



# An Introduction to Convolutional Neural Networks(CNN)

Simon Schupp | 24.07.23

## **Table-Of-Content**



- 1. Introduction
- 2. Convolutional Neural Networks(CNN)
- 3. Self-trained CNN
- 4. Summary

## Introduction

Introduction •00000

Convolutional Neural Networks(CNN)

Self-trained CNN 0000

# **Visual Data and Algorithms**





Tesla Autopilot object Detection



Source: Google Blog "PaLM-E

Google PaLm-E robot

Introduction 000000

Convolutional Neural Networks(CNN)

Self-trained CNN

## **How Computer see**



## What I see



# What a computer sees

08 02 22 97 38 15 00 40 00 75 04 05 07 78 52 12 50 77 91 08 49 99 40 17 81 18 57 60 87 17 40 98 43 69 48 04 56 62 00 81 49 31 73 57 91 42 99 37 14 06 73 38 33 00 34 91 13 36 65 52 70 95 22 04 60 11 42 69 24 68 56 01 32 56 71 37 02 36 91 22 31 16 71 51 67 63 89 41 92 36 56 22 40 40 28 66 33 13 50 24 47 32 60 99 03 45 02 44 75 33 53 78 36 84 20 35 17 12 50 32 98 81 28 64 23 67 10 26 38 40 67 59 54 70 66 18 38 64 70 67 26 20 68 02 62 12 20 98 63 94 39 63 08 40 91 66 49 94 21 45 55 80 66 73 99 26 97 17 78 78 96 83 14 88 38 49 94 37 24 55 58 08 66 73 99 26 97 17 78 78 96 83 14 88 38 49 94 37 22 13 66 23 09 75 00 76 44 20 45 35 14 00 61 33 97 34 31 33 95 78 17 53 28 27 75 31 67 15 94 03 80 46 21 61 40 95 33 65 25 16 39 05 42 96 35 71 19 07 05 44 44 37 44 60 21 58 51 54 17 55 19 00 61 68 05 94 47 69 28 73 92 13 86 52 17 77 04 89 55 40 42 56 60 07 57 62 20 72 03 46 33 67 45 62 21 18 62 26 79 33 27 98 66 88 56 60 87 57 62 20 72 03 46 33 67 45 52 18 22 26 39 35 36 98 60 42 26 77 38 25 78 17 57 89 12 86 52 17 77 04 89 55 40 42 26 73 38 25 98 11 24 94 72 18 08 62 17 77 04 89 55 40 42 26 73 38 25 98 11 24 94 72 18 08 46 27 93 27 98 66 88 56 60 47 17 73 38 25 98 11 24 94 72 18 08 46 29 32 79 32 79 36 66 88 56 60 41 72 30 23 88 34 62 99 69 82 67 59 85 74 04 36 16 20 73 35 29 78 31 90 01 78 31 49 71 48 86 81 16 23 57 05 41 17 05 38 52 77 33 15 29 78 31 90 01 78 31 49 71 48 86 81 16 23 57 05 44 10 17 05 47 18 85 18 47 18 31 69 14 69 12 62 37 05 48 10 17 05 47 18 35 15 46 91 62 33 34 66 14 35 20 18 91 97 78

Source: GitHub, hosamelsafty/Cats-VS-Dogs

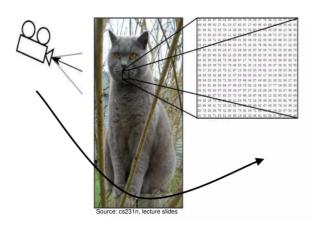
Introduction

Convolutional Neural Networks(CNN)

Self-trained CNN

# **Challenges: Viewpoint**





When we change the viewpoint, all the pixel change!

Introduction

Convolutional Neural Networks(CNN)

Self-trained CNN

## More challenges



- Viewpoint
- Illumination
- Occlusions
- Background Clutter
- etc.

