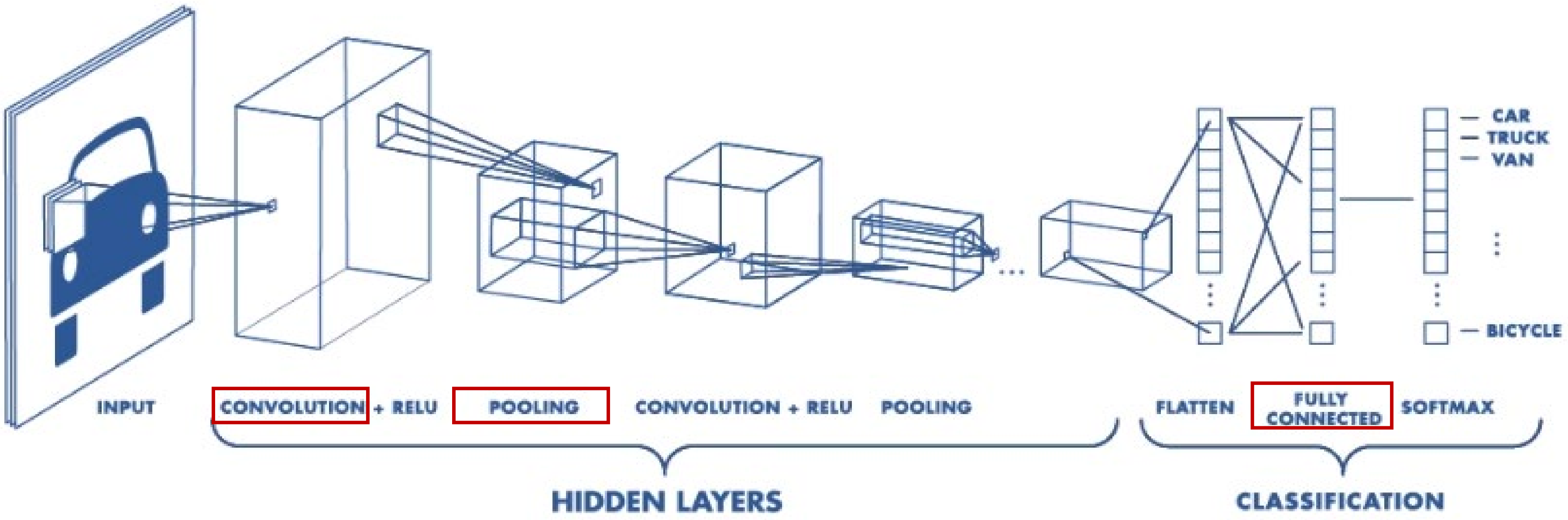
Convolutional Neural Network

TA. Jaewoong Han

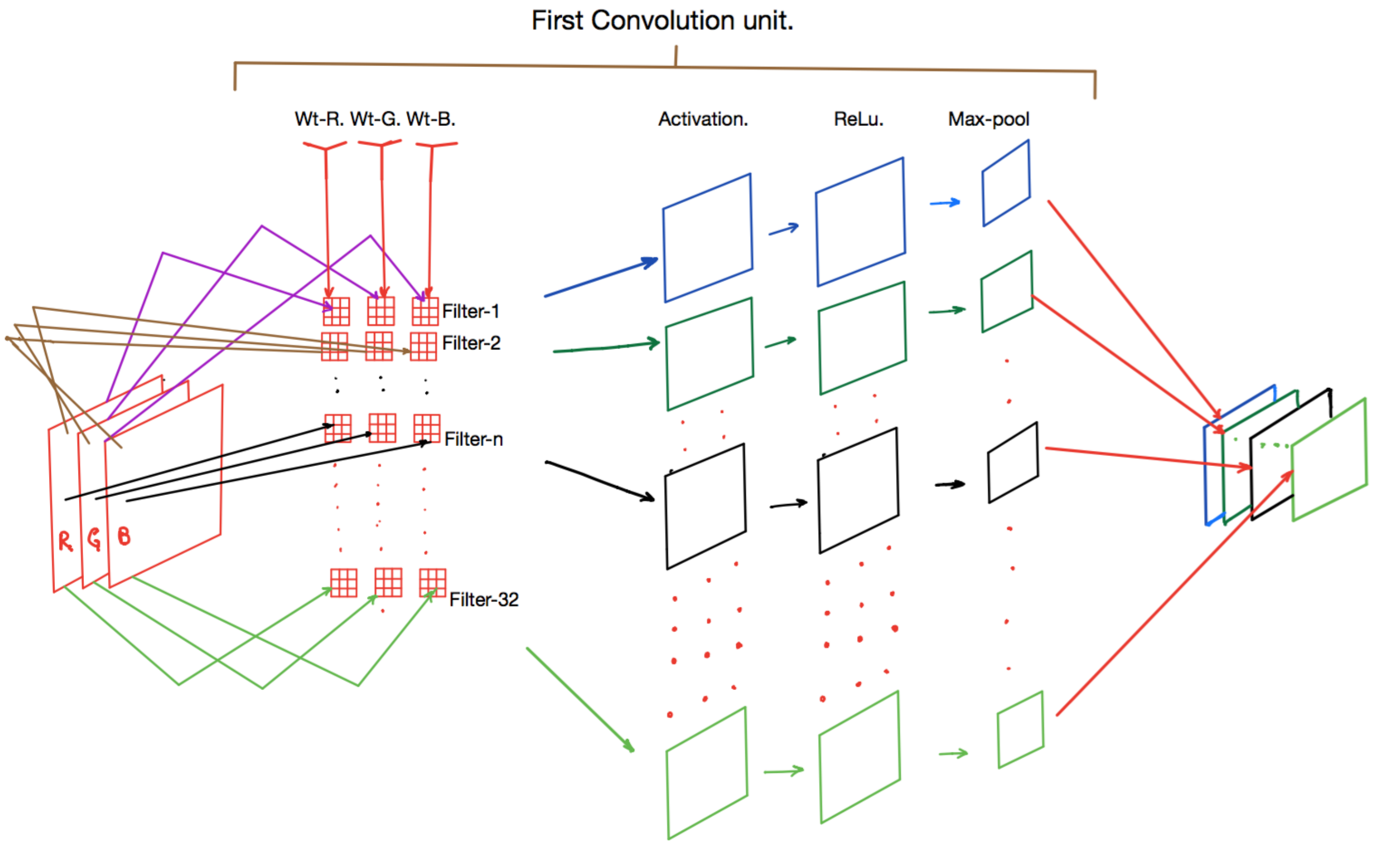
Machine Learning &
Control Systems Lab.

