

Computer Vision III

Fundamentals of Artificial Intelligence

Instructor: Chenhui Chu

Email: chu@i.kyoto-u.ac.jp

Teaching Assistant: Youyuan Lin

E-mail: youyuan@nlp.ist.i.kyoto-u.ac.jp

Schedule

- 1. Overview of AI and this Course (4/14)
- 2. Introduction to Python (4/21)
- 3, 4. Mathematics Concepts I, II (4/28, 5/12)
- 5, 6. Regression I, II (5/19, 5/26)
- 7. Classification (6/2)
- 8. Introduction to Neural Networks (6/9)
- 9. Neural Networks Architecture and Backpropagation (6/16)
- 10. Fully Connected Layers (6/23)
- 11, 12, **13. Computer Vision** I, II, **III** (6/30, 7/7, **7/14**)
- 14. Natural Language Processing (7/17)

Overview of This Course

11, 12, 13. Computer vision I,
II, III

14. Natural language
processing

Deep Learning Applications



8. Neural network
introduction

9. Architecture and
backpropagation

10. Feedforward
neural networks

Deep Learning



5. Regression I

6. Regression II

7. Classification

Basic Supervised Machine Learning



2. Python

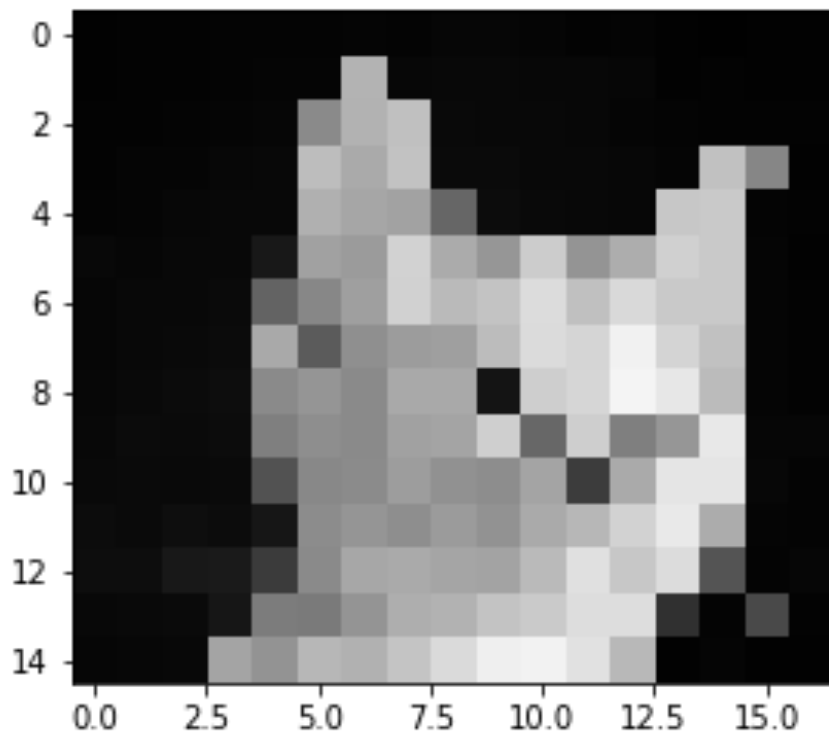
3, 4. Mathematics concepts I, II

Fundamental of Machine Learning

An Image as a 2D Array

- Greyscale image: each pixel has a grey value between 0 (black) and 1 (white)

Image with 18x20 pixels (greyscale)



