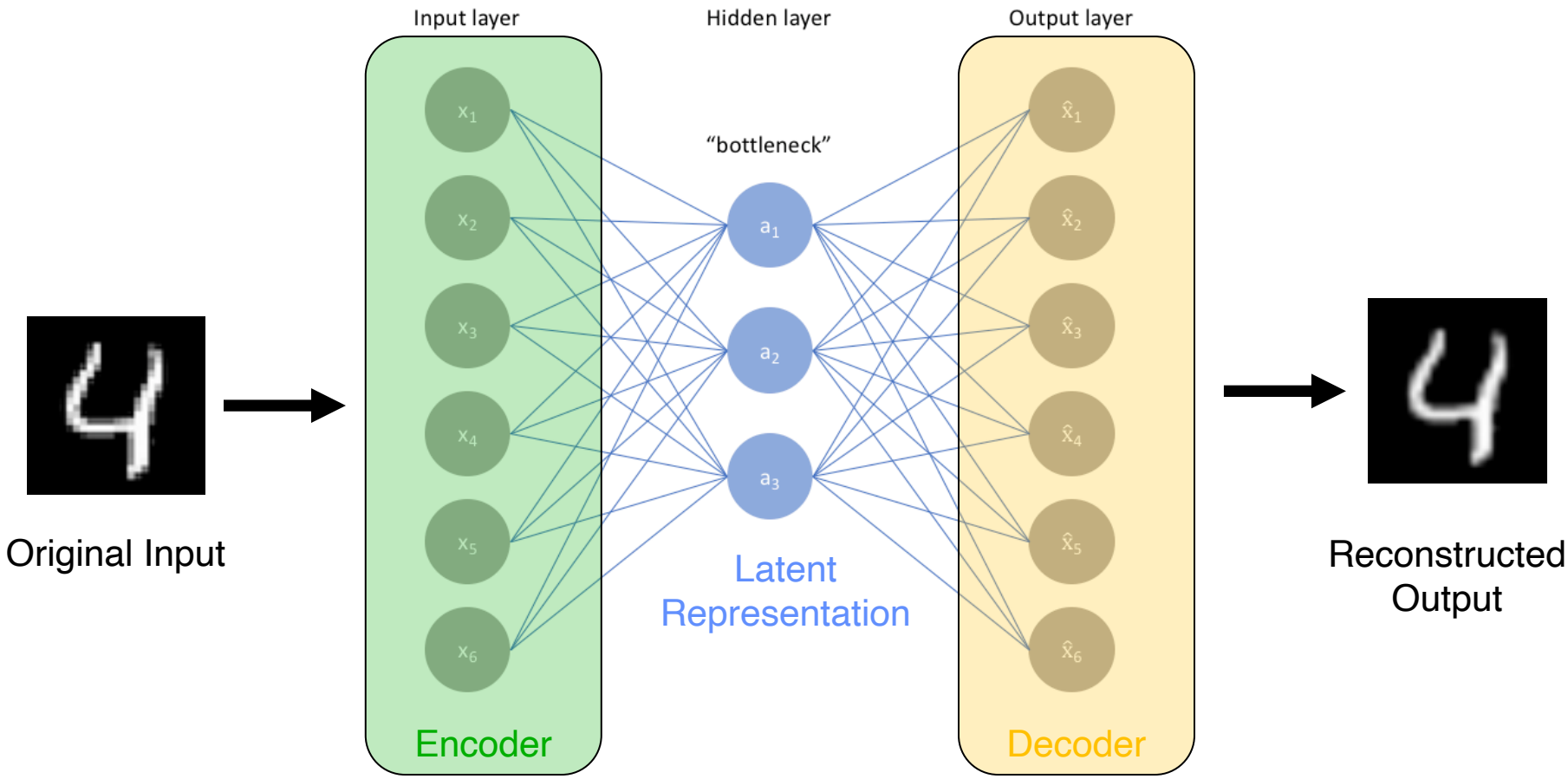
Autoencoder



https://www.jeremyjordan.me/autoencoders/

- Input is fed to hidden layer
- Output is reconstructed version of input
- Model learns to reconstruct input data

Autoencoder



- "Bottleneck" layer provides encoding of input
- Used to generate latent representation of data

https://www.jeremyjordan.me/autoencoders/

Autoencoder

- **Variations**
  - Sparse
  - Denoising
  - Contractive
  - Variational
- **Uses**
  - Feature learning
    - Generated features useful for downstream tasks (e.g., classification, anomaly detection, clustering)
  - As part of larger deep learning model

U-Net

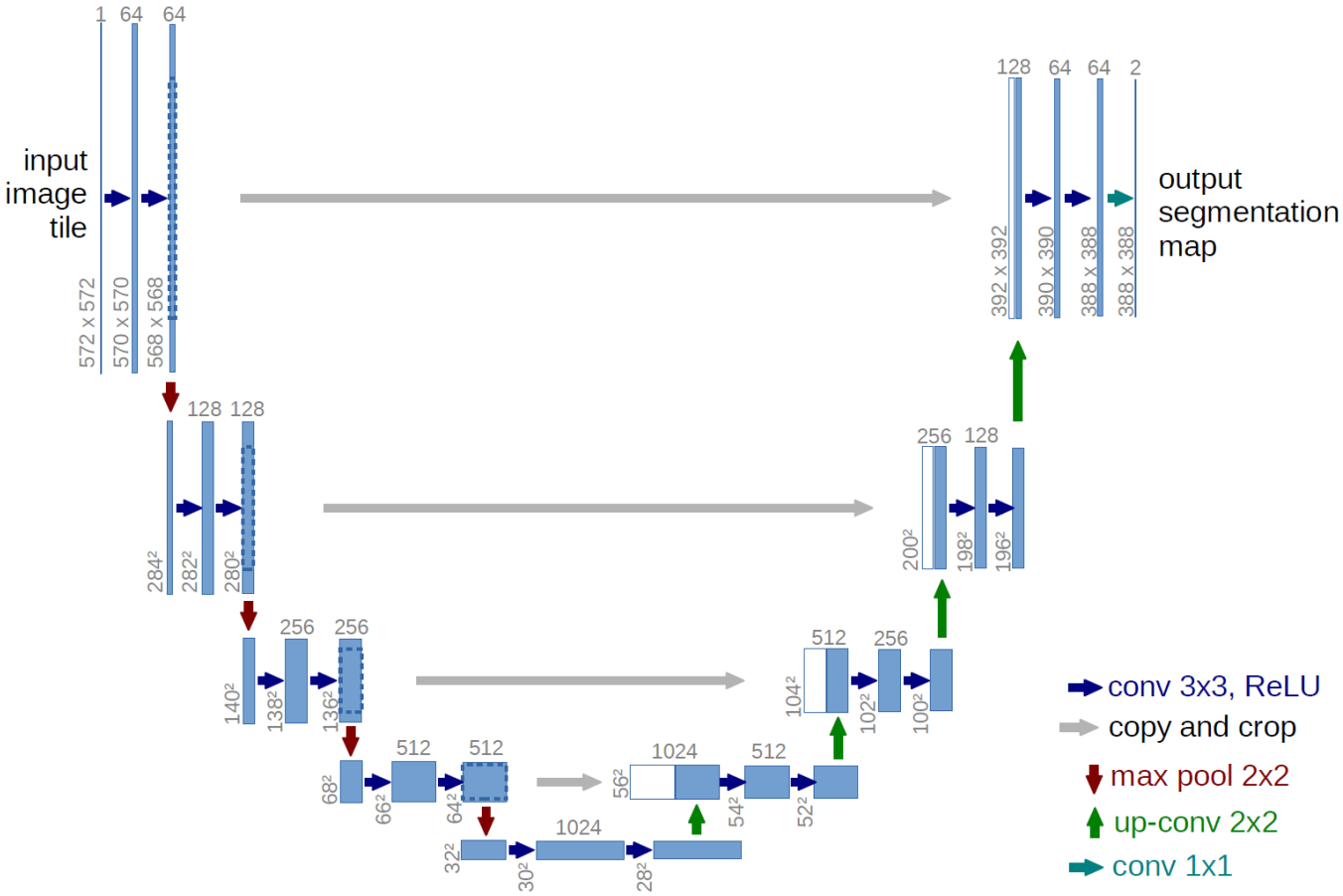
- **Semantic Segmentation**
  - Dividing image into multiple salient image regions
  - Assign label to every pixel in image
  - Pixels with same label are similar

https://medium.com/@keremturgutlu/semantic-segmentation-u-net-part-1-d8d6f6005066