Contents

- Convolutional Neural Network
    - Convolutional layer
    - Convolution operation
    - Pooling layer
- Data Augmentation
- Transfer Learning



- Main Components
  - Convolution (Shared weights, dimension reduction)
  - Nonlinearity (ReLU)
  - Max pooling (translation invariance, dimension reduction for FC layer)
  - Fully connected Layer (classification)



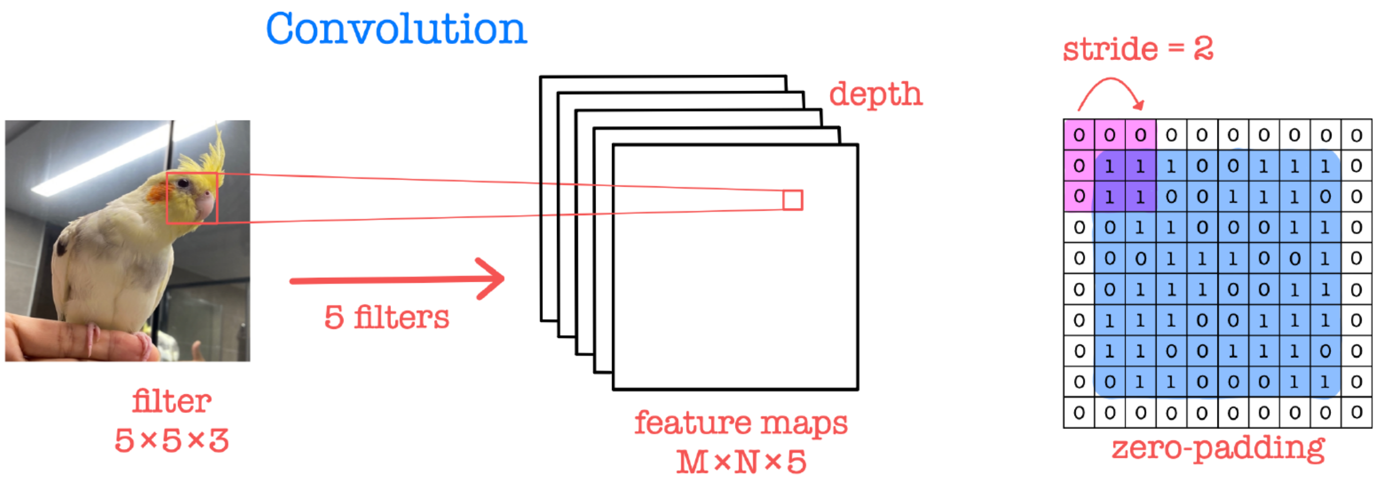
- The weights and the biases which are the components of the filters change when training the network.

