Introduction to Deep Learning

15. Image Augmentation, Fine Tuning, Style Transfer

STAT 157, Spring 2019, UC Berkeley

Alex Smola and Mu Li

courses.d2l.ai/berkeley-stat-157

Image Augmentation

p=1.0

size_percent=0.30

p=0.50

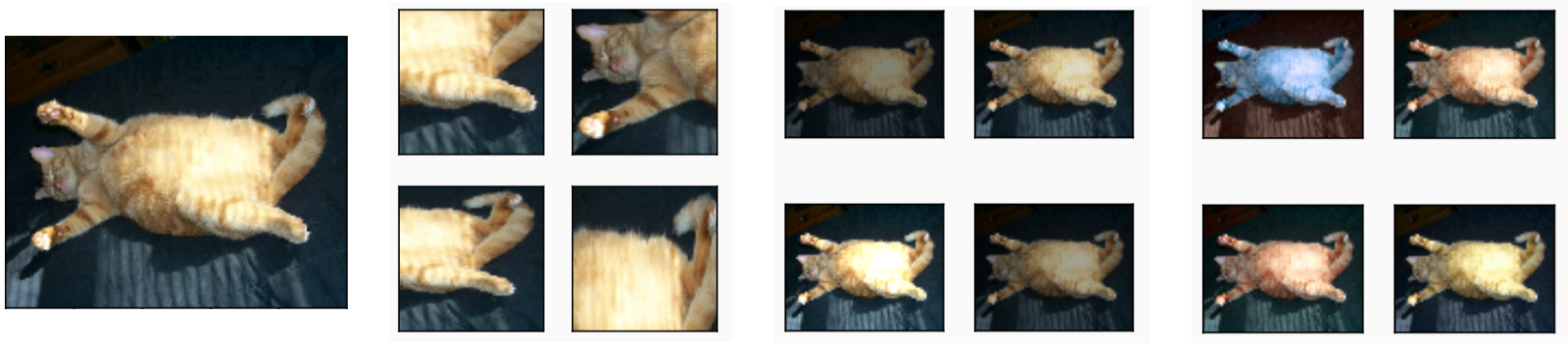
cutoff=0.00

A Real Story at CES'19

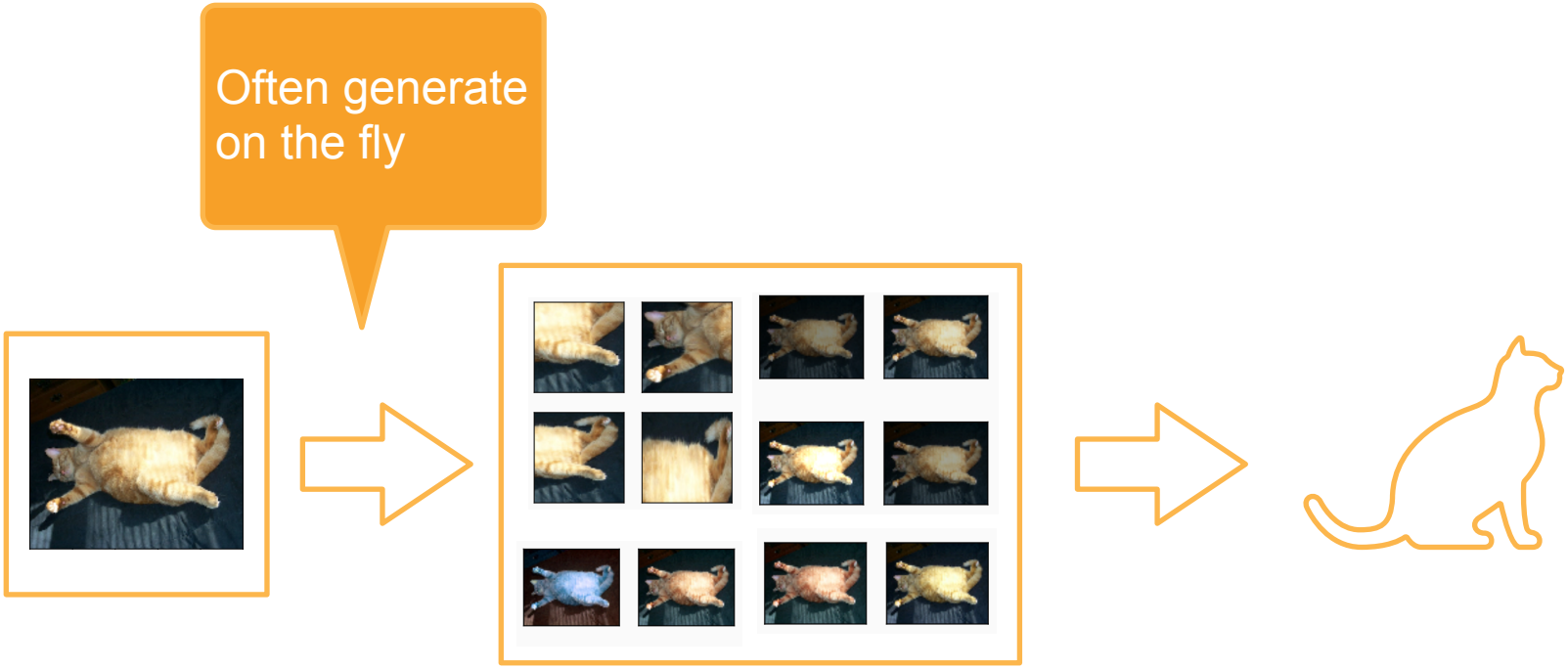
- A startup found their demo, a smart vending machine that identifies what customers picked through a camera, didn't work because the showroom has
  - a different light temperature
  - light reflection from the desk
- They worked all night to re-collect data and train a new model, and ordered a tablecloth