U-Net

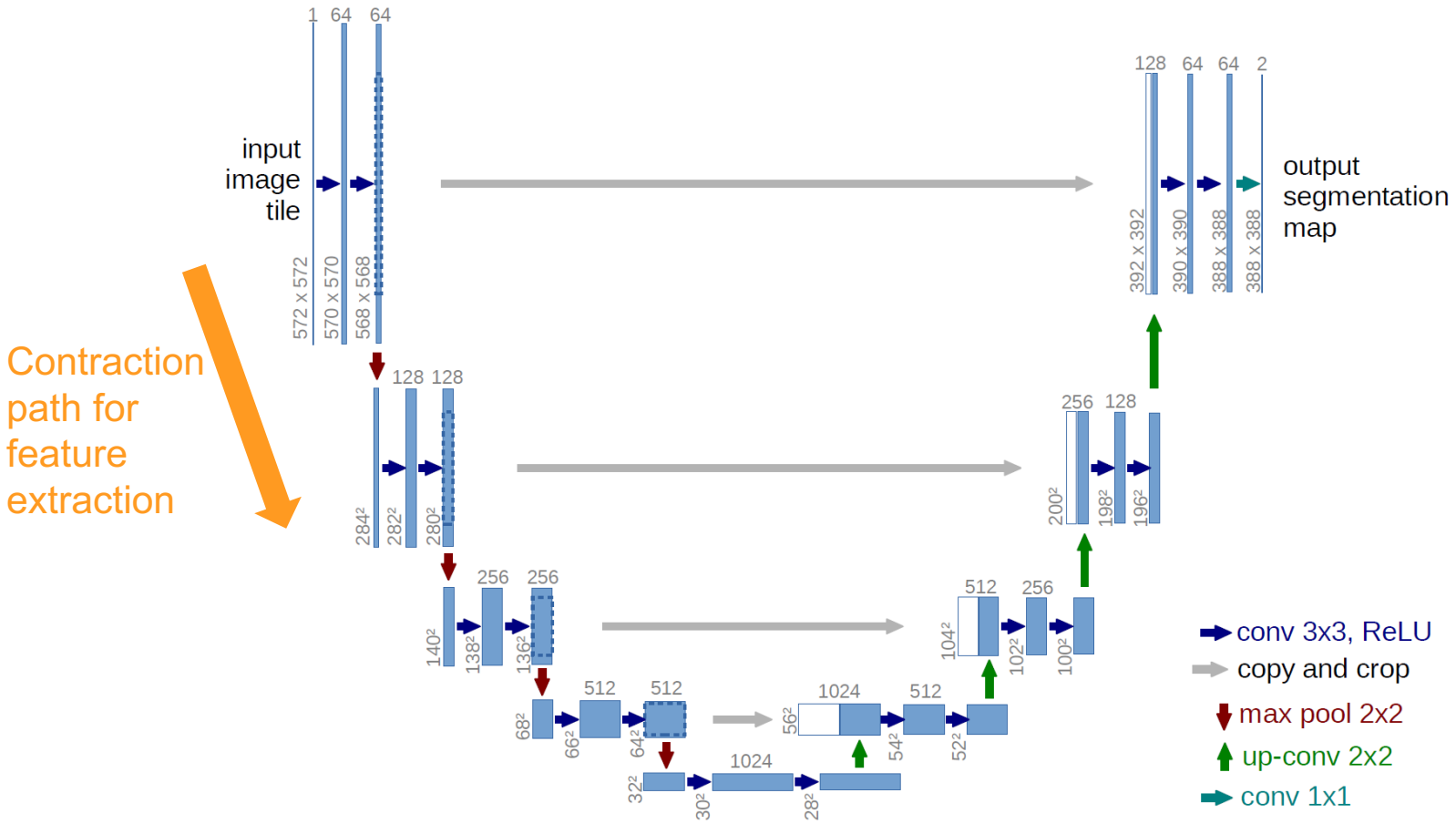
- **Used for image segmentation**
- **Architecture**
  - Encoder-decoder network
  - Contracting part of network performs feature extraction
    - Encoding path
  - Expansion part of network performs segmentation
    - Decoding path

U-Net Architecture



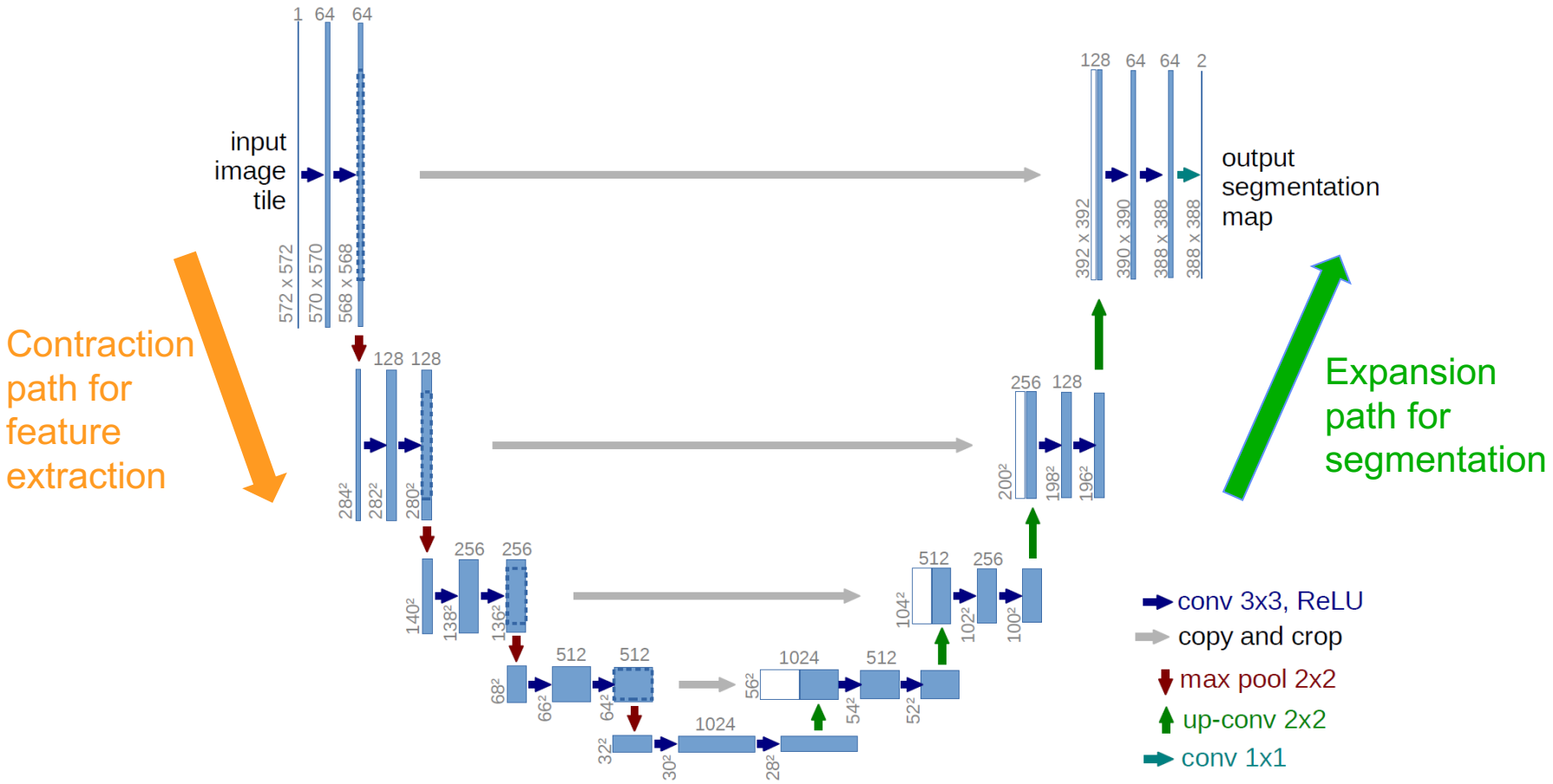
https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