1-1. Convolution Layer

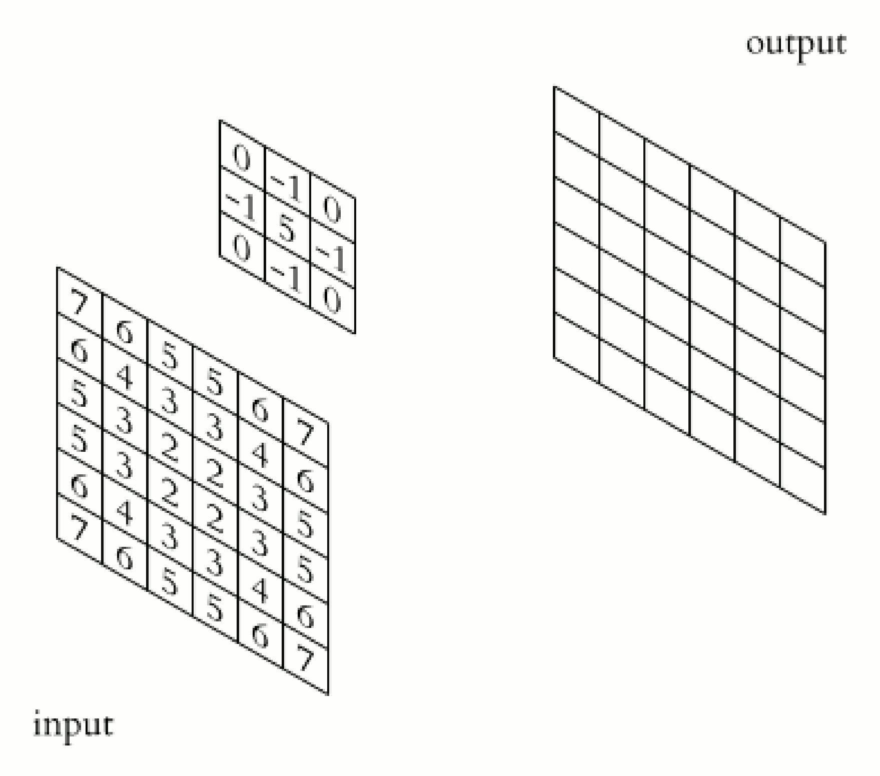
Hyperparameters on Convolution layer

- filters: number of filters (output feature maps)
- kernel size: size of kernel (convolution window: height x width)
- strides: distance between two successive kernel positions
- padding: “valid” = no padding