Data Augmentation

- Augment an existing dataset with more diversities
- Add various background noises into a speech
- Transform an image to several others by altering colors or/and changing shapes

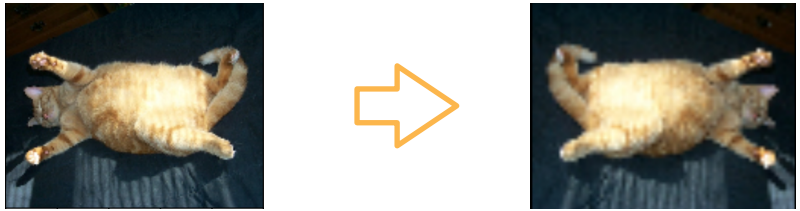


Training with Augmented Data

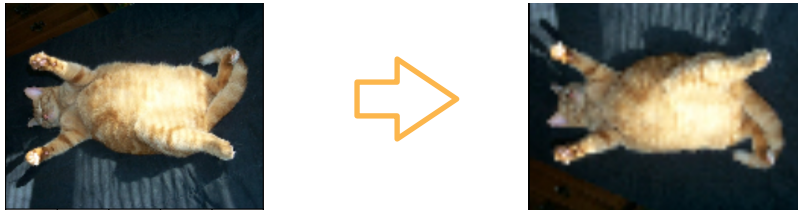


Flip

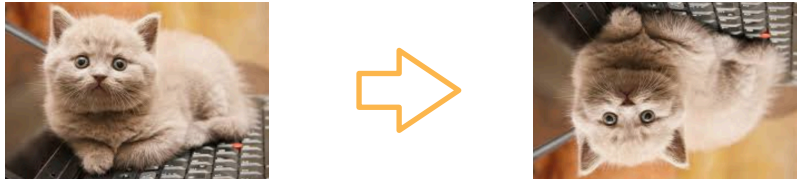
- Left-right flip



- Top-bottom flip

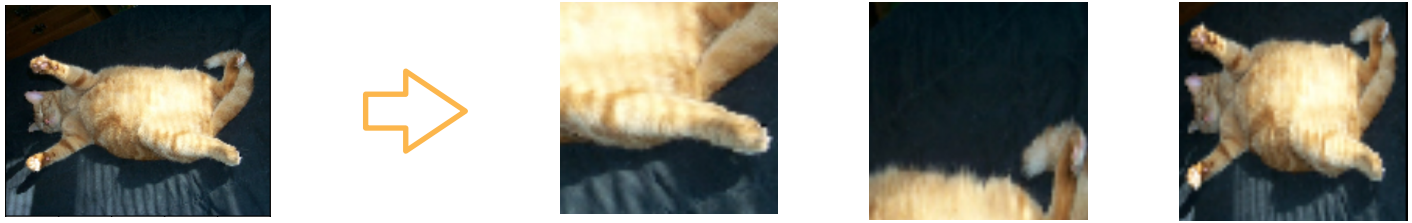


  - Not always makes sense



Crop

- Crop an area from the image and then resize it
  - A random width-height ratio (e.g. [3/4, 4/3])
  - A random area size (e.g. [8%, 100%])
  - A random position