Array with 18x20 numbers

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.5	0.7	0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.7	0.7	0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.8	0.5	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.7	0.7	0.6	0.4	0.0	0.0	0.0	0.0	0.8	0.8	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.1	0.6	0.6	0.8	0.7	0.6	0.8	0.6	0.7	0.8	0.8	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.4	0.5	0.6	0.8	0.7	0.8	0.9	0.8	0.9	0.8	0.8	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.7	0.4	0.6	0.6	0.6	0.7	0.9	0.8	0.9	0.8	0.8	0.0	0.0	0.0
0.0	0.0	0.0	0.1	0.5	0.6	0.5	0.7	0.7	0.1	0.8	0.8	1.0	0.9	0.7	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.5	0.6	0.5	0.6	0.6	0.8	0.4	0.8	0.5	0.6	0.9	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.3	0.5	0.5	0.6	0.6	0.6	0.6	0.2	0.7	0.9	0.9	0.0	0.0	0.0
0.0	0.0	0.1	0.0	0.1	0.5	0.6	0.6	0.6	0.6	0.7	0.7	0.8	0.9	0.7	0.0	0.0	0.0
0.1	0.1	0.1	0.1	0.2	0.5	0.7	0.7	0.6	0.6	0.7	0.9	0.8	0.9	0.3	0.0	0.0	0.0
0.0	0.0	0.0	0.1	0.5	0.5	0.6	0.7	0.7	0.8	0.8	0.9	0.9	0.2	0.0	0.3	0.0	0.0
0.0	0.0	0.0	0.6	0.6	0.7	0.7	0.8	0.9	0.9	0.9	0.9	0.7	0.0	0.0	0.0	0.0	0.0

“Flat” Convolution

- How we compute:

Input Array

0	0	0	0	0	0
0	0	1	1	0	0
0	1	1	1	0	0
0	1	1	1	1	0
0	1	1	1	1	0
0	1	1	1	1	1

Kernel Array

2	0	2
0	1	0
-1	1	0

Output Array

1	1	1	-1
4	3	3	2
4	5	3	3
4	5	5	3

$$0 \times 2 + 1 \times 0 + 1 \times 2 + 1 \times 0 + 1 \times 1 + 1 \times 0 + 1 \times -1 + 1 \times 1 + 1 \times 0 = 3$$

Input Array (6x6)

0	0	0	0	0	0
0	0	1	1	0	0
0	1	1	1	0	0
0	1	1	1	1	0
0	1	1	1	1	0
0	1	1	1	1	1

Padded Input Array (8x8)

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	1	1	1	0	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	1	0
0	0	0	0	0	0	0	0

Padding

- If one wants the **output size** to be the same as the **input size**, the input is padded with zeros
- In practice, we will always suppose we use padding in the following

Kernel Array (3x3)

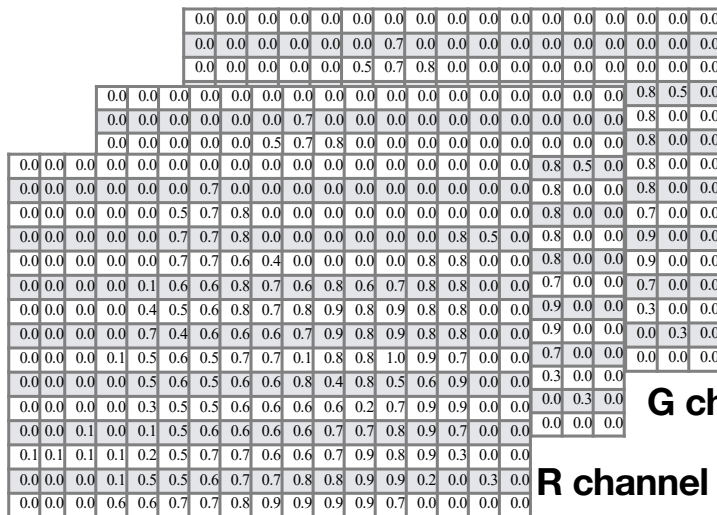
$$\begin{array}{|c|c|c|} \hline 2 & 0 & 2 \\ \hline 0 & 1 & 0 \\ \hline -1 & 1 & 0 \\ \hline \end{array} =$$

Output Array (6x6)