U-Net Architecture



https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

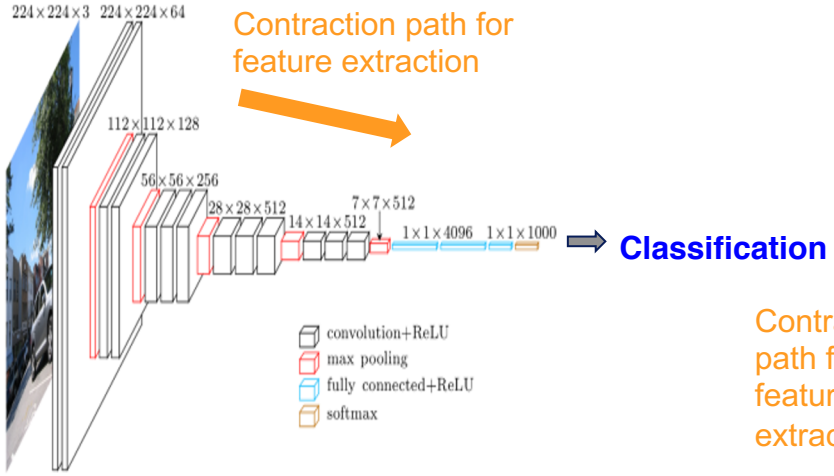
U-Net Architecture



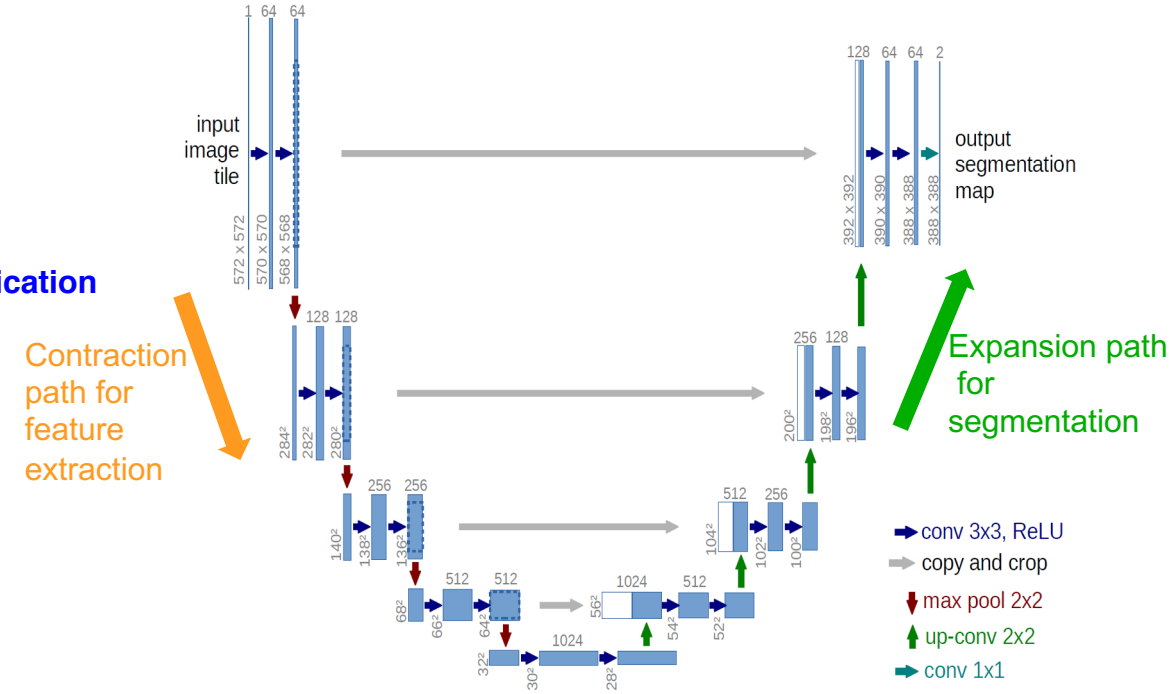
https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

U-Net Architecture

VGG16 CNN Architecture



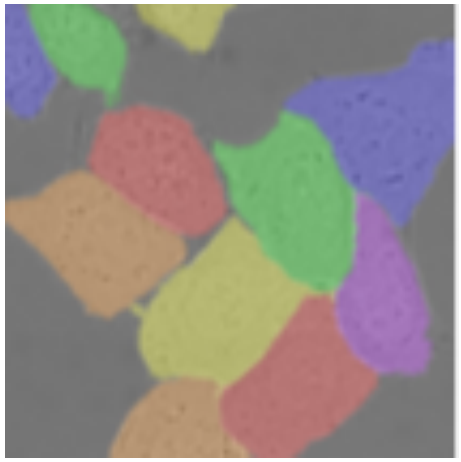
U-Net Architecture



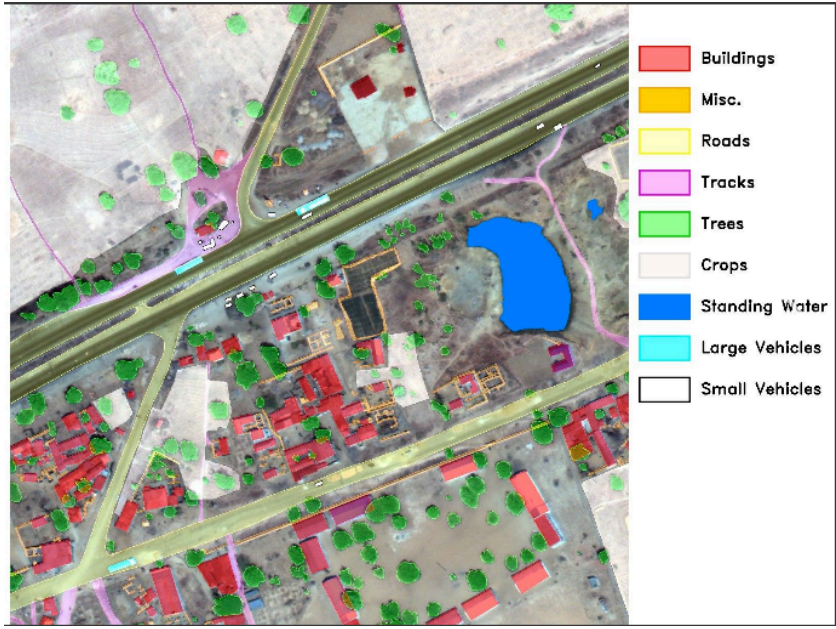
https://spark-in.me/post/unet-adventures-part-one-getting-acquainted-with-unet.