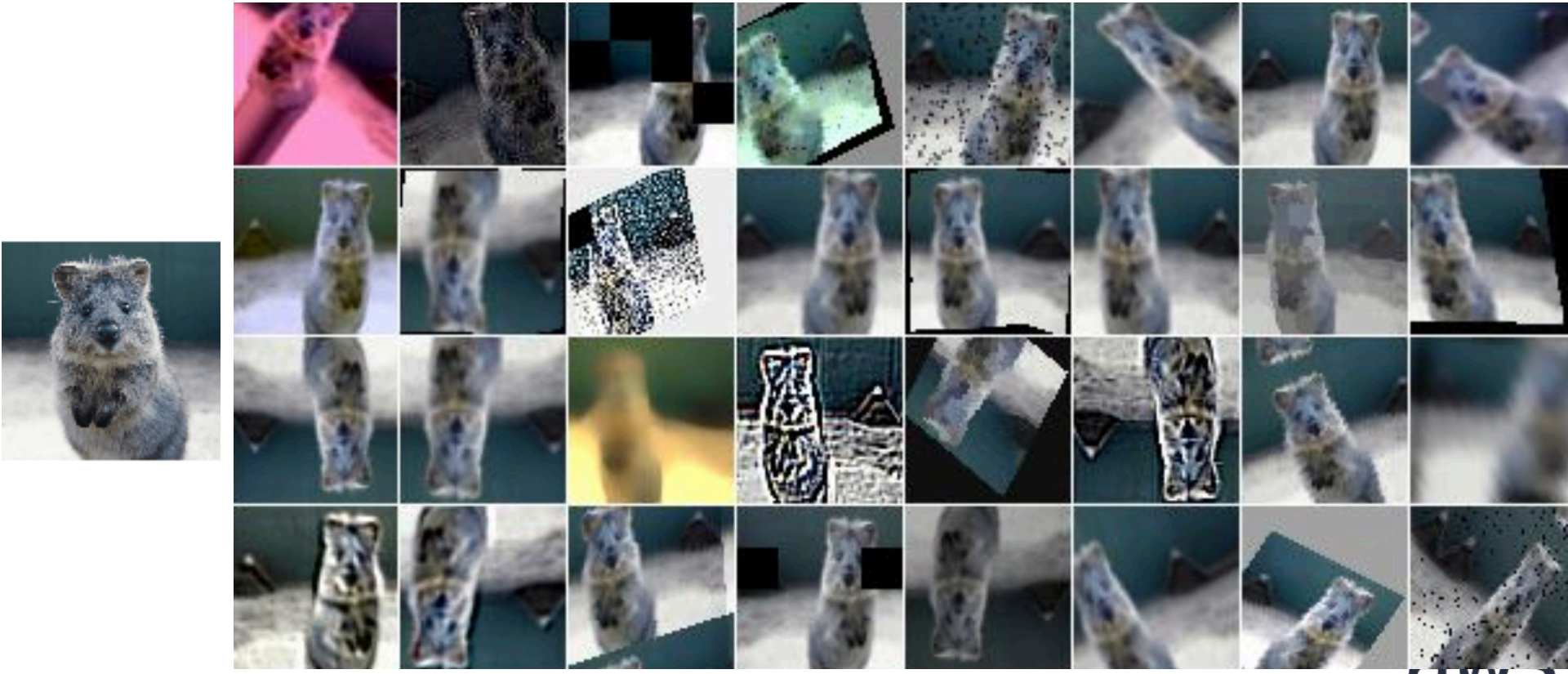
Color

- Scale hue, saturation, and brightness (e.g. [0.5, 1.5])

Brightness

Hue

Tens of Other Ways to Augment



https://github.com/aleju/imgaug

Fine Tuning

Labelling a Dataset is Expensive

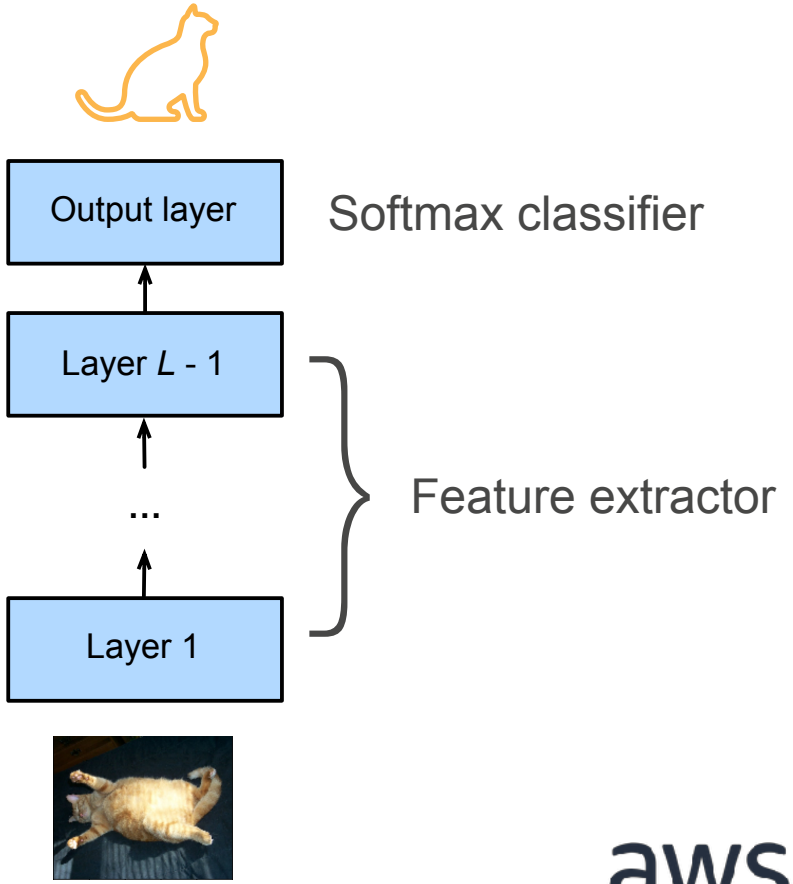
My dataset

# examples	1.2 M	50K	60 K
# classes	1,000	100	10

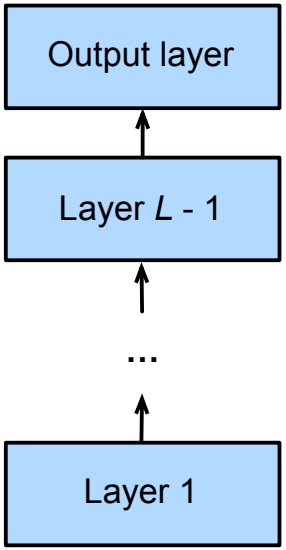
aws

Network Structure

- A neural network can be roughly partitioned into two parts
  - A feature extractor maps raw pixels into linearly separable features
  - A linear classify to make decisions