  “same” = padding with zeros to make the output with the same size as the input

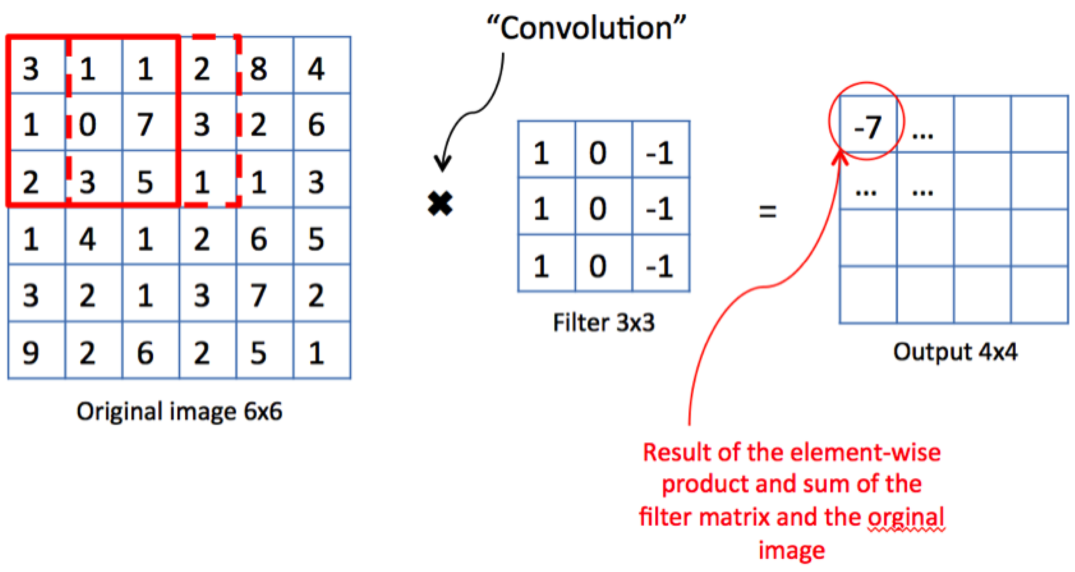


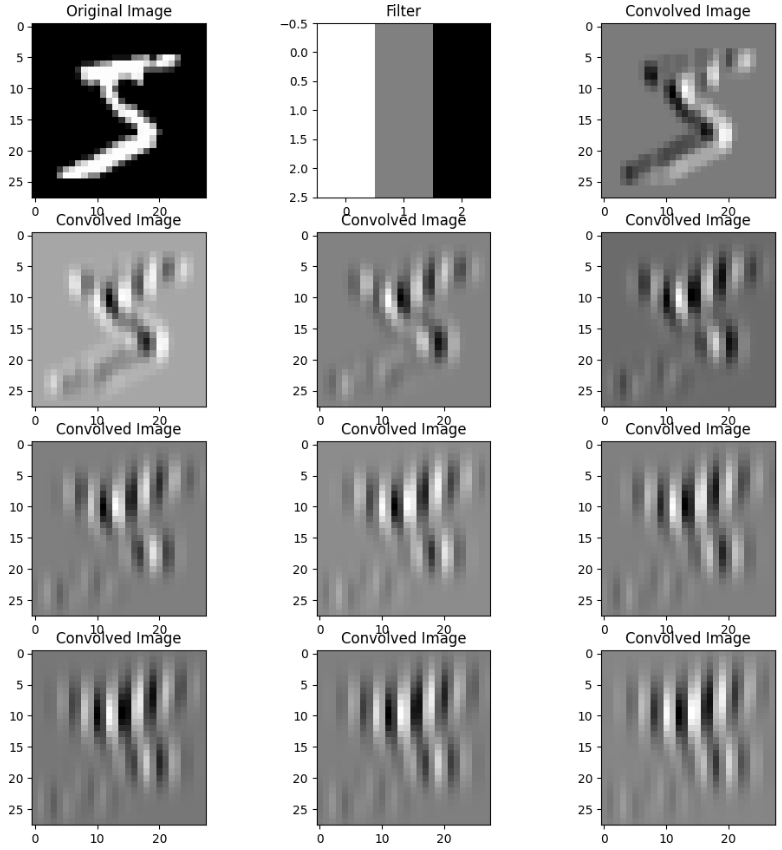
1-2. Convolution Operation



- $g(x,y) = w * f(x,y) = \sum_{i=-a}^{a}\sum_{j=-b}^{b} w(i,j)f(x-i,y-j)$

1-2. Convolution Operation – Kernel

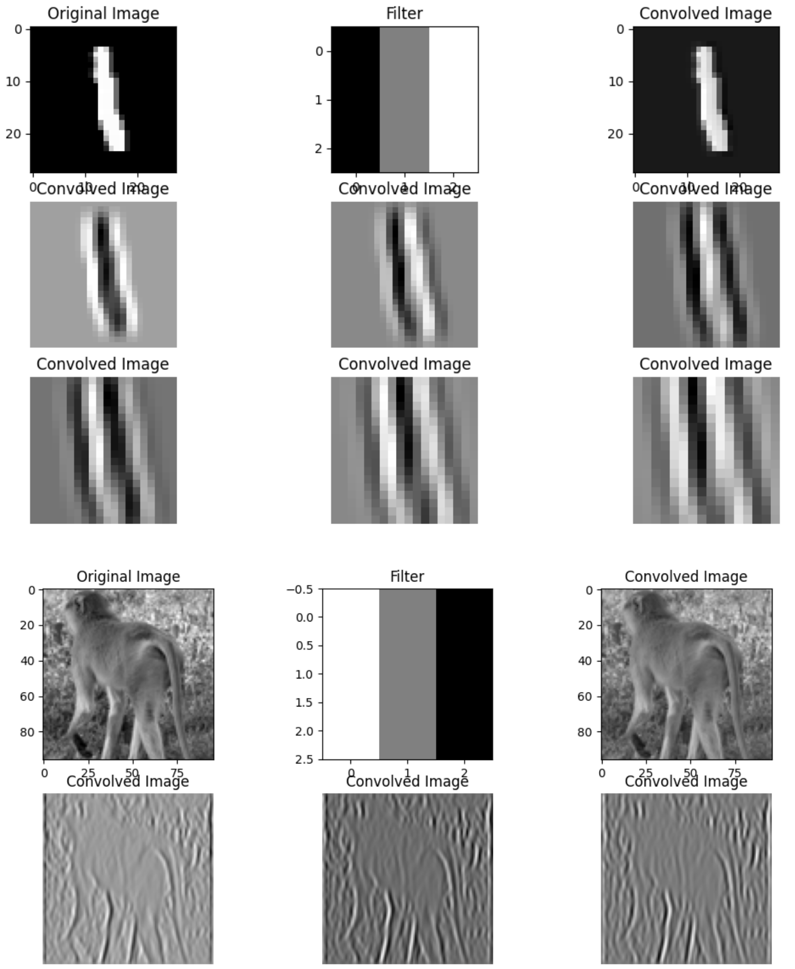




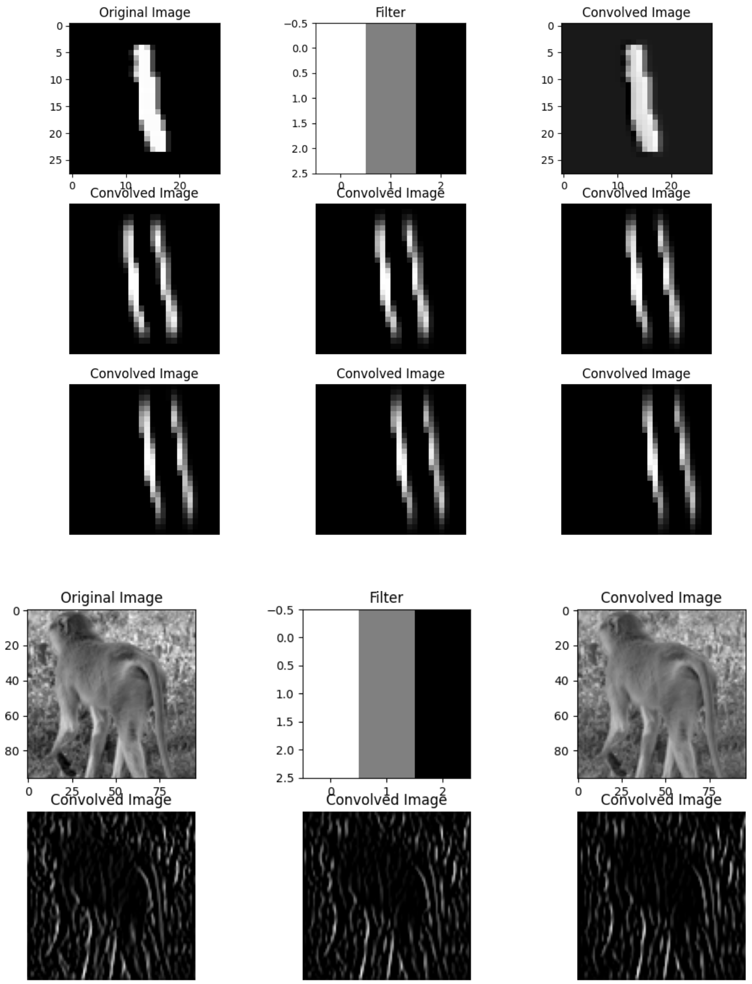
- Note that the size of the output may be different from the size of the input.
- The output of the filter will be large if the input has the feature that the filter can detect.