0	0	1	0	-1	0
0	1	1	1	-1	0
0	4	3	3	2	-1
2	4	5	3	3	-1
2	4	5	5	3	2
2	3	5	5	3	3

Color Images

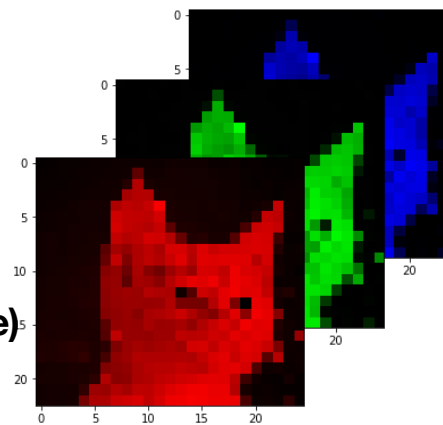
An 18x20x3 array



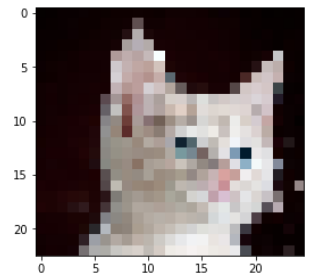
B channel (blue)

G channel (green)

R channel (red)



Superpose all channels



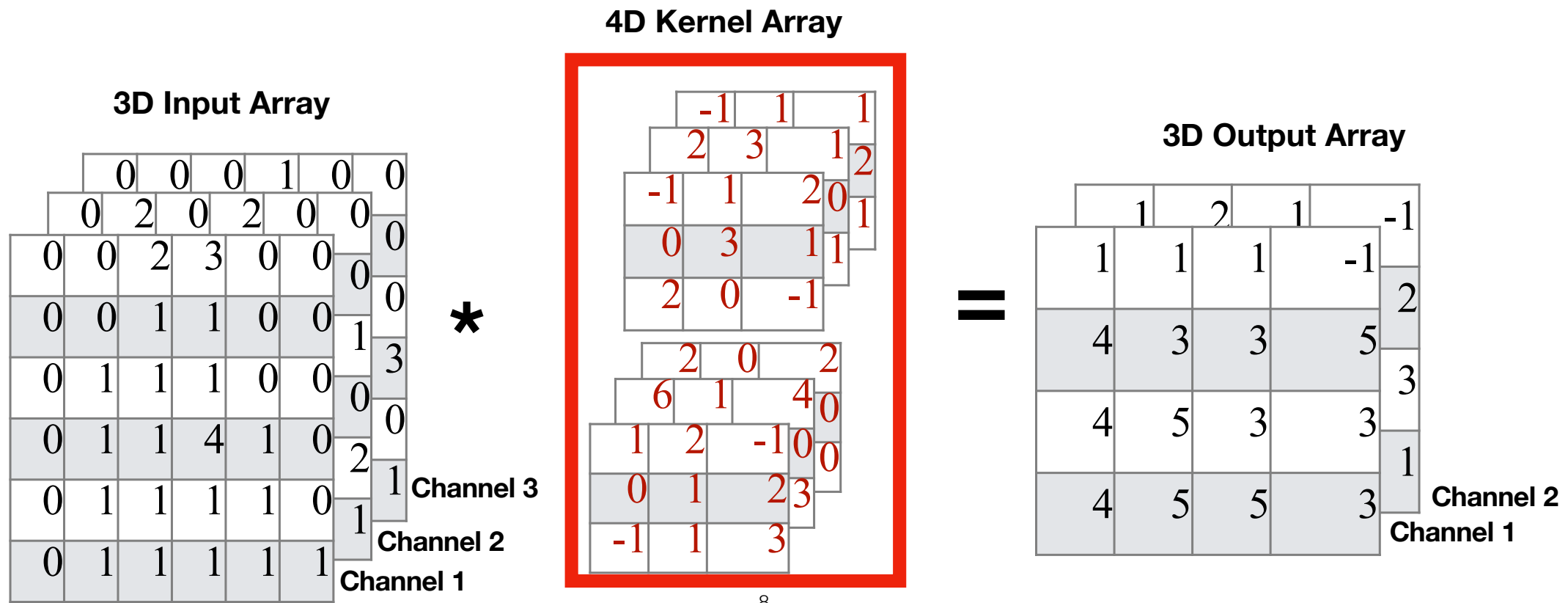
In a computer, a color image is usually represented in the RGB format