Fine Tuning



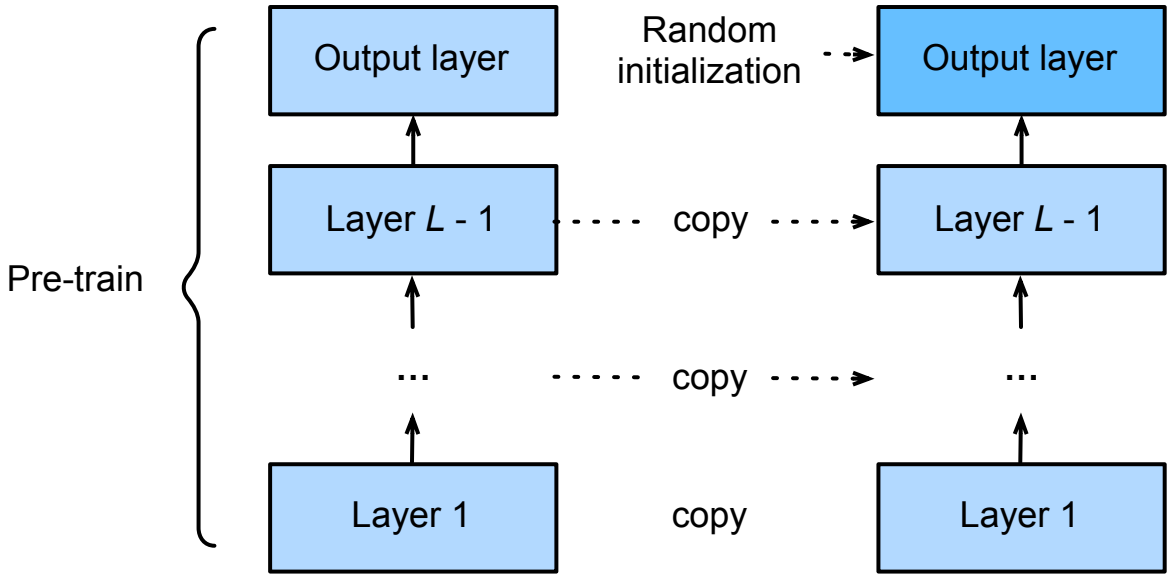
Maybe not use the classifier parameters directly because labels are changed

Maybe also a good extractor for my dataset

Source Dataset

Target Dataset

Weight Initialization for Fine Turning



Source Dataset

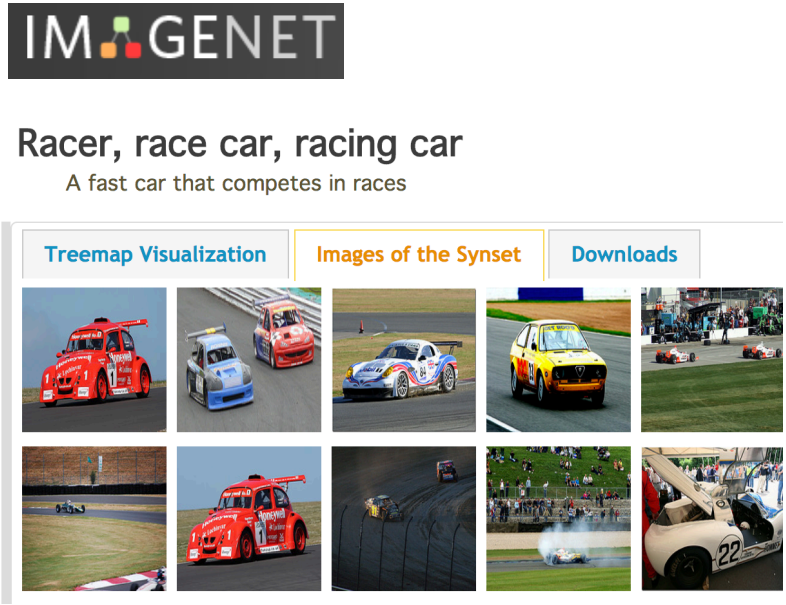
Target Dataset

Training

- Train on the target dataset as a normal training job, but with a strong regularization
  - Uses a small learning rate
  - Uses less epochs
- If source dataset is more complex than the target dataset, fine-tuning often leads to better quality models

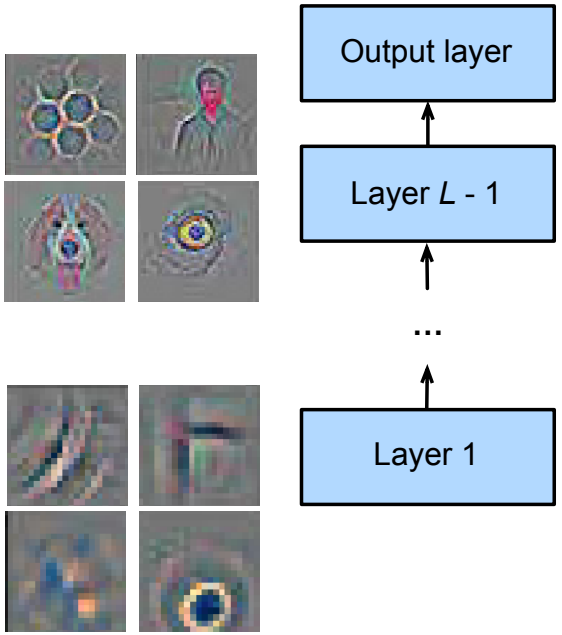
Re-use Classifier Parameters

- The source dataset may contain some categories from the target datasets
- Use the according weight vectors from the pre-trained model during initialization