1-2. Convolution Operation – Some Examples of Kernel

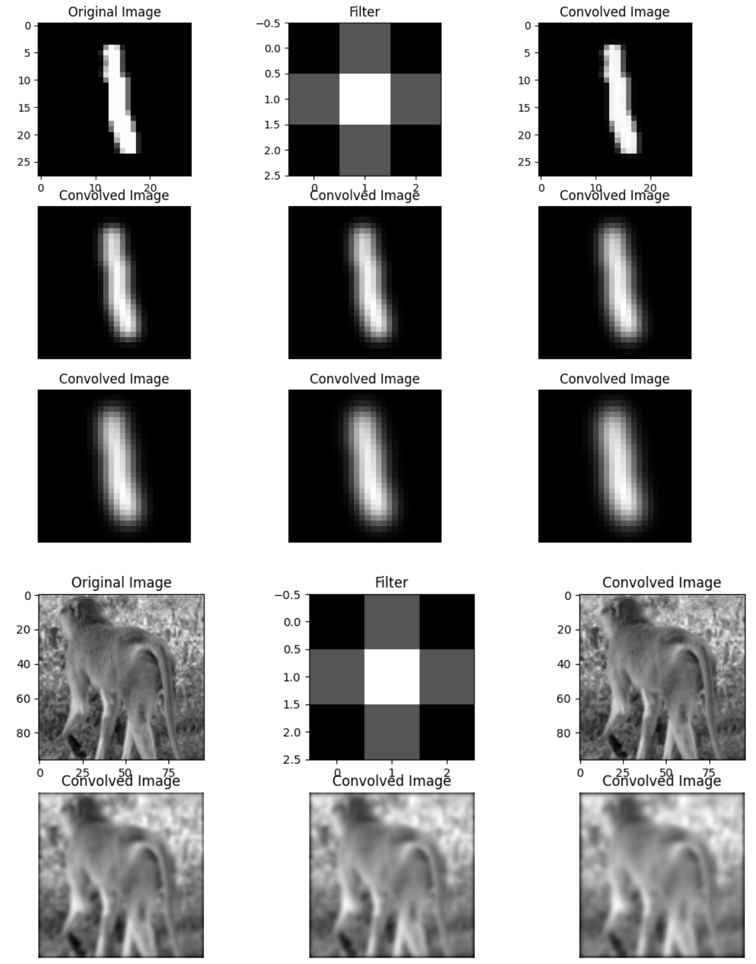
vertical filter (with padding)



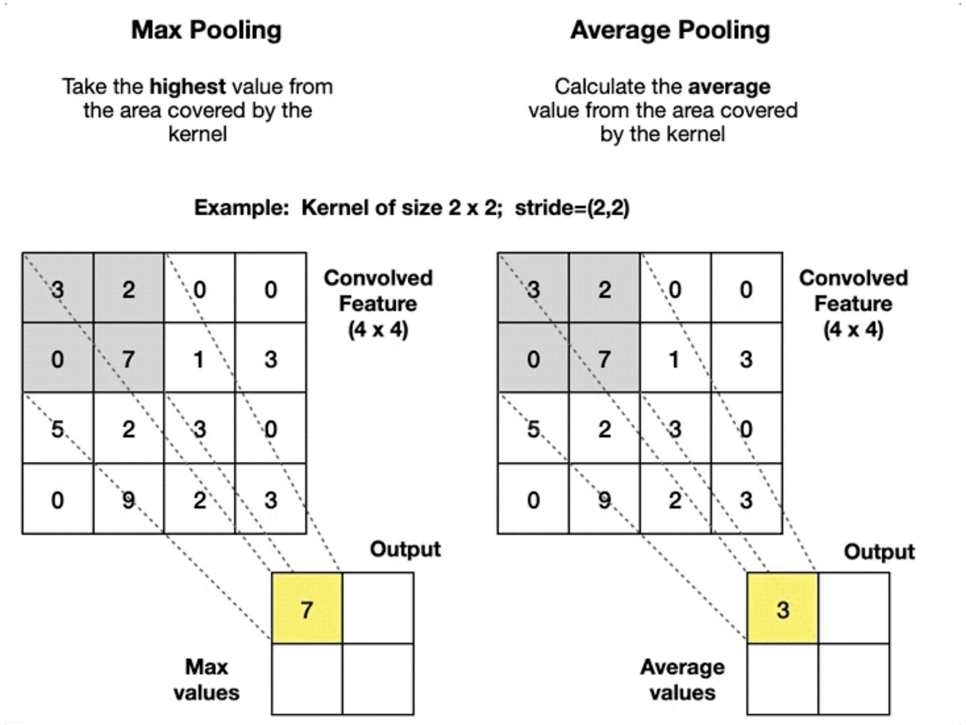
vertical filter (with padding & ReLU)



