Agenda

- 5:30 - Networking
- 6:00 - Talk
- 6:30 - Discussion and Networking

WiFi

- Name: RokkinCat Guest
- Password: makingstuff

Bathroom

- Take a bathroom key and go down a floor and the bathrooms are at the end of the hall

RK
KN

Image Style Transfer

Milwaukee Machine Learning Meetup

Mitchell Henke / @MitchellHenke

About Me/This Meetup

- Software Architect
- Specialize in data, databases, APIs
- Self-taught R, Python/Keras, Machine Learning

Chairs sponsored by:

MKE PRODUCTION RENTAL

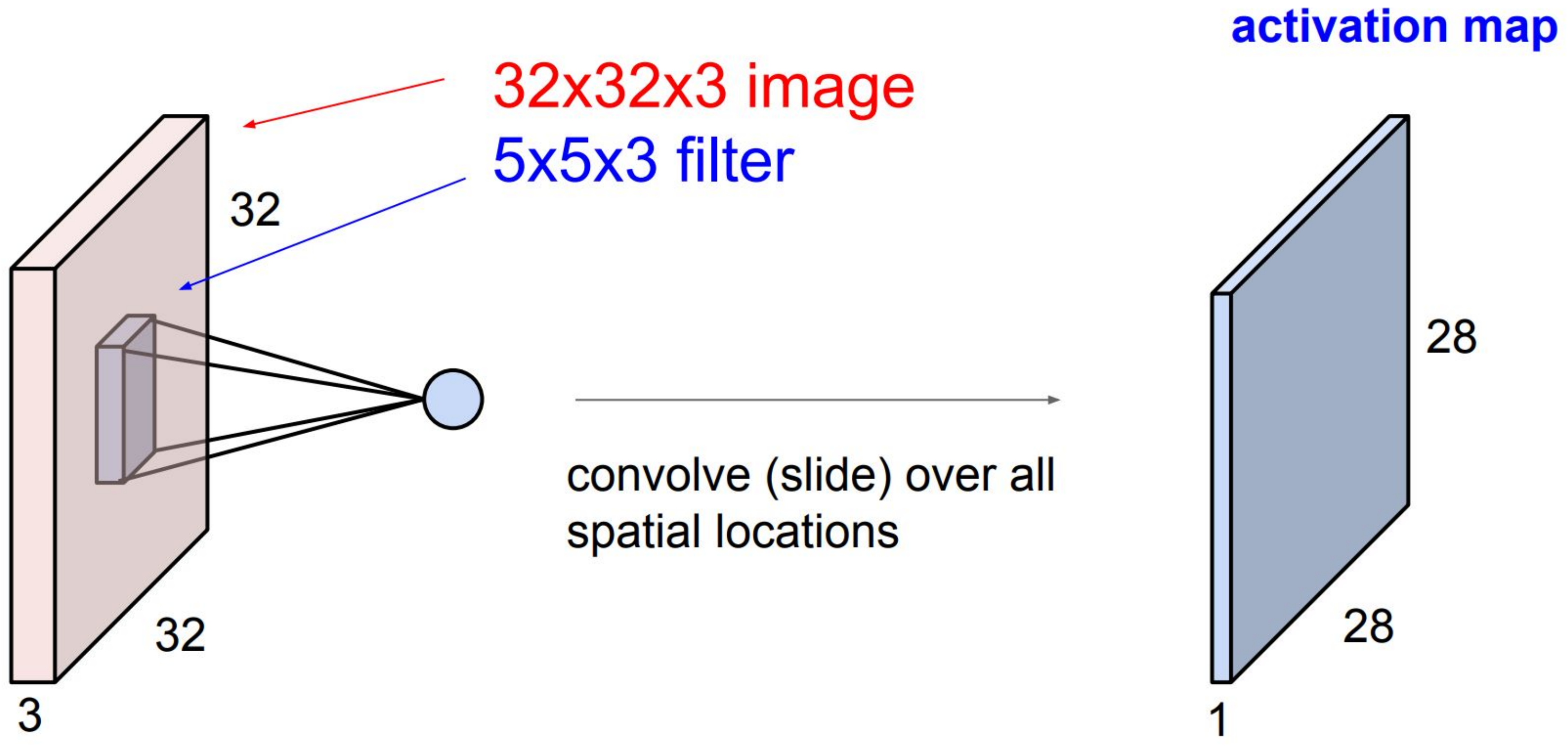
Image Style Transfer

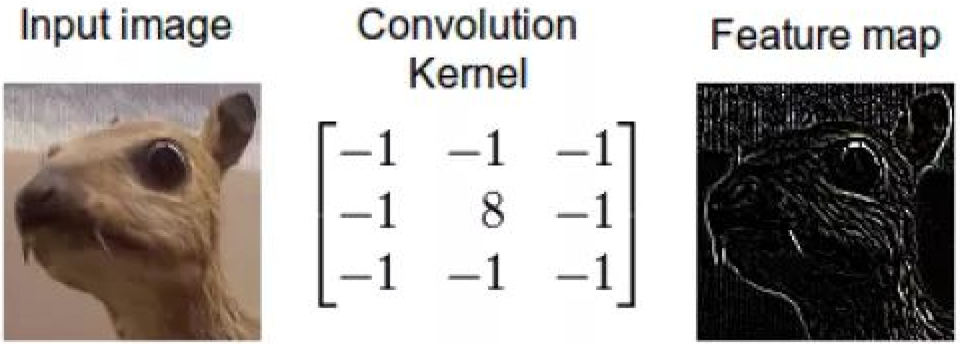
c

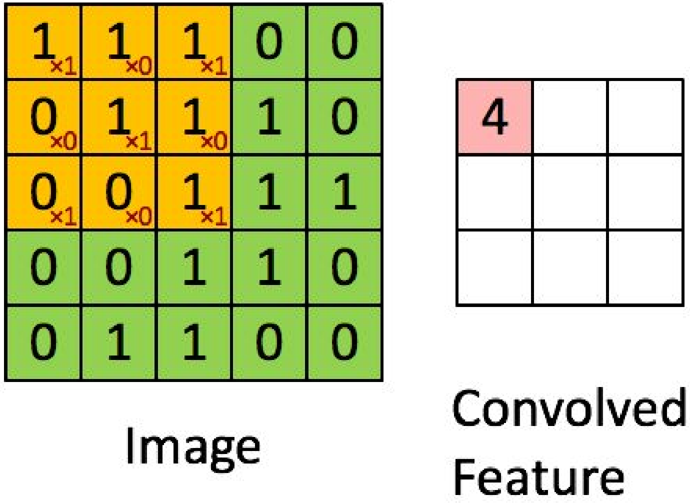
A Neural Algorithm of Artistic Style (Gatys, 2015)