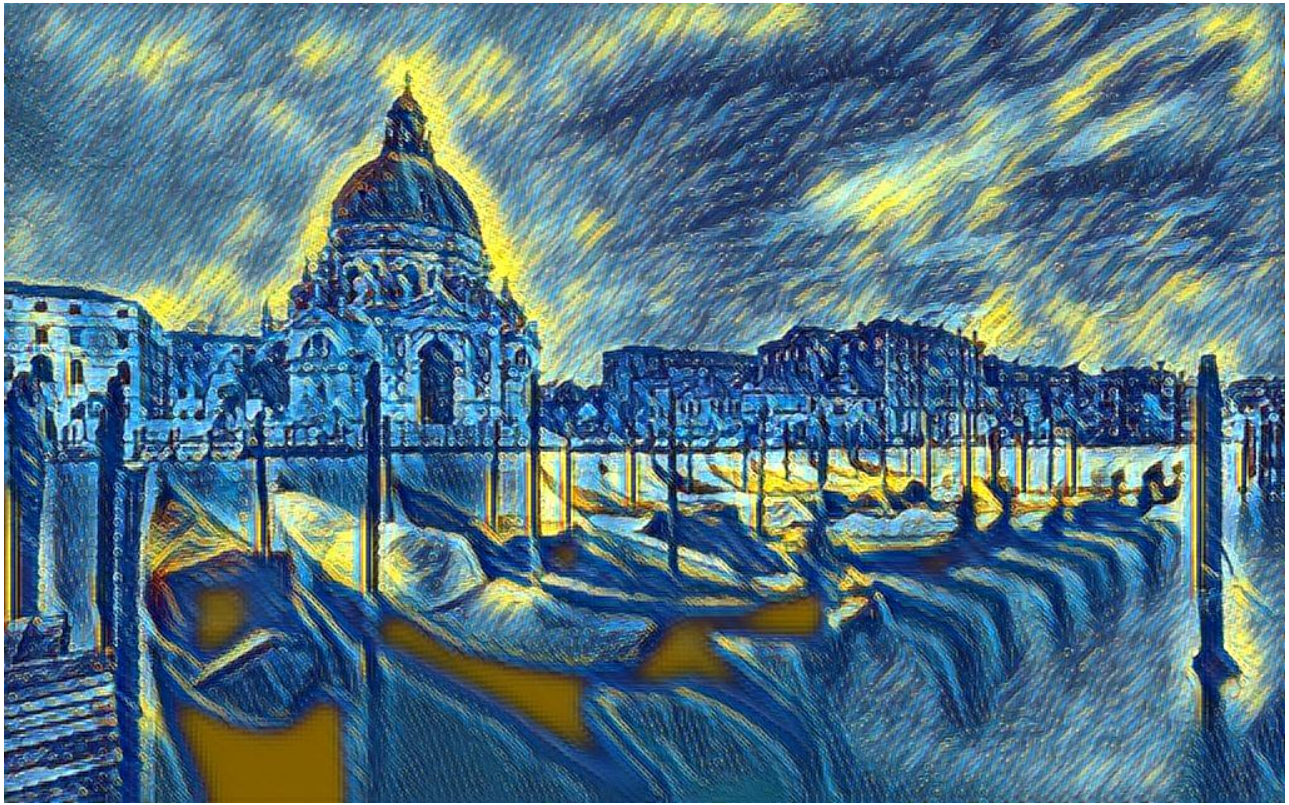
Fix Some Layers

- Neural networks learn hierarchical feature representations
  - Low-level features are universal
  - High-level features are more related to objects in the dataset
- Fix the bottom layer parameters during fine tuning
  - Another strong regularizer