Gaussian blur filter (with padding)



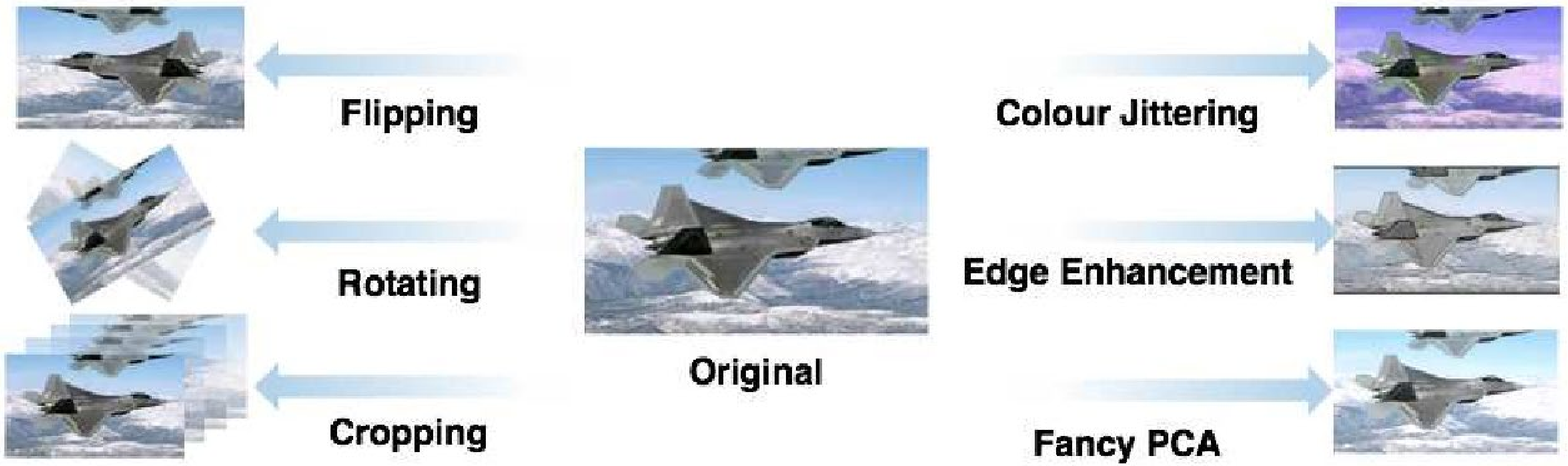
* The code is uploaded on GitHub. (convolution_operation_visualization.py)