https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

U-Net Applications

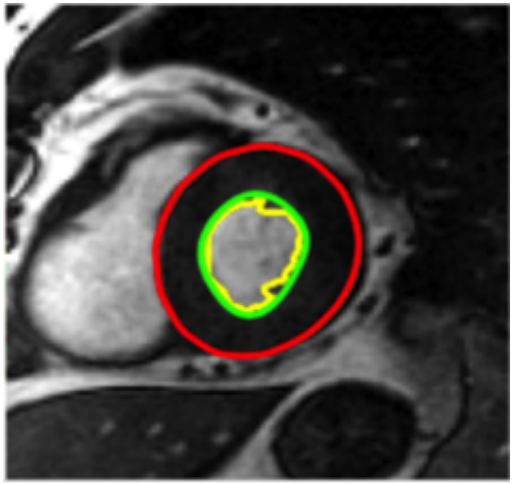


Biomedical Segmentation



Satellite Image Processing

Object Detection



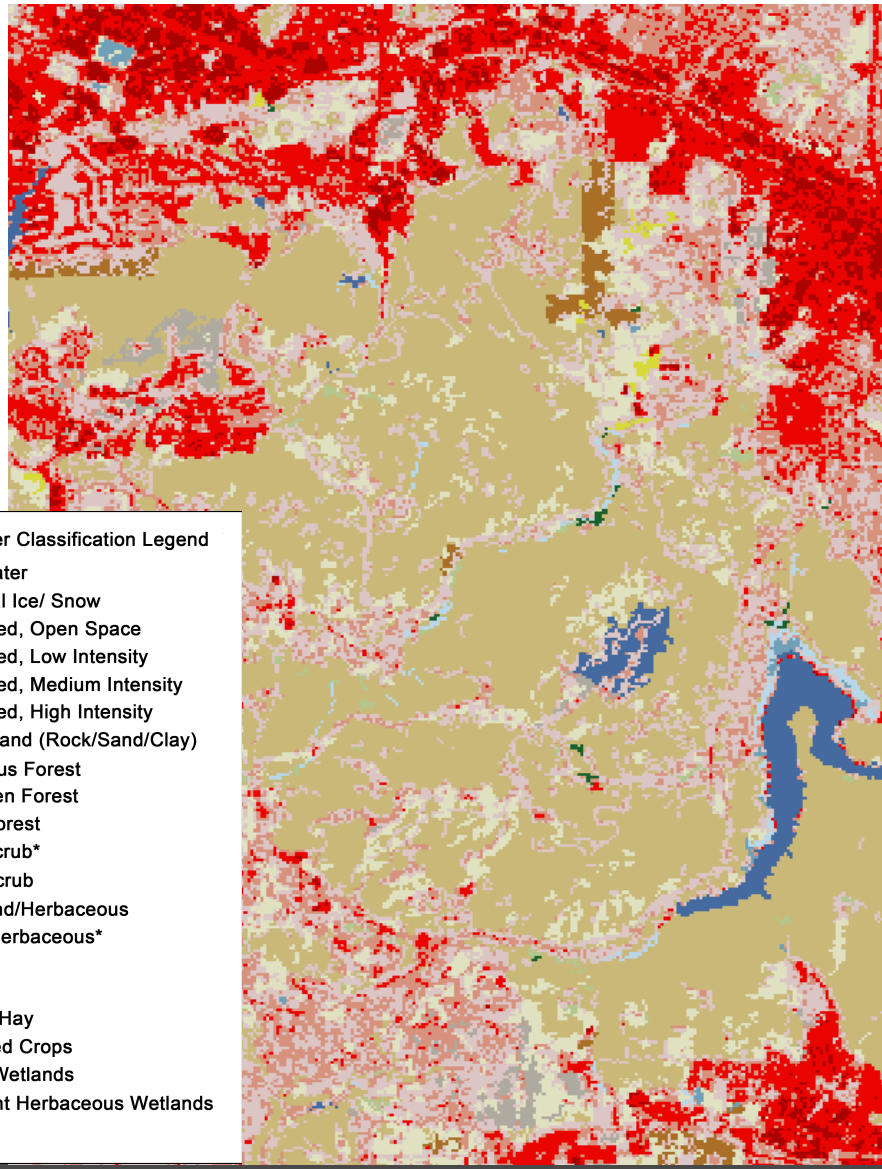
Medical Image Analysis

U-Net Use Case

Satellite Image Analysis

- **Goal**
  - Generate land cover maps from satellite imagery
- **Motivation**
  - Land data products are critical for many applications
  - Current land data products are released every few years
  - Want to generate land data products at scale, as needed, and based on up-to-date data
- **Approach**
  - Use deep/machine learning techniques to extract and analyze features from satellite imagery

Land Cover Map Example



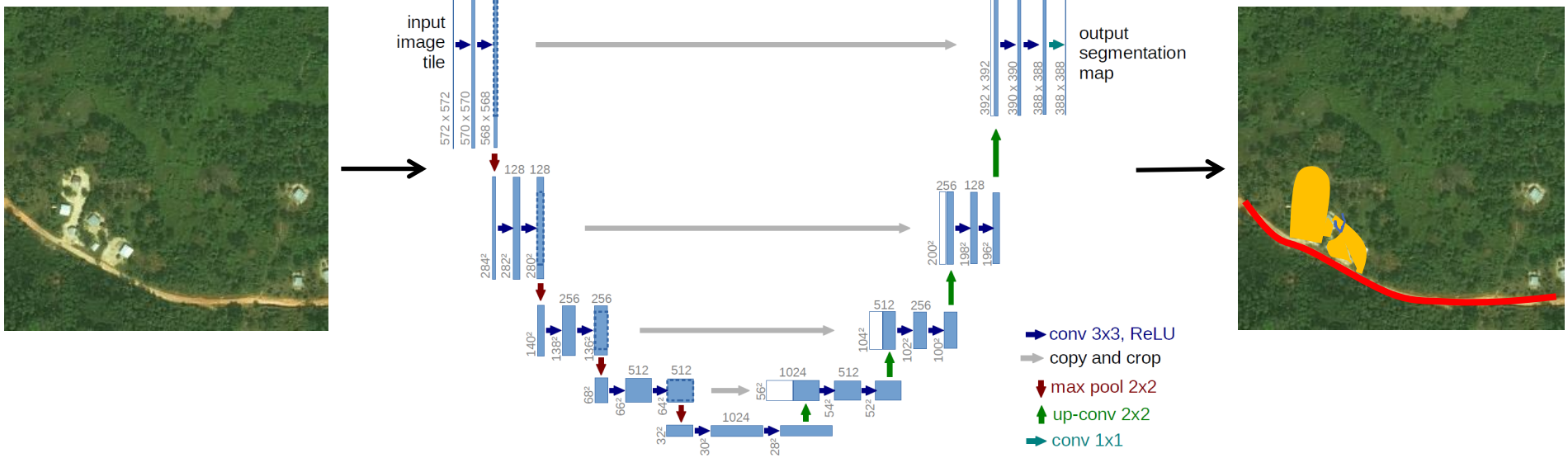
NLCD Land Cover Classification Legend

- 11 Open Water
- 12 Perennial Ice/ Snow
- 21 Developed, Open Space
- 22 Developed, Low Intensity
- 23 Developed, Medium Intensity
- 24 Developed, High Intensity
- 31 Barren Land (Rock/Sand/Clay)
- 41 Deciduous Forest
- 42 Evergreen Forest
- 43 Mixed Forest
- 51 Dwarf Scrub*
- 52 Shrub/Scrub
- 71 Grassland/Herbaceous
- 72 Sedge/Herbaceous*
- 73 Lichens*
- 74 Moss*
- 81 Pasture/Hay
- 82 Cultivated Crops
- 90 Woody Wetlands
- 95 Emergent Herbaceous Wetlands