Convolutional Neural Networks

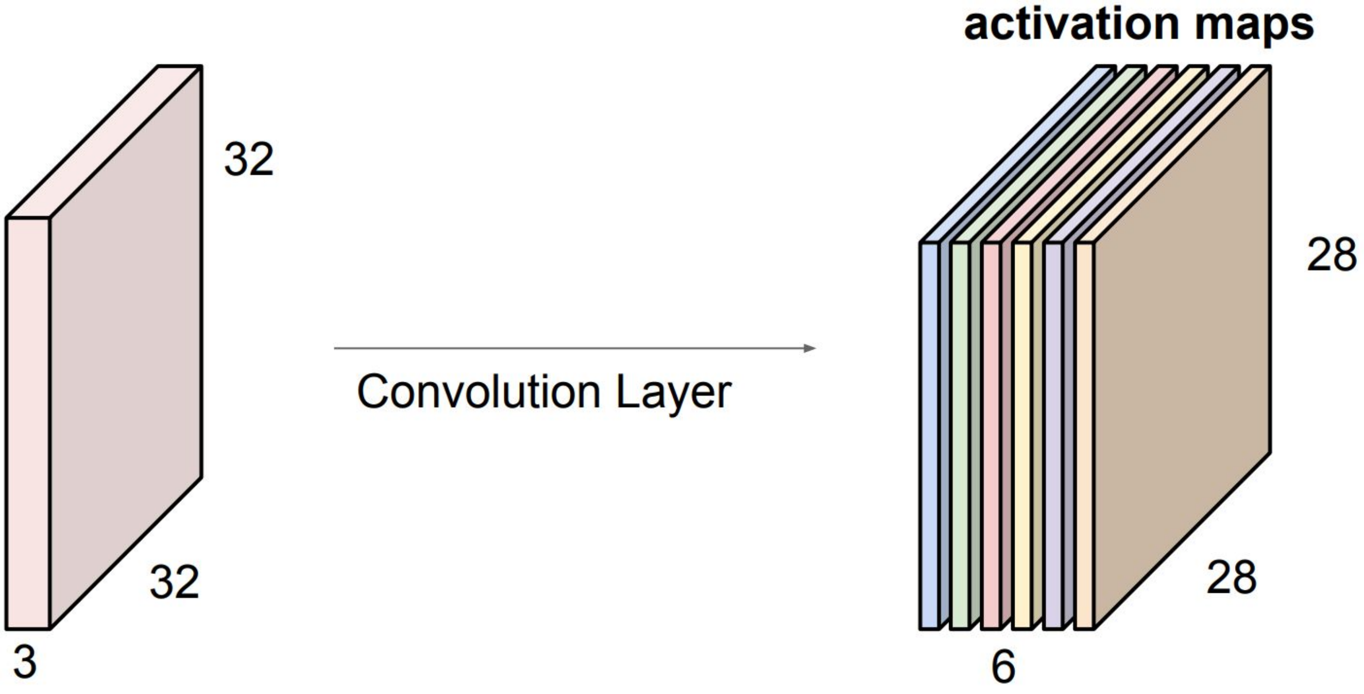
CNN Feature Extraction




Input image
Convolution Kernel
Feature map
$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

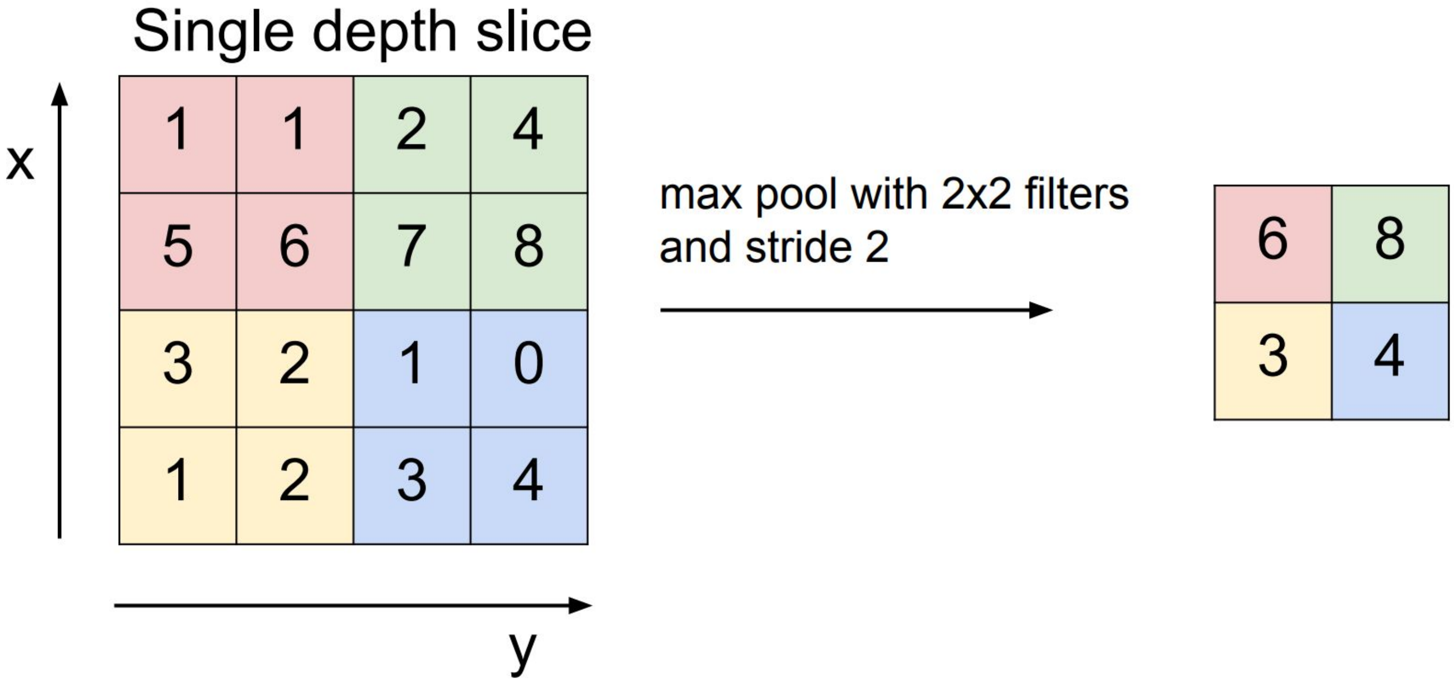

1 ×1, 1 ×0, 1 ×1, 0, 0
0 ×0, 1 ×1, 1 ×0, 1, 0
0 ×1, 0 ×0, 1 ×1, 1, 1
0, 0, 1, 1, 0
0, 1, 1, 0, 0
Image
4
Convolved Feature