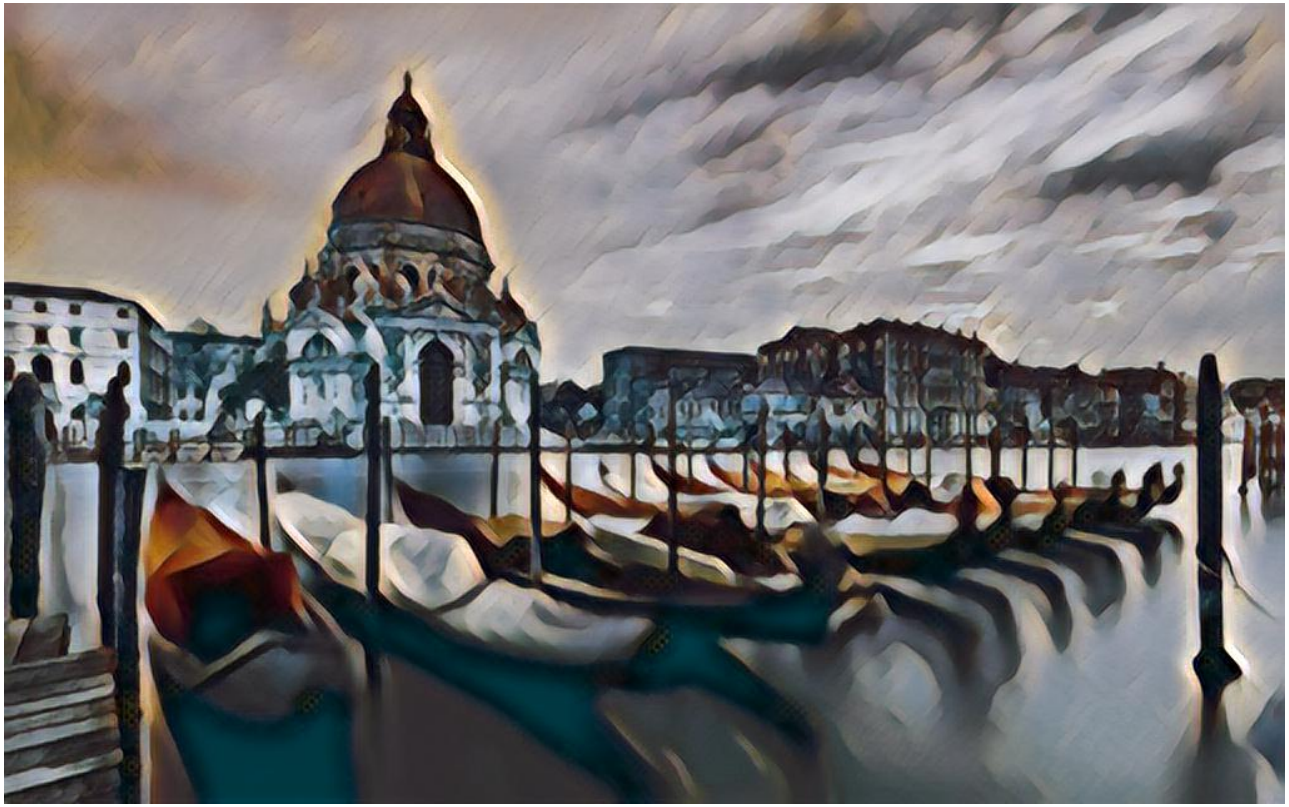
Style Transfer

+

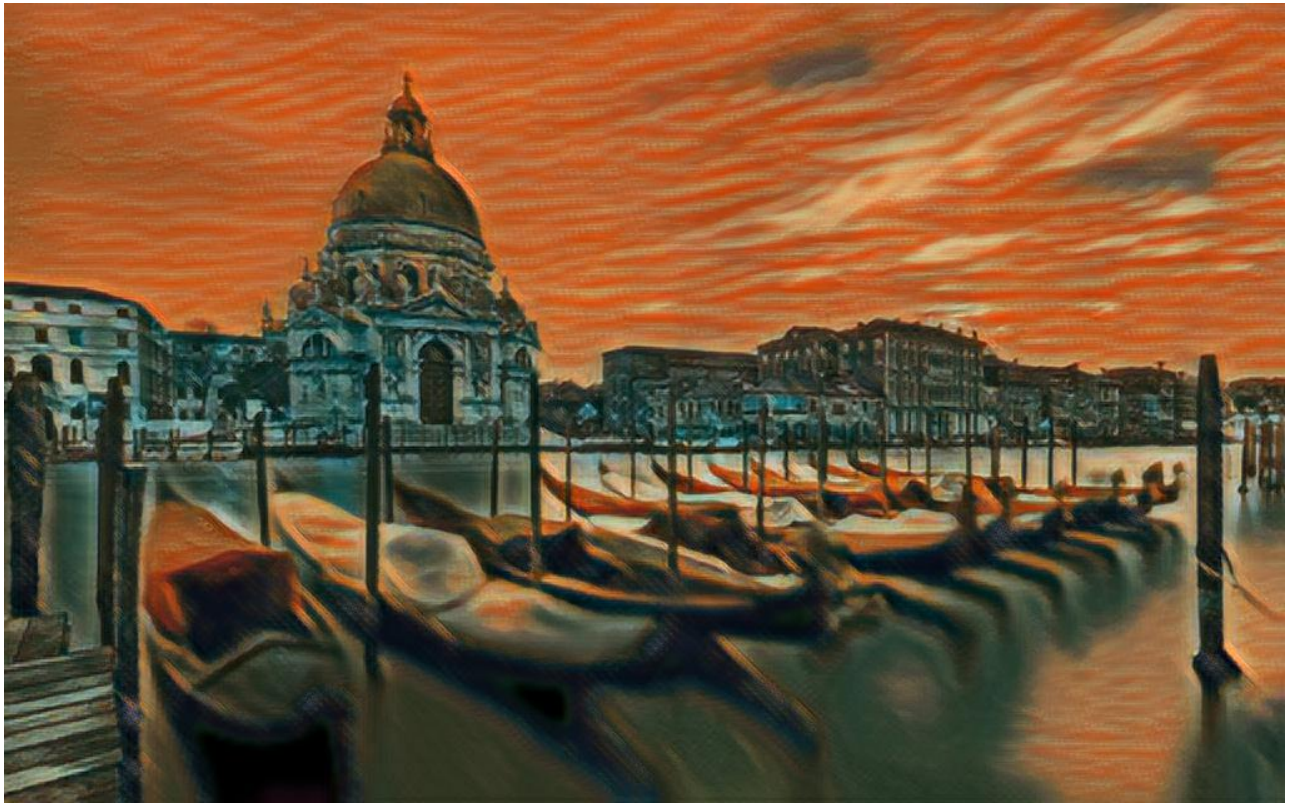


https://github.com/zhanghang1989/MXNet-Gluon-Style-Transfer/

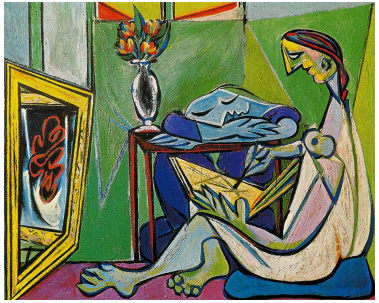
+



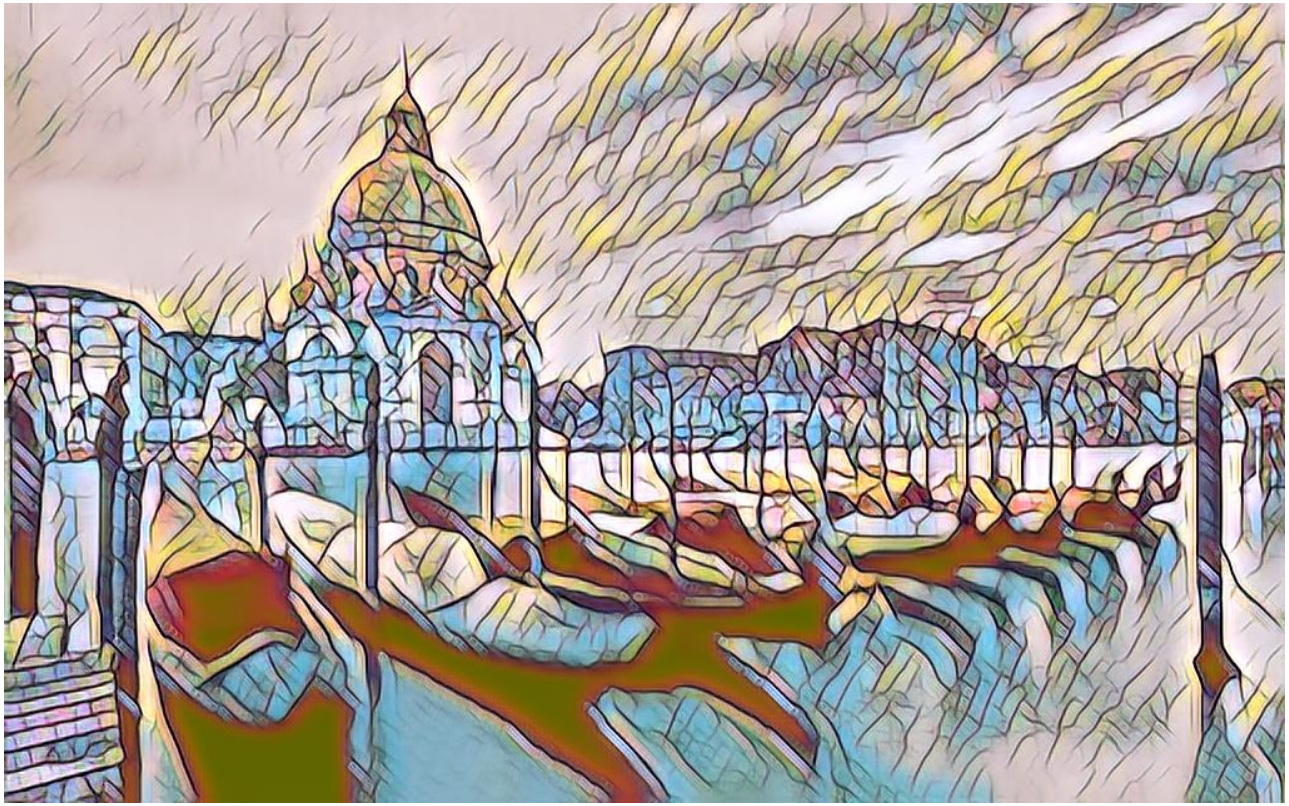
+



+

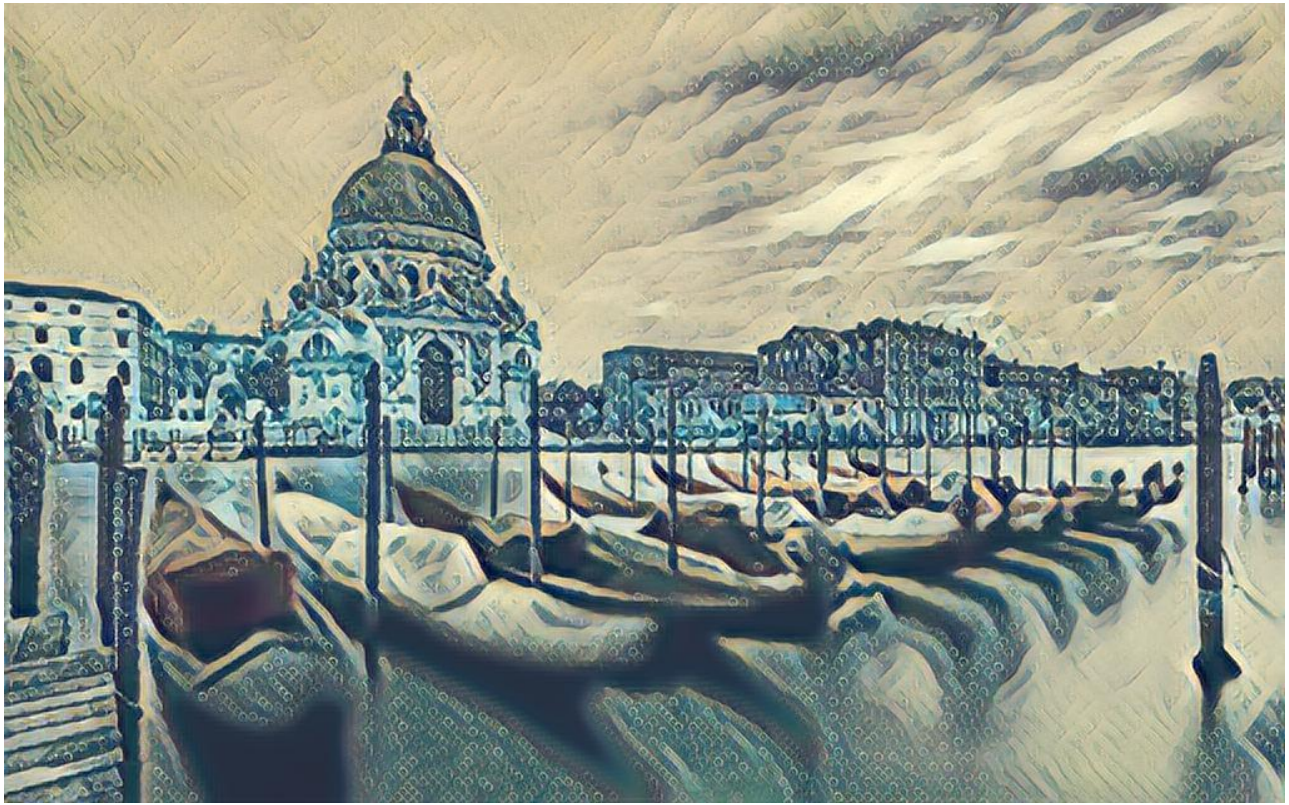


+



+

+



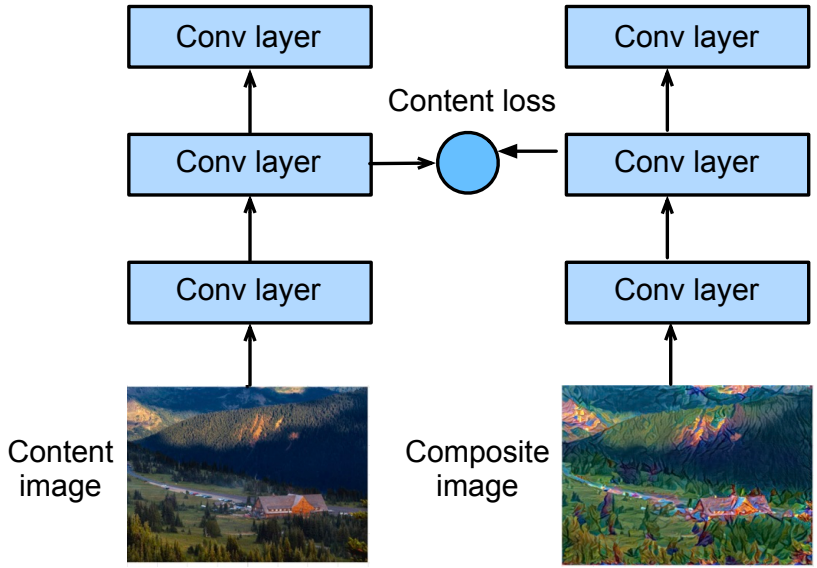
Neural Style

- Learn a composite image to match the contents from a content image and the styles from the style image