* Alaska only

Deep Learning

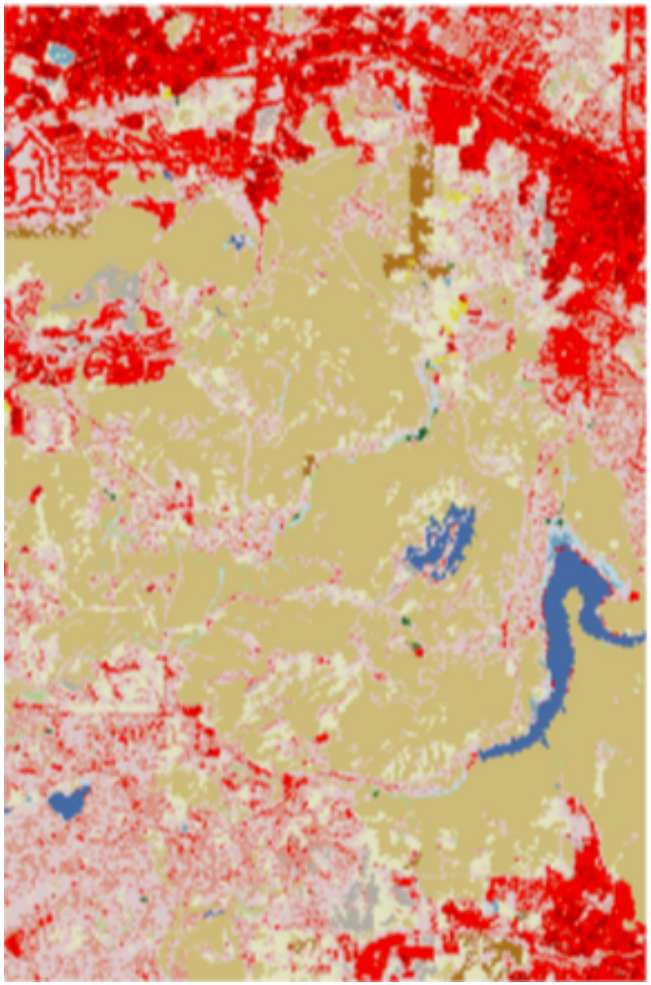
U-Net



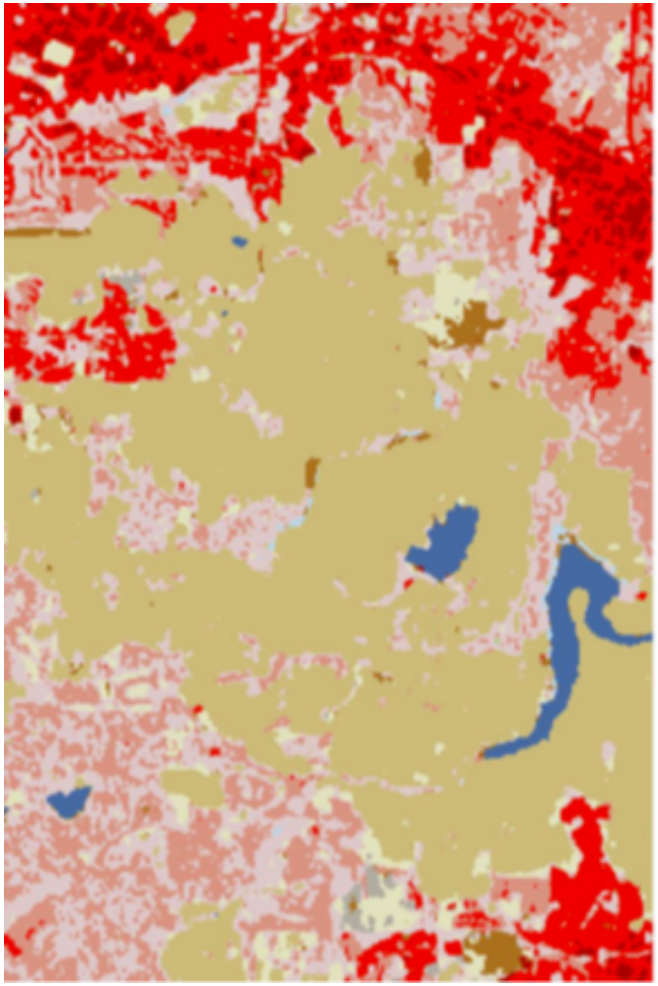
https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