CNN Feature Extraction

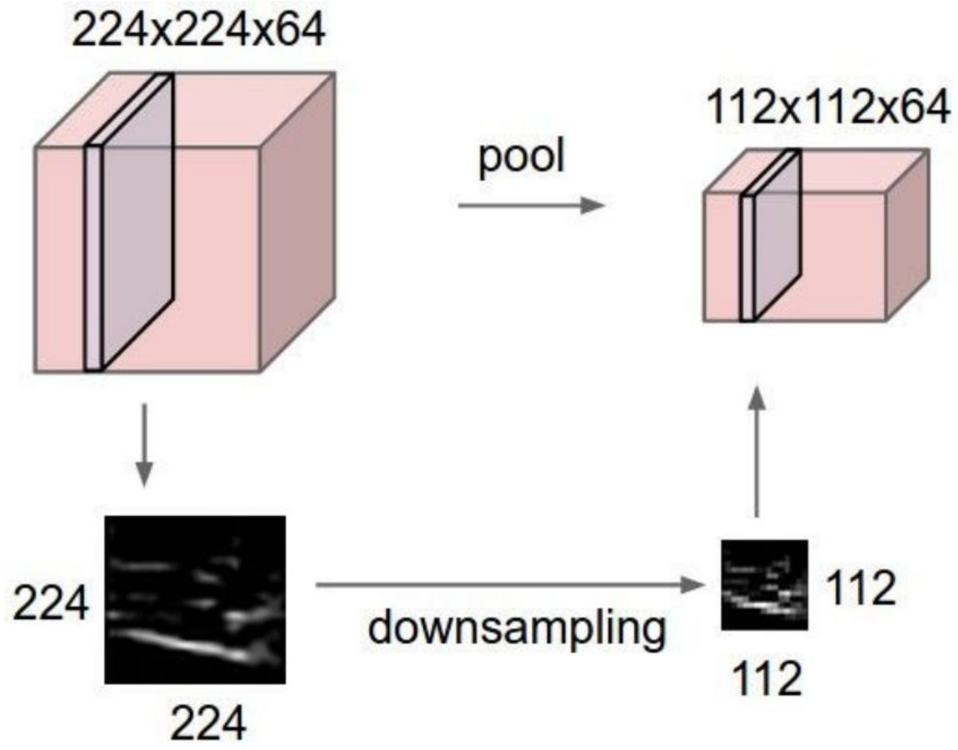


We stack these up to get a “new image” of size 28x28x6!