We separate the color of the images in Red, Green and Blue Components

We have then 3 images that can each be represented by a 2D array

Volume Convolution

- Also, we can convolve the input with more than one kernel at a time to produce a 3D output with more than one channel



Max Pooling

- Another operation commonly used in CNN: Max Pooling
- Simply takes the max of an area of the input
- Used to reduce the size of the input

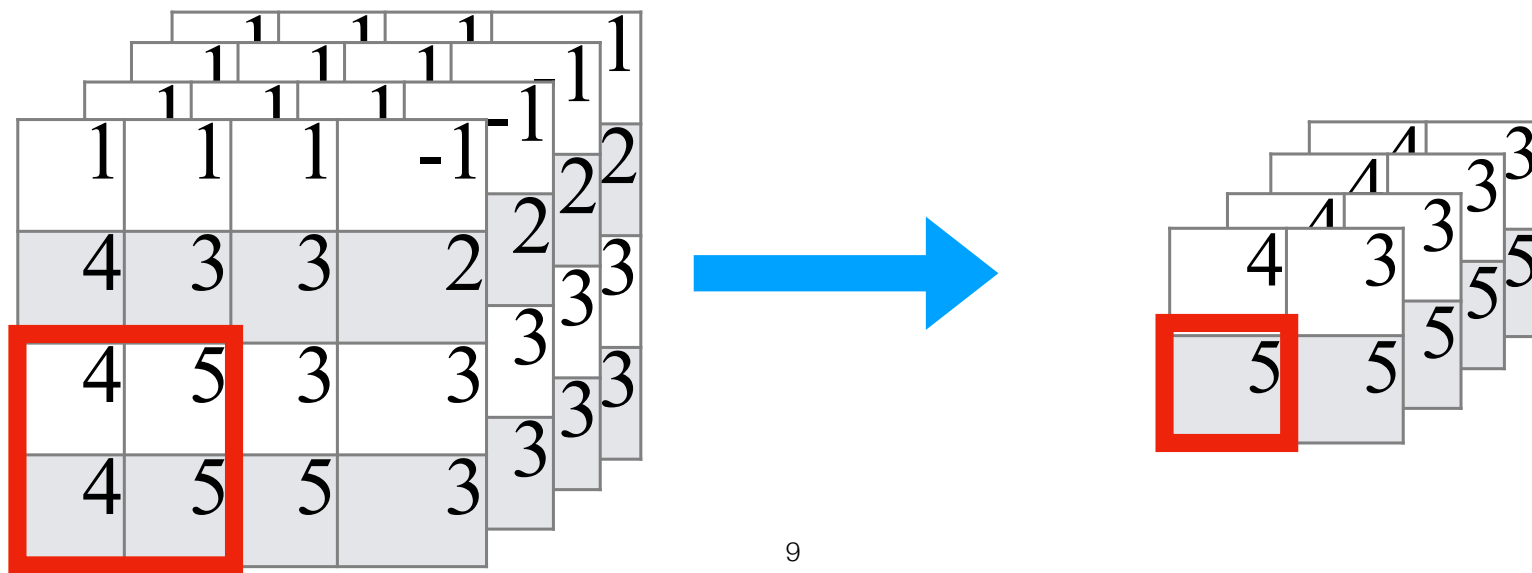
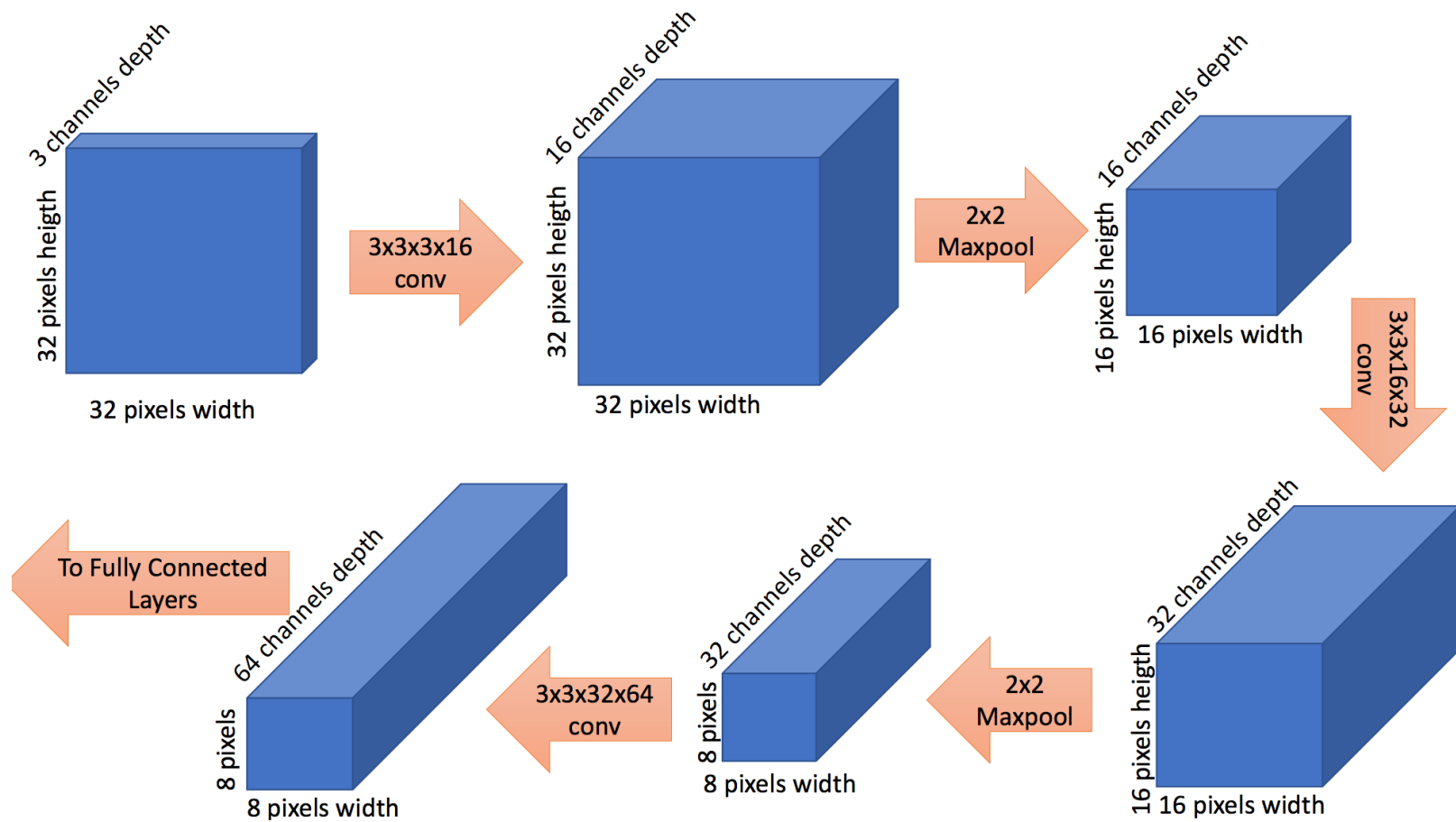
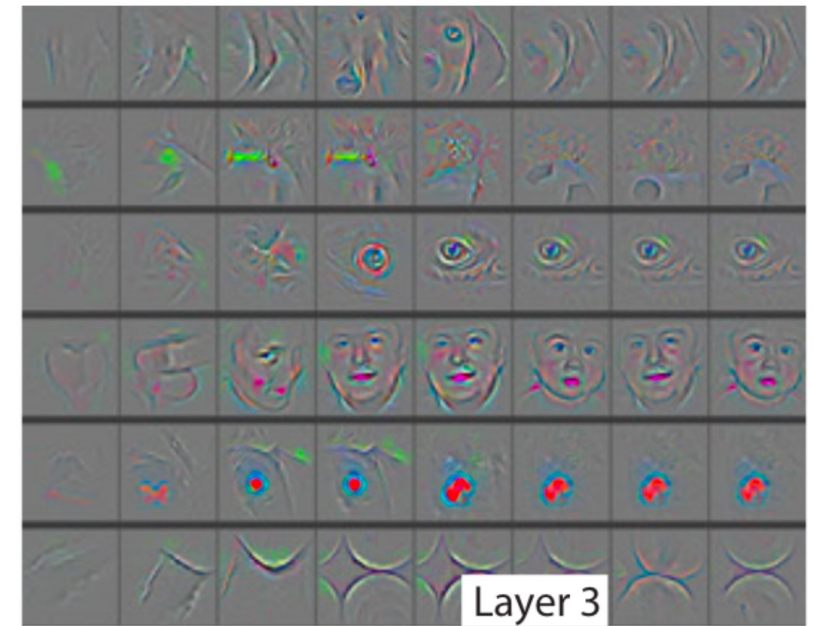
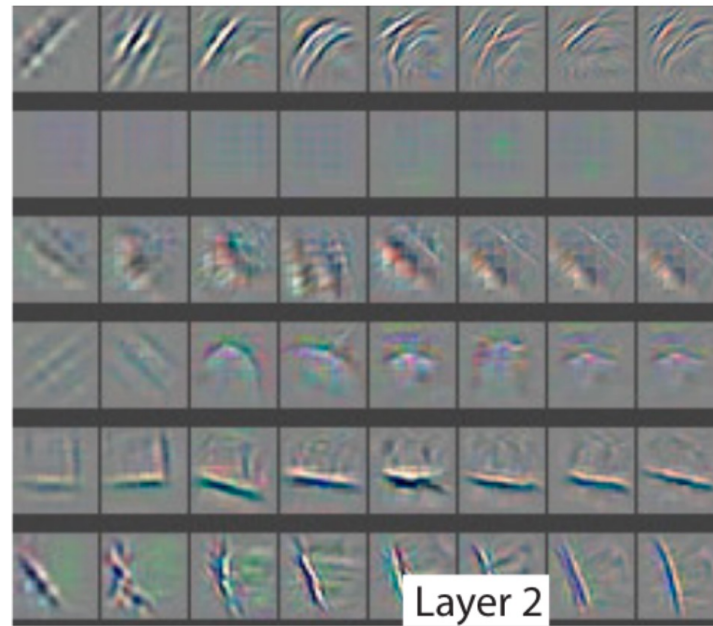
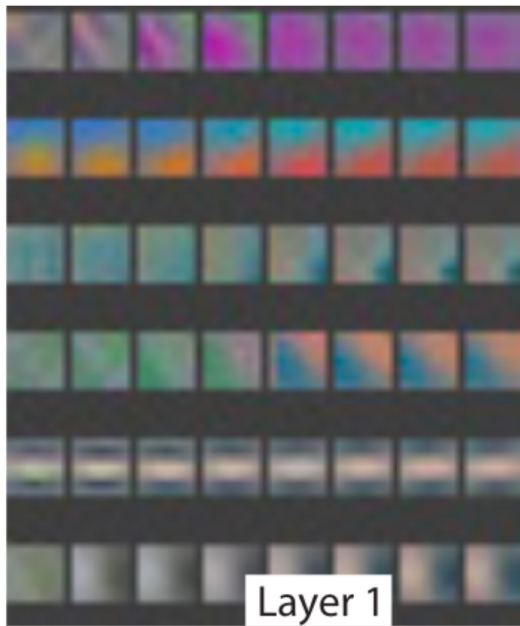