Land Cover Predictions

Original Labels



U-Net Predicted Labels

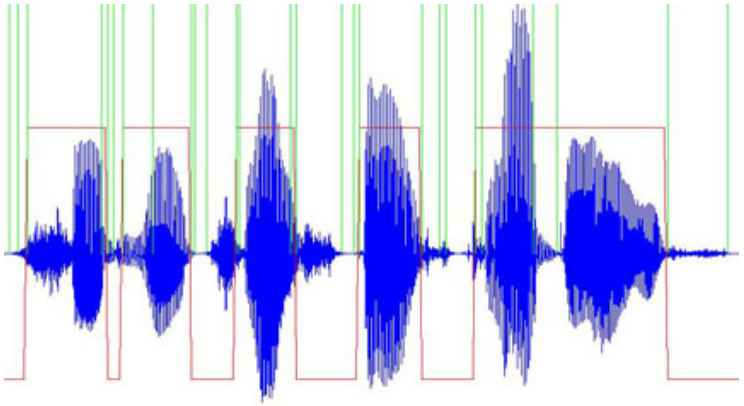


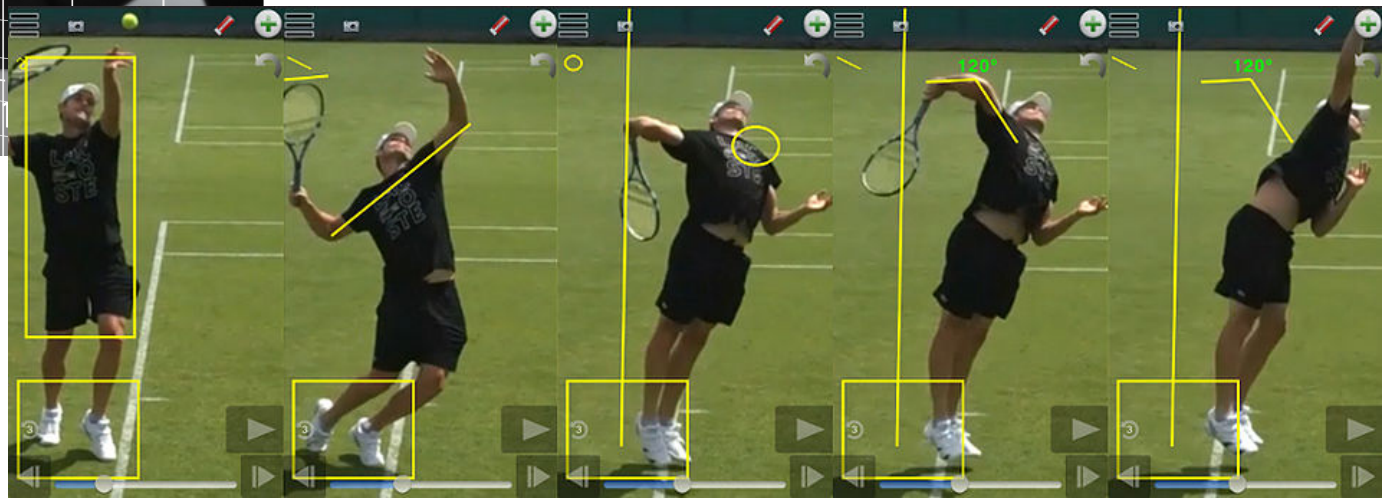
LSTM

- **Sequence Learning**
  - Learning a signal with an ordering or time component

Stock Price

Speech

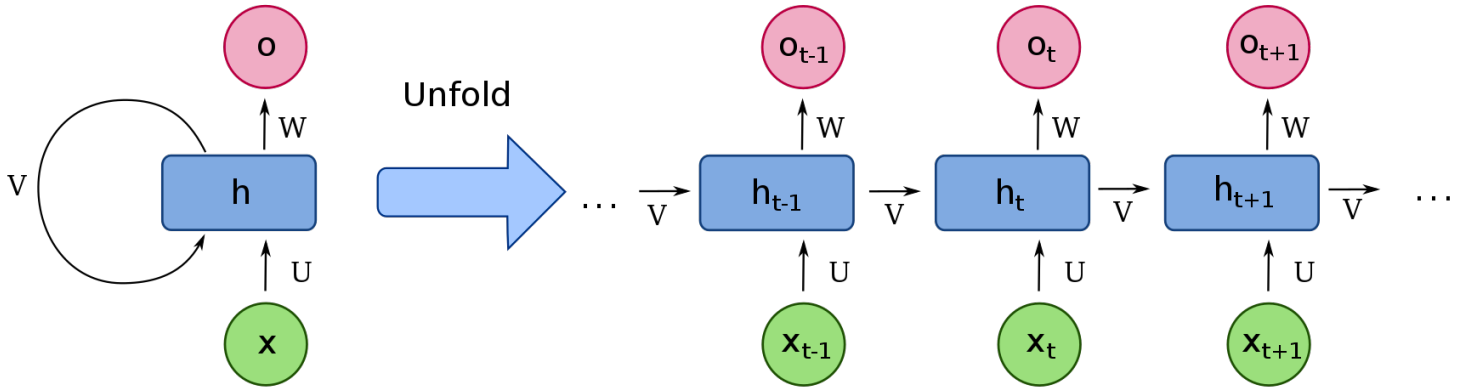




Video

Recurrent Neural Network (RNN)

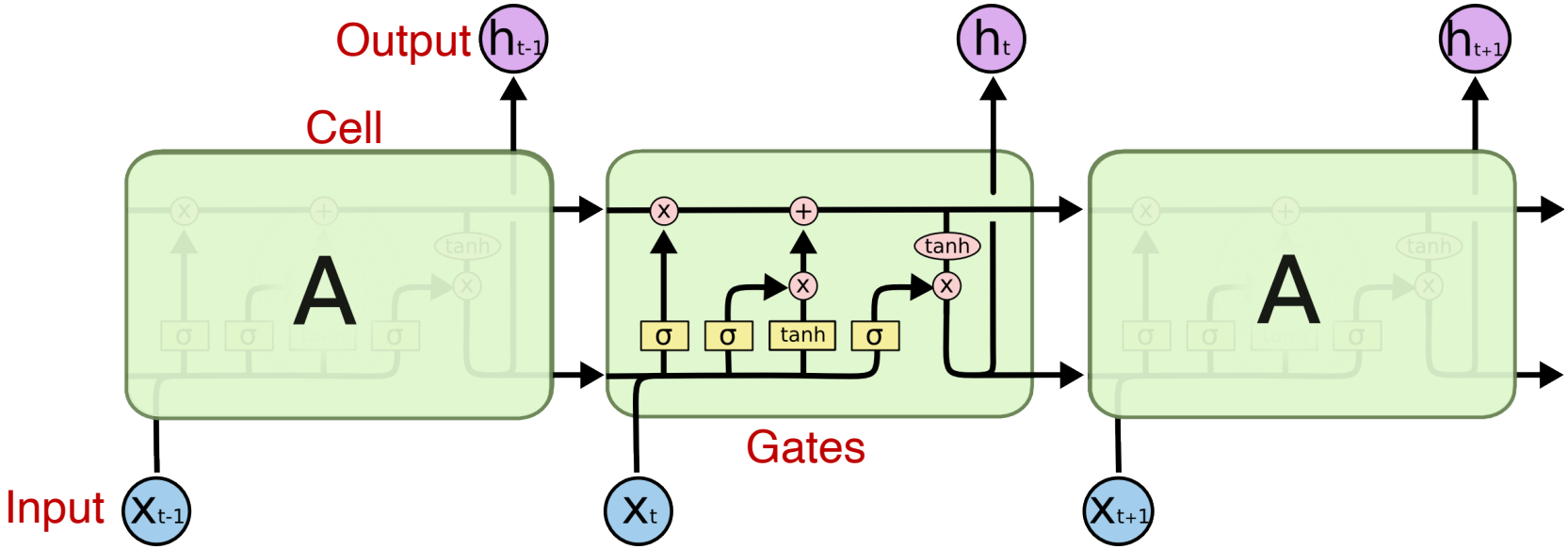
- **Can model sequences and time-dependent signals**
- **Have cyclic connections that feed previous activations as part of input back to network**
  - Allows for temporal contextual information to be stored
  - Predictions at current time step depend on current input and previous predictions
  - Context required must be learned



Long Short-Term Memory (LSTM)

- **Issues with RNN training**
  - Vanishing gradients and exploding gradients
  - Weight of contextual input decays or blows up
  - Thus, contextual info is limited in practice
- **LSTM**
  - Type of RNN
  - Addresses (some of) gradient issues with conventional RNN training