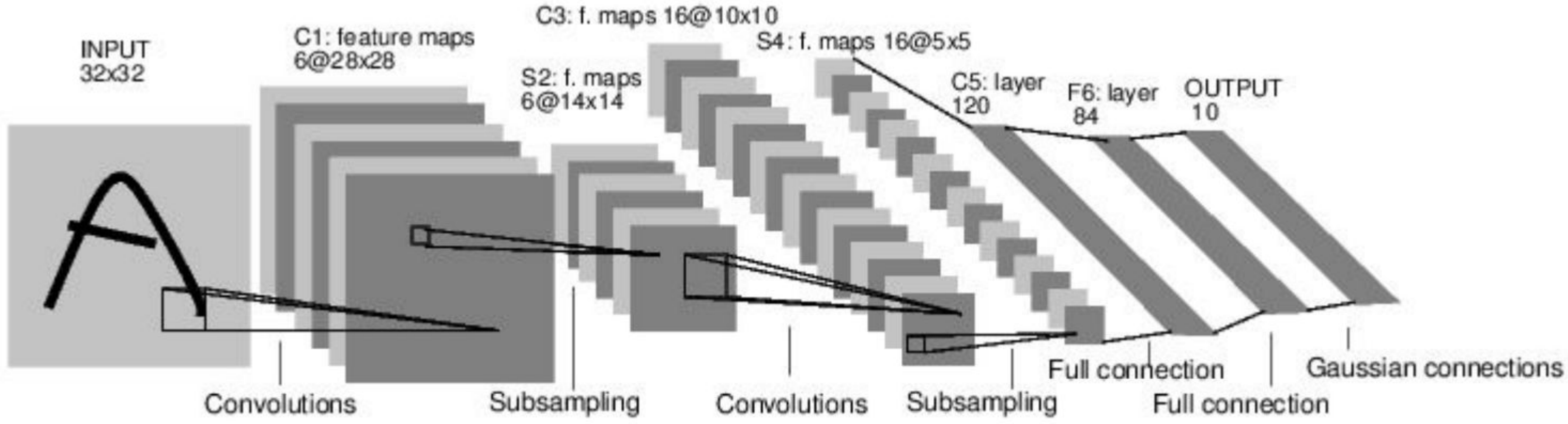
CNN Downsampling



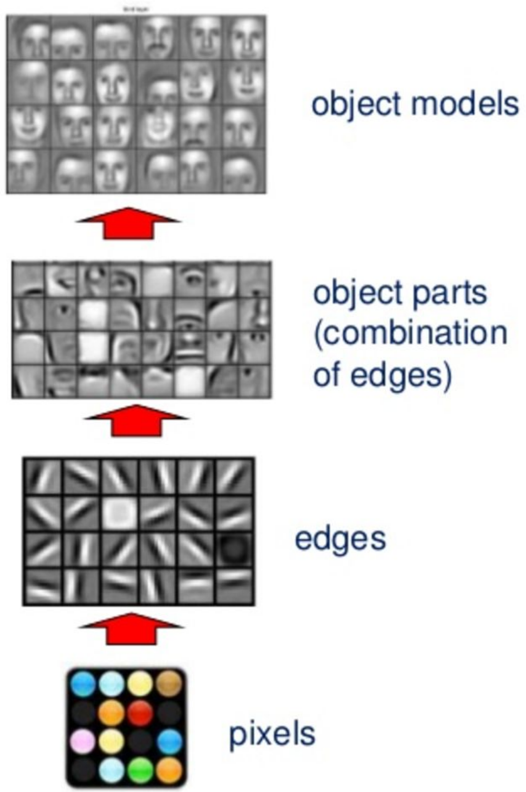
CNN Downsampling



Convolutional Neural Networks



Convolutional Neural Networks



Convolutional Neural Networks

- Excellent pre-trained CNN image models are published and shared
  - VGGNet (2014)
  - ResNet (2015)
- Transfer Learning
  - Pre-trained model can be fine-tuned for specific problem