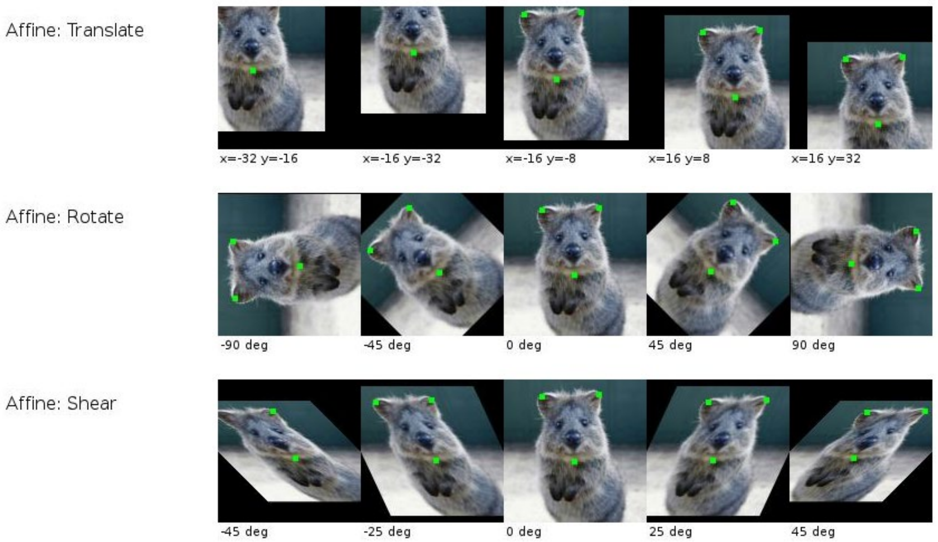


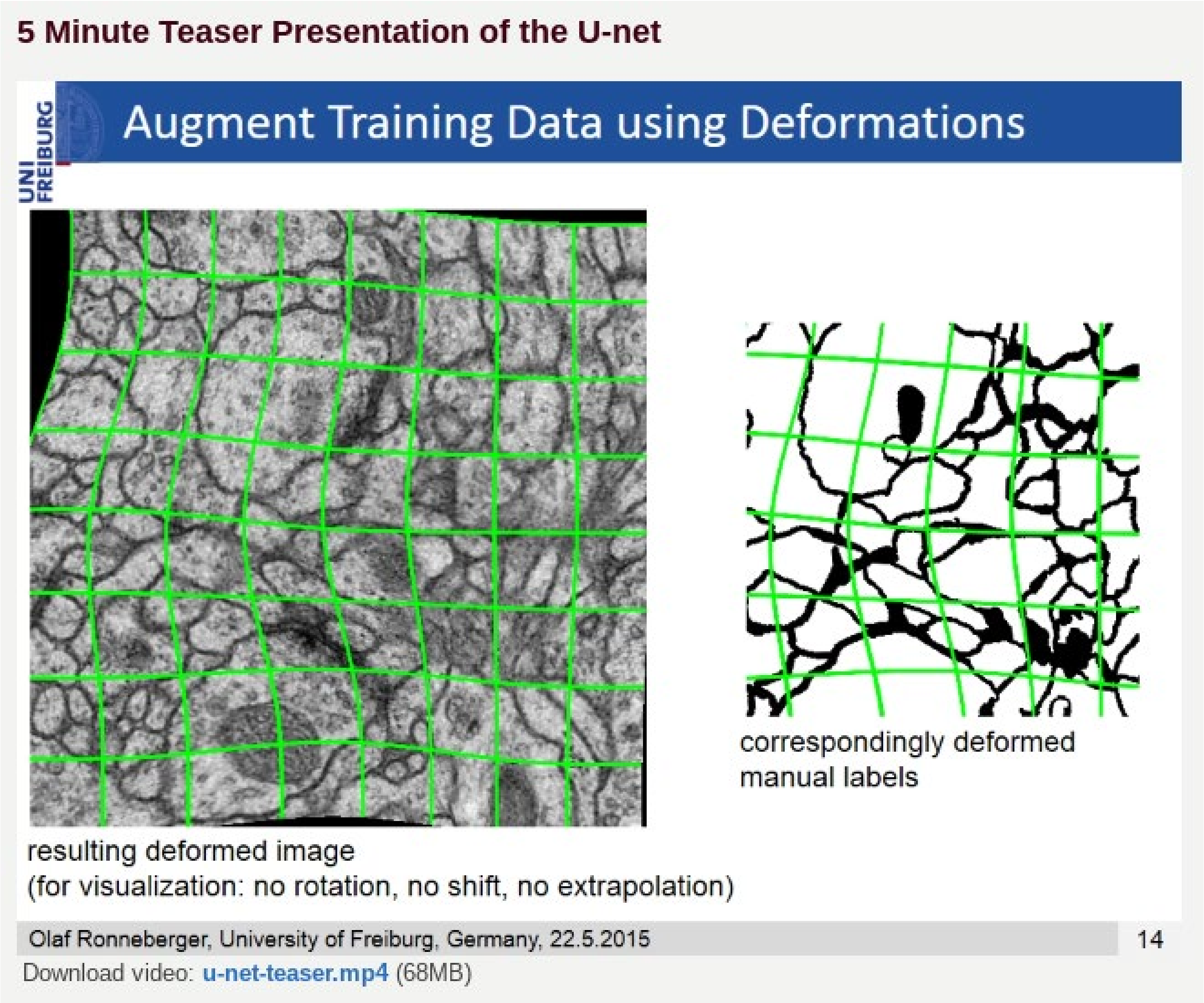
- Pooling layers reduce the dimensions of data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer.
- Max pooling: Take the highest value from the area covered by the kernel
- Average pooling layer: Calculate the average value from the area covered by the kernel

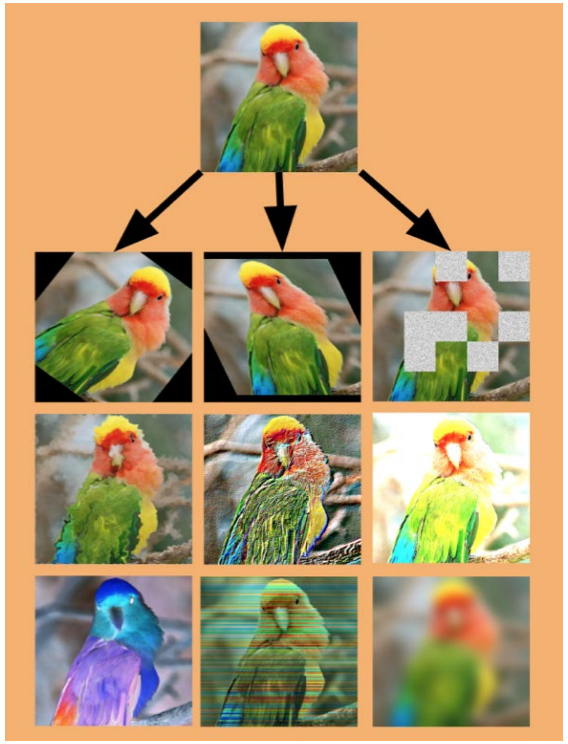
Data Augmentation

- One way of reducing overfitting.