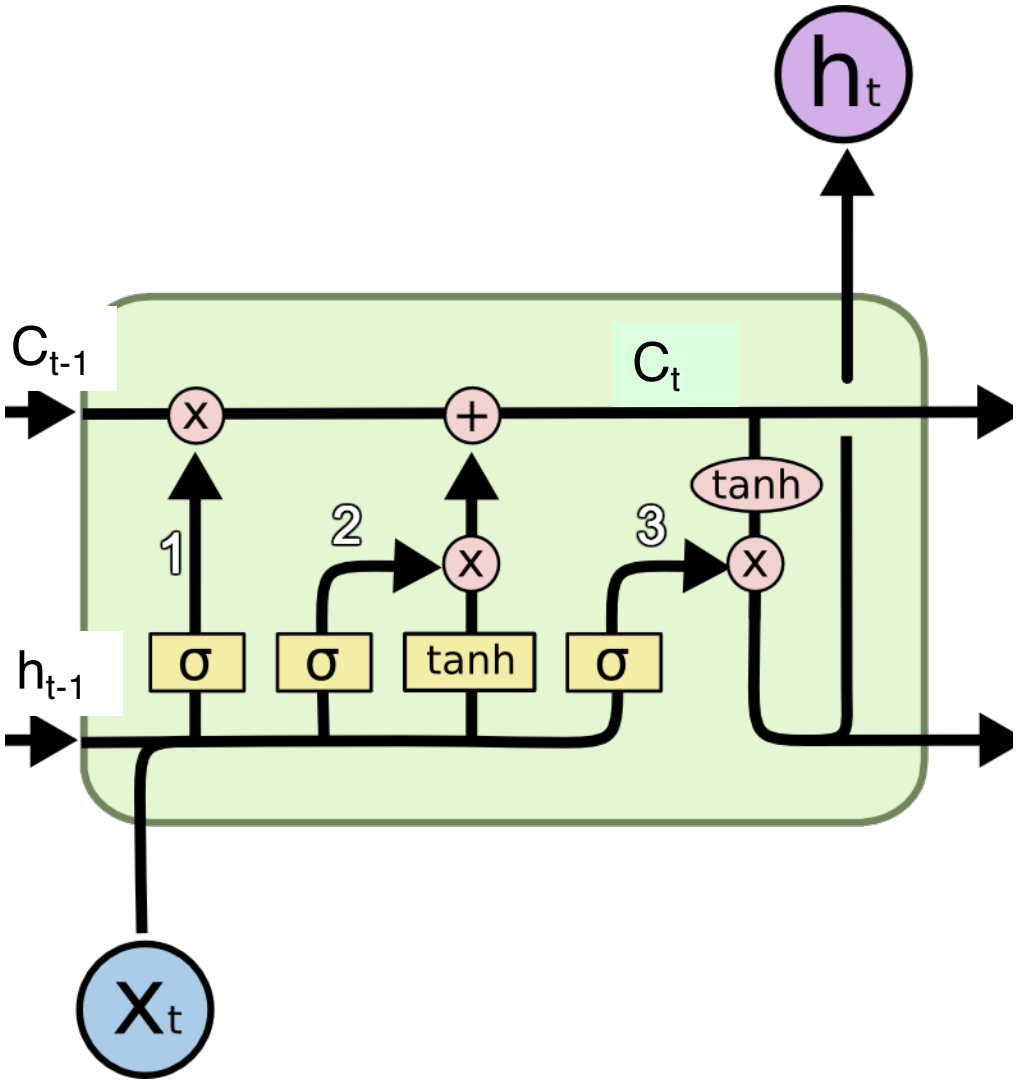
LSTM Architecture



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

- Info flows through memory blocks called ‘cells’
- Structure of cell allows LSTM to selectively remember/forget pieces of information
- Each cell manipulates memory through ‘gates’

LSTM Cell



- X_t
  - Current input
- C_{t-1}
  - Previous cell state
  - Long-term memory
- h_{t-1}
  - Previous hidden state
  - Output from last cell
  - Working memory
- h_t
  - Current output
- **1: forget gate**
  - Removes info
- **2: input gate**
  - Adds info
- **3: output gate**
  - Selects useful info as output

LSTM for Performance Prediction

- **Predict job status of computer processes**
  - To maximize resource allocation
- **Complex and distributed workflows**
  - Combine several executables
  - Require various hardware and software support
- **Characterize workload**
  - Predict whether job will fail
  - Move job to different resources as needed for efficient processing

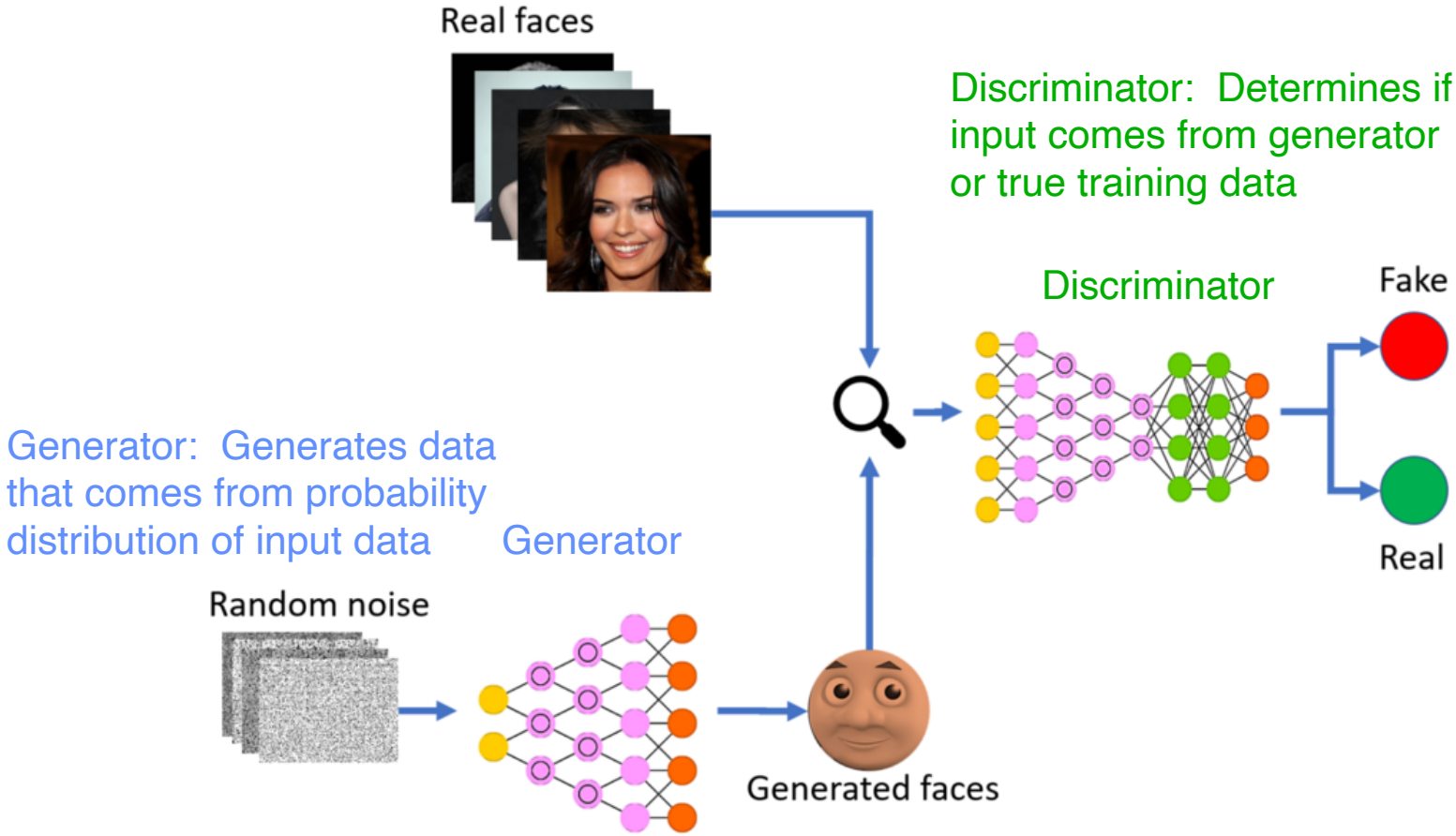
LSTM Applications

- Speech recognition
- Machine translation
- Language modeling
- Speech synthesis
- Handwriting recognition
- Text generation
- Video analysis
- Protein structure prediction
- Stock price prediction

Generative Adversarial Networks (GANs)

- **Deep learning approach to generative modeling**
- **Allows for model to generate data**
  - Model learns structure of input data to generate new data with similar characteristics as input data
- **Consists of two models**
  - Generator: Generates new samples
  - Discriminator: Determines if sample is generated (fake) or from input data (real)
  - Trained in an adversarial way

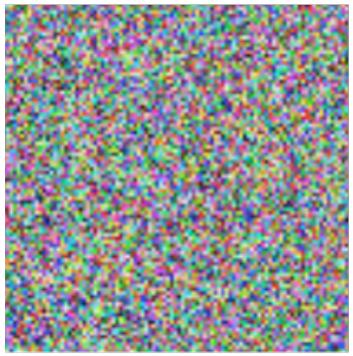
GAN Architecture



https://medium.com/sigmoid/a-brief-introduction-to-gans-and-how-to-code-them-2620ee465c30

GAN Applications

Noise ~ N(0,1)