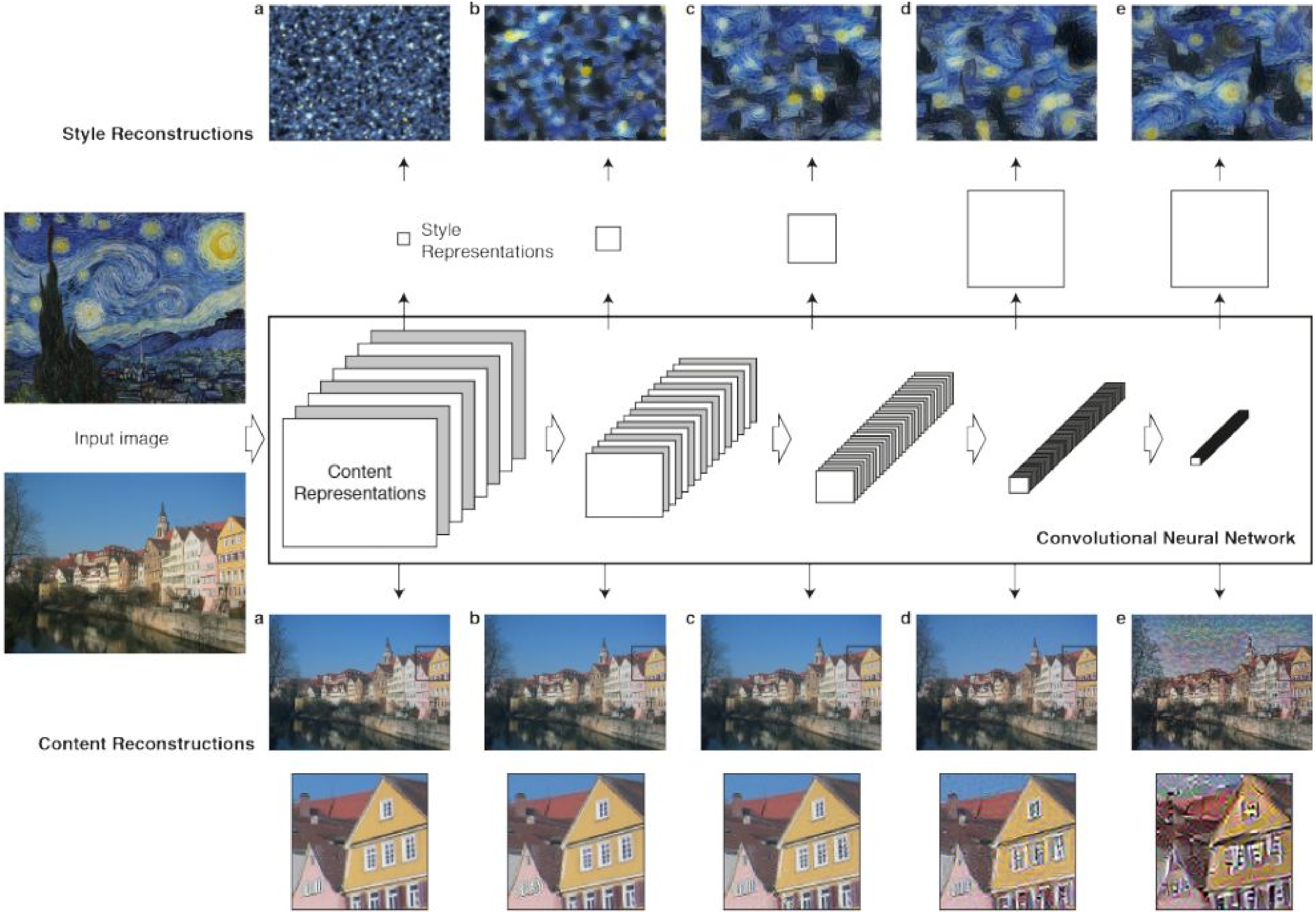
Loss Functions

Loss Functions

- Defines how close your model is to producing expected output
  - Mean Absolute Error
  - Mean Squared Error
  - Categorical/Binary Cross Entropy
  - ???
- Function of many factors
  - Factors can be weighted differently
  - Ultimately has to be a single number
- Network optimizes layer parameters from this output

A Neural Algorithm of Artistic Style (Gatys, 2015)

- Uses frozen pre-trained VGG-19 model to recognize features
- Content features
  - High level features like buildings, people, or animals
- Style features
  - Low level features (colors, textures)
  - Higher level features (multiple yellow stars on top of blue)



Style Reconstructions
a
b
c
d
e
Style
Representations
Input image
Content
Representations
Convolutional Neural Network
Content Reconstructions
a
b
c
d
e

Overview of A Neural Algorithm of Artistic Style

- Save style features from style image
- Save content features from content image
- Begin with a randomized white noise image
- Compare style features in white noise image to saved style features
- Compare content features in white noise image to saved content features
- Neural net modifies white noise image pixels to get closer to matching both style and content features

Style Transfer Loss Function

- Pass style image and white noise image through 5 of VGG's convolutional layers
  - VGG output captures a range of features from simple to complex
  - Compare difference in activations
- Pass content image and white noise image through 1 of VGG's later layers
  - VGG output only captures more complex features
  - Compare difference in activations
- Train the initially white noise pixels to minimize the difference in style and content

$$\text{Loss} = (\text{Difference in style features}) + (\text{Difference in content features})$$