$$\arg\min_{I} w_1 \ell_{\text{content}}(I_{\text{content}}, I) + w_2 \ell_{\text{style}}(I_{\text{style}}, I) + w_3 \ell_{\text{noise}}(I)$$

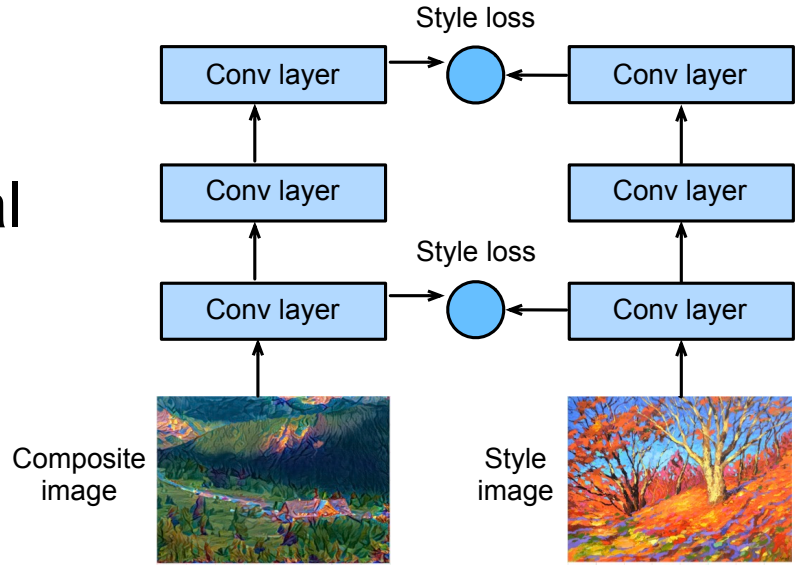
Content Loss

- Feed both content and composite images to the a CNN
- Compare internal layer outputs with a squared loss
  - Bottom layers matches details
  - Top layers matches global contents



Style Loss

- Gram matrix G : $G_{i,j}$ is the inner product between channel *i* and *j*
- Compare gram matrices of internal layer outputs by a squared loss
  - Bottom layers matches local styles
  - Top layers matches global styles



Noise Loss

- The learned composite image may have a lot of high-frequency noise
- Use total variation to de-noise

$$\sum_{i,j} |x_{i,j} - x_{i+1,j}| + |x_{i,j} - x_{i,j+1}|$$

Put All Things Together

$$\arg\min_I w_1 \ell_{\text{content}}(I_{\text{content}}, I) + w_2 \ell_{\text{style}}(I_{\text{style}}, I) + w_3 \ell_{\text{noise}}(I)$$