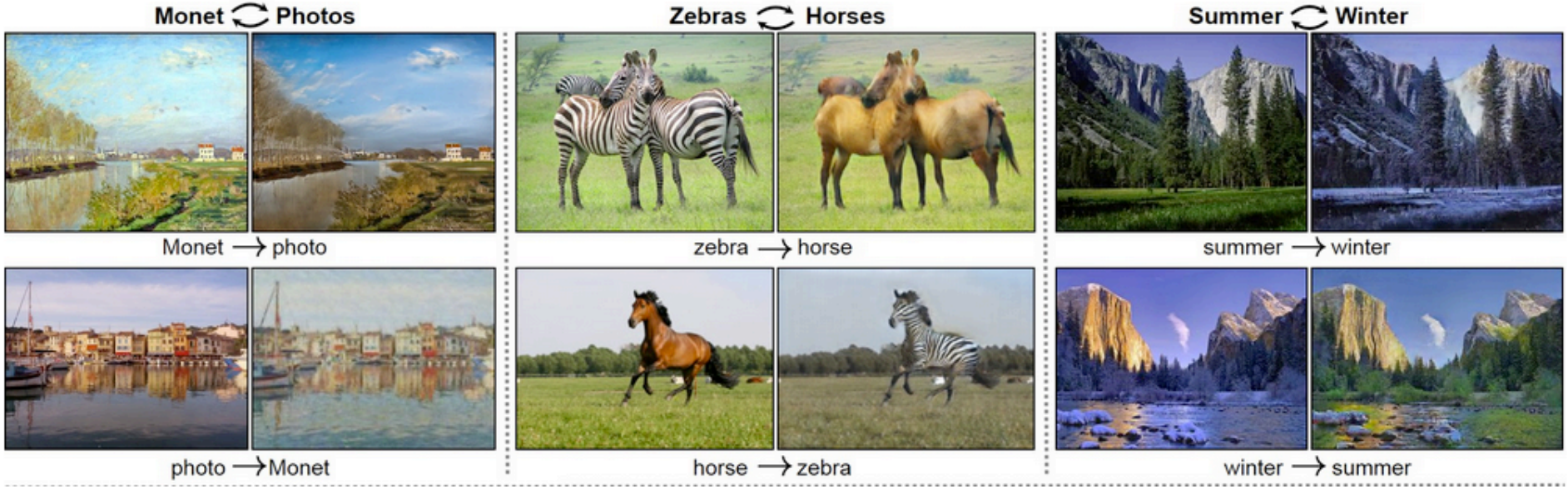


Generative Model

https://arxiv.org/pdf/1710.10196.pdf

GAN Applications

- **Image-to-Image Translation**
  - Transform images from one domain (e.g., real scenery) to another domain (Monet paintings)



https://junyanz.github.io/CycleGAN/

GAN Applications

- **Superresolution**
  - Create high-resolution images from lower-resolution images

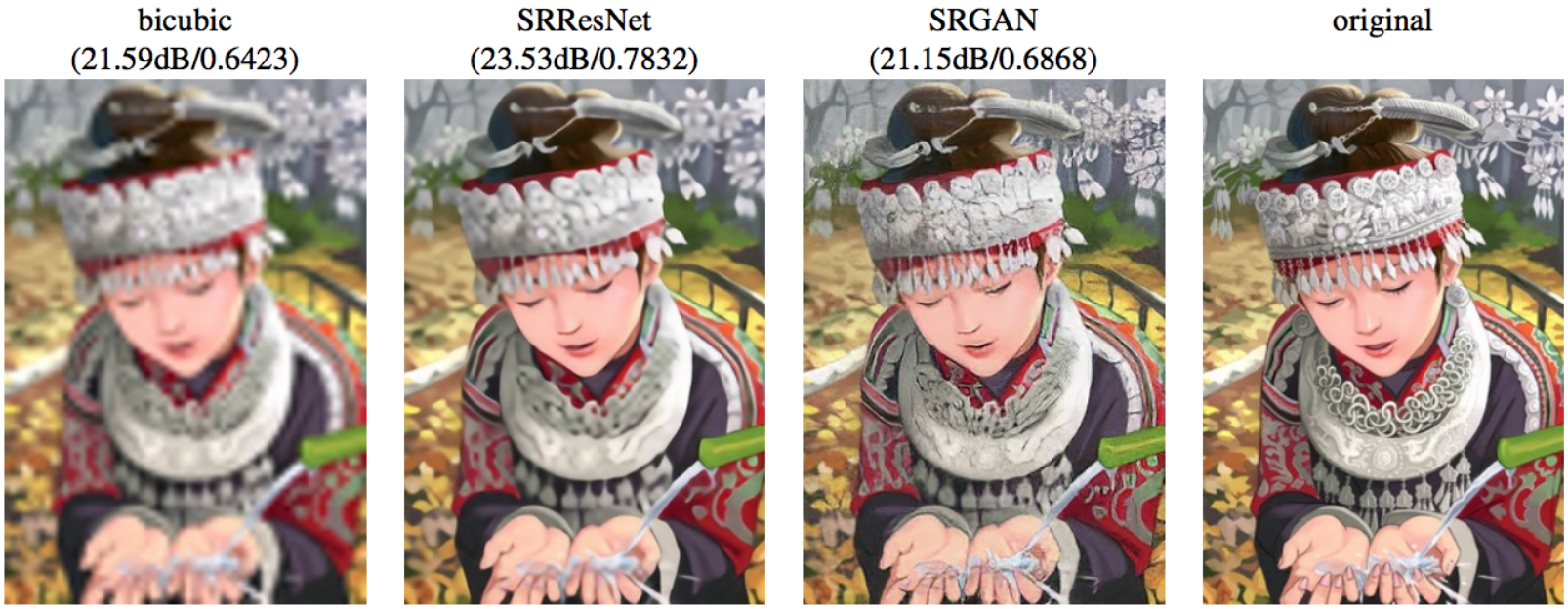


Figure 2: From left to right: bicubic interpolation, deep residual network optimized for MSE, deep residual generative adversarial network optimized for a loss more sensitive to human perception, original HR image. Corresponding PSNR and SSIM are shown in brackets. [4× upscaling]

https://arxiv.org/pdf/1609.04802.pdf

GAN Applications

- **Others**
  - Text-to-image translation
  - Face view generation
  - Pose generation
  - Photos to emojis
  - Face aging
  - …

Python Deep Learning Libraries

- **TensorFlow**
  - https://www.tensorflow.org/
  - ML framework developed by Google
- **Keras**
  - https://keras.io/
  - High-level NN API. Runs on TensorFlow, CNTK, or Theano
- **PyTorch**
  - https://pytorch.org/
  - ML framework developed by Facebook
- **Caffe & Caffe2**
  - https://caffe2.ai/docs/caffe-migration.html
  - Caffe2 now merged into PyTorch
- **Apache MXNet**
  - https://mxnet.apache.org/
  - DL framework used by AWS

Other Deep Learning Libraries

- **Java**
  - Deeplearning4j
- **R**
  - TensorFlow, Keras, MXNet
- **Cloud**
  - Google Cloud ML
  - AWS SageMaker
  - Microsoft Azure
  - IBM Watson ML

References

- **U-Net**
  - Original paper:
    - https://arxiv.org/abs/1505.04597
  - U-Net & Keras
    - https://spark-in.me/post/unet-adventures-part-one-getting-acquainted-with-unet
  - U-Net for medical image segmentation
    - https://towardsdatascience.com/medical-image-segmentation-part-1-unet-convolutional-networks-with-interactive-code-70f0f17f46c6
  - Satellite image analysis use case:
    - https://ieeexplore.ieee.org/abstract/document/8621883

References

- **LSTM**
  - Original paper
    - https://www.mitpressjournals.org/doi/abs/10.1162/neco.1997.9.8.1735
  - Understanding LSTM Networks
    - http://colah.github.io/posts/2015-08-Understanding-LSTMs/
  - Introduction to LSTM
    - https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/
  - Understanding LSTM
    - https://towardsdatascience.com/understanding-lstm-and-its-quick-implementation-in-keras-for-sentiment-analysis-af410fd85b47

References

- **GAN**
  - Original paper
    - https://arxiv.org/abs/1406.2661
  - Understanding GANs
    - https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/
  - Introduction to GAN
    - https://developers.google.com/machine-learning/gan
  - Applications of GAN
    - https://medium.com/@jonathan_hui/gan-some-cool-applications-of-gans-4c9ecca35900

Questions?