A Neural Algorithm of Artistic Style

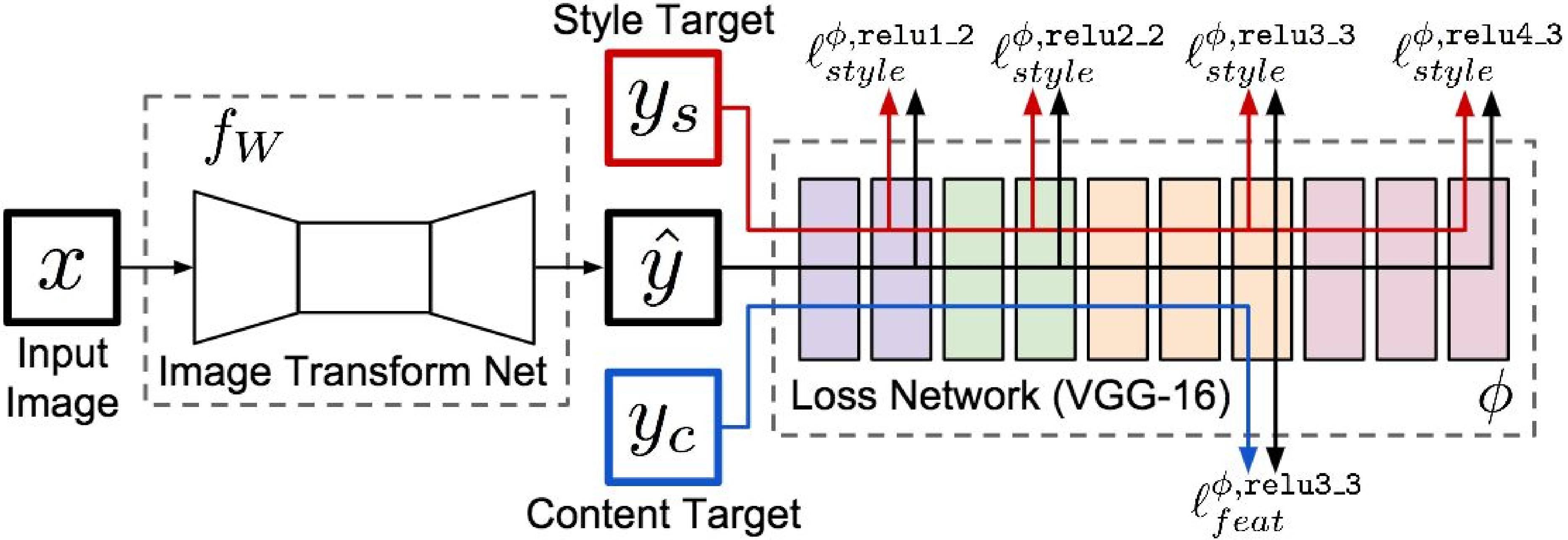
- Loss is calculated with respect to generated image pixels
- Training produces a single image, not a model
- Each new image or style requires re-training

Perceptual Losses for Real-Time Style Transfer and Super-Resolution (Johnson, 2016)

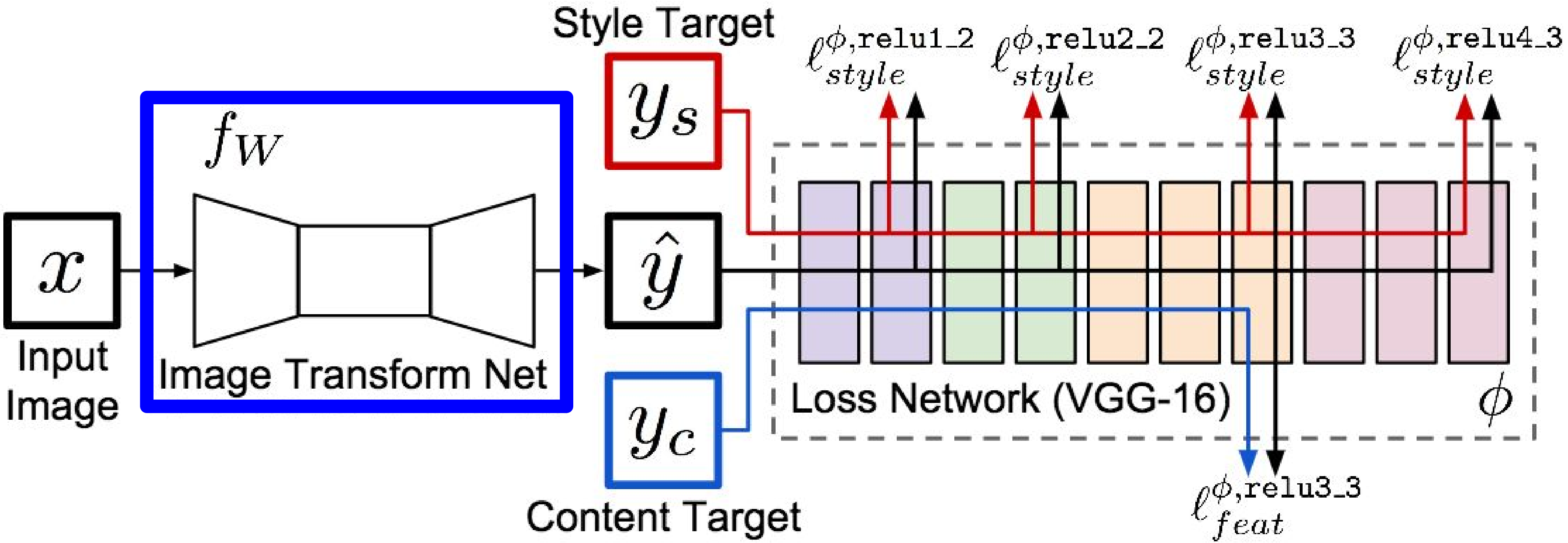
- Similar techniques to *A Neural Algorithm of Artistic Style*
- Builds a network that can produce images in a single style without retraining
- Also uses similar concepts to do “super-resolution” or enhancement of low resolution images