Image Classifier

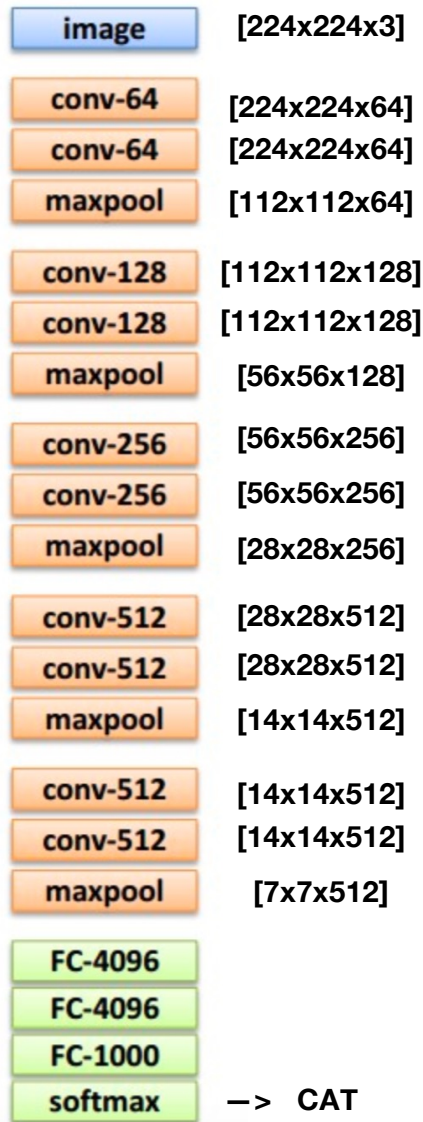
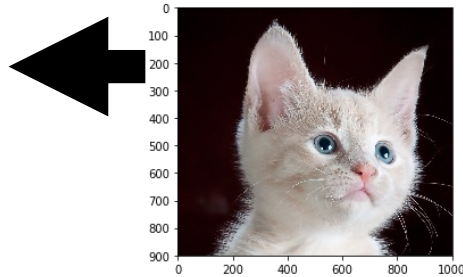


What Do the Convolution Kernels Learn to Recognize?



From Zeiler&Fergus “Visualizing and Understanding Convolutional Networks”

Example: VGG Network



- “Very Deep Convolutional Networks for Large-Scale Image Recognition” Simonyan and Zisserman, 2015
- Best Image Classifier in 2015
- Today the best models have much more layers (> 100)
- Could you compute the number of parameters in each layer?
- All kernels are of size 3 (k=3)
- Conv-64 mean convolutional layer with 64 output channels
- FC-4096 means Fully Connected Layer with 4096 neurons
- Size of input can be guessed from the size of output of the previous layer

Google Colab Notebook

- Let us train an image recognizer with Google Colab notebook

<https://shorturl.at/NfaCU>

Class Questionnaire for the First Semester in AY2025

- Students answer from the “Questionnaire System” in “Common Portal for All Students (<https://student.iimc.kyoto-u.ac.jp/index.html>)”

Report

- Submit the report **in the notebook in pdf** via PandA
 - Submission due: **next lecture**
 - Name the pdf file as **student id_name**

Next Time

- That will be all for image recognition
- Next topic we will discuss is Text Processing

Final Report

- Content: Describe a real problem in your study that can be solved by the models (either basic supervised machine learning or deep learning) introduced in this course together with the model structure and the reason for using the model. It would be better if you implemented the model and conducted experiments. The report should have no less than 2,000 words in English. Figures & diagrams are allowed to assist your illustration.
- Submit your final report in pdf named as [student id_name] via Panda by August 1st, 2025.