Flipping
Rotating
Cropping
Original
Colour Jittering
Edge Enhancement
Fancy PCA


Affine: Translate
x=-32 y=-16
x=-16 y=-32
x=-16 y=-8
x=16 y=8
x=16 y=32
Affine: Rotate
-90 deg
-45 deg
0 deg
45 deg
90 deg
Affine: Shear
-45 deg
-25 deg
0 deg
25 deg
45 deg


5 Minute Teaser Presentation of the U-net
Augment Training Data using Deformations
UNI FREIBURG
correspondingly deformed manual labels
resulting deformed image
(for visualization: no rotation, no shift, no extrapolation)
Olaf Ronneberger, University of Freiburg, Germany, 22.5.2015
14
Download video: u-net-teaser.mp4 (68MB)



IMAGENET

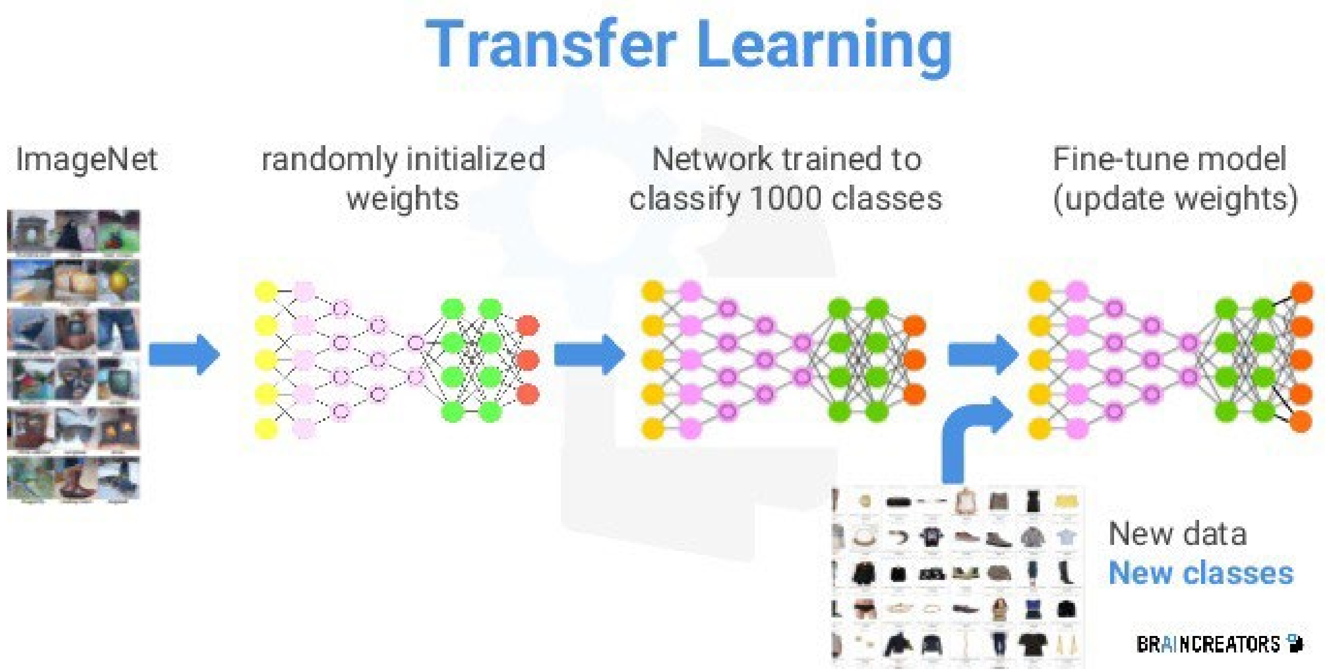
IMAGENET

14,197,122 images, 21841 synsets indexed

Explore Download Challenges Publications Updates About

Not logged in. Login I Signup

ImageNet is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images. Currently we have an average of over five hundred images per node. We hope ImageNet will become a useful resource for researchers, educators, students and all of you who share our passion for pictures.
Click here to learn more about ImageNet, Click here to join the ImageNet mailing list.



- The core idea of transfer learning is to get better initial weights to extract good visual features

3. Transfer Learning using PyTorch

```python
import torchvision.models as models
model = models.resnet50(weights=models.ResNet50_Weights.IMAGENET1K_V1).to(device)

for name, param in model.named_parameters():
    if param.requires_grad:
        print(f"Trainable parameter: {name}")
    else:
        print(f"Non-trainable parameter: {name}")
```

```python
import torchvision
for name in dir(torchvision.models):
    print(name)
```

Check pretrained models in torchvision without resnet50

- Load pretrained model at torchvision.models
- Check model's layer name & trainable / non-trainable parameters

Question

- GitHub에 공유된 코드는 직접 제작한 모델입니다. 바로 위 페이지의 코드를 참고하여 transfer learning을 직접 구현해보고, 결과를 notion에 공유해주세요.