Style
The Starry Night,
Vincent van Gogh,
1889

Style
The Muse,
Pablo Picasso,
1935



Style Target
y_s
$\ell_{style}^{\phi,\texttt{relu1\_2}}$
$\ell_{style}^{\phi,\texttt{relu2\_2}}$
$\ell_{style}^{\phi,\texttt{relu3\_3}}$
$\ell_{style}^{\phi,\texttt{relu4\_3}}$
f_W
x
Input Image
Image Transform Net
$\hat{y}$
y_c
Content Target
Loss Network (VGG-16)
ϕ
$\ell_{feat}^{\phi,\texttt{relu3\_3}}$


Style Target
$\ell_{style}^{\phi,\mathtt{relu1\_2}}$
$\ell_{style}^{\phi,\mathtt{relu2\_2}}$
$\ell_{style}^{\phi,\mathtt{relu3\_3}}$
$\ell_{style}^{\phi,\mathtt{relu4\_3}}$
f_W
y_s
x
$\hat{y}$
Input Image
Image Transform Net
y_c
Loss Network (VGG-16)
ϕ
Content Target
$\ell_{feat}^{\phi,\mathtt{relu3\_3}}$

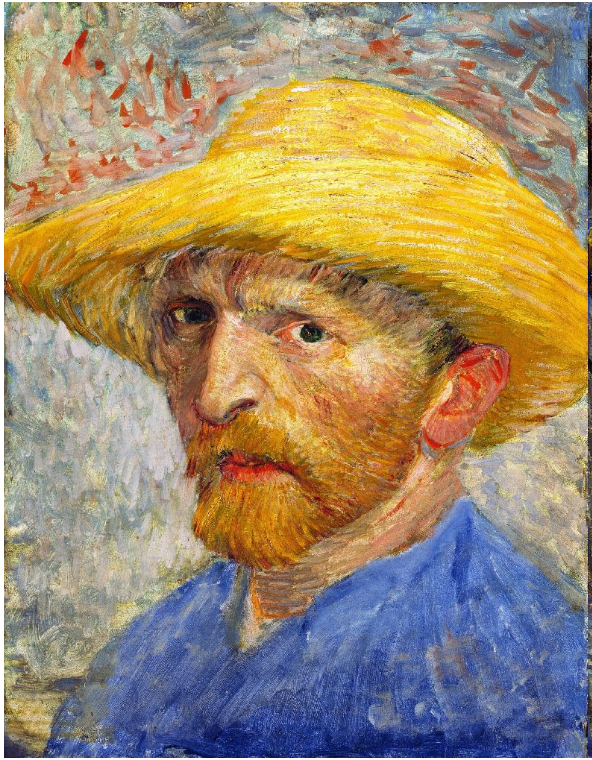
Architecture

Layer	**Activation size**
Input	$3 \times 256 \times 256$
Reflection Padding (40×40)	$3 \times 336 \times 336$
$32 \times 9 \times 9$ conv, stride 1	$32 \times 336 \times 336$
$64 \times 3 \times 3$ conv, stride 2	$64 \times 168 \times 168$
$128 \times 3 \times 3$ conv, stride 2	$128 \times 84 \times 84$
Residual block, 128 filters	$128 \times 80 \times 80$
Residual block, 128 filters	$128 \times 76 \times 76$
Residual block, 128 filters	$128 \times 72 \times 72$
Residual block, 128 filters	$128 \times 68 \times 68$
Residual block, 128 filters	$128 \times 64 \times 64$
$64 \times 3 \times 3$ conv, stride $1/2$	$64 \times 128 \times 128$
$32 \times 3 \times 3$ conv, stride $1/2$	$32 \times 256 \times 256$
$3 \times 9 \times 9$ conv, stride 1	$3 \times 256 \times 256$

Examples

Examples

Examples



Thanks!

References:

- Stanford CS231
- A Neural Algorithm of Artistic Style, Gatys et al. (2015)
- Perceptual Losses for Real-Time Style Transfer and Super-Resolution, Johnson et al. (2016)
- Deepart.io
- pikazoapp.com

Next Month:

Multi-class Tagging of Notes with Naive Bayes by Rob Hoelz

August 22nd, 2017