# **Image Classifier**



function CLASSIFYIMAGE(Image : Image)
 // What to do here? Magic?
 return label
end function

It's not obvious how to hard code this.

# Image Classifier



**function** CLASSIFYIMAGE(*Image* : *Image*) // What to do here? Magic? return label end function

It's not obvious how to hard code this.

**Idea:** Learn how to classify the Images — Use Convolutional Neural Networks(CNN)

# **Convolutional Neural Networks(CNN)**

Introduction

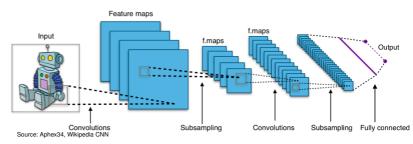
Convolutional Neural Networks(CNN) •00000000

Self-trained CNN

### **CNN Overview**



- Used mostly to classify images
- A CNN is a specialized NN
- It can pick up patterns and make sense
- Consists of the following Components:
  - Convolutional Layer
  - Pooling Layer
- Convolutional Layer consists out of multiple Filters



A Typical CNN

Introduction

Convolutional Neural Networks(CNN) 0.0000000

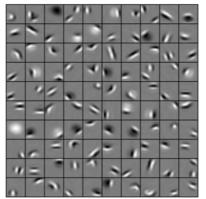
Self-trained CNN

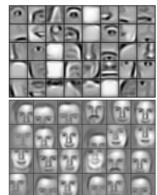
## **Patterns**



- Simple patterns: edges, shapes, textures
- More complex patterns: ears, eyes, noses
- Even more complex patterns: faces, humans

These patterns are detected by filters of the Convolutional Layers.





Source: Brandon Rohrer, course 193

Introduction 000000 Convolutional Neural Networks(CNN)

Self-trained CNN

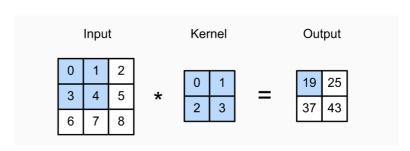
### Filters/Kernel



#### Filter/Kernel: A matrix

- initalized with random values
- values adjusted over time, to detect pattern

Kernel of Convolutional Layer slides over Input



Source: d2l.ai, Chapter 7.2

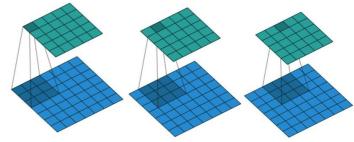
Convolution Operation

## Convolution



#### Two Important Hyperparameters:

- Stride
- Filter-Size



Source: Jelo Salomon, Lung Cancer Detection using Deep Learning

A Convolution with Stride 1

## **Convolution Animation**



#### Convolution Animation

Source: https://youtu.be/xjqCTp4xAtA

Introduction

Convolutional Neural Networks(CNN)

Self-trained CNN

# **Pooling Layer**



- A way to reduce the dimensionality of our Network
  - Requires less computation
- Many different ways to do this
  - Max-Pooling
  - Average-Pooling
  - 1x1 Convolutions

# **Pooling Layer**



12	20	30	0			
8	12	2	0	$2 \times 2$ Max-Pool	20	30
34	70	37	4		112	37
112	100	25	12			

Source: cs231n lecture slides

A Max Pooling Layer

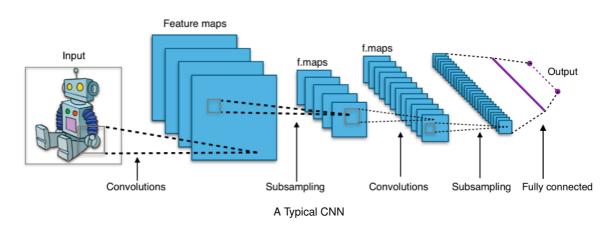
Introduction

Convolutional Neural Networks(CNN) 000000000

Self-trained CNN 0000

# **CNN: Bringing it Together**





Introduction

Convolutional Neural Networks(CNN)

Self-trained CNN

## **Self-trained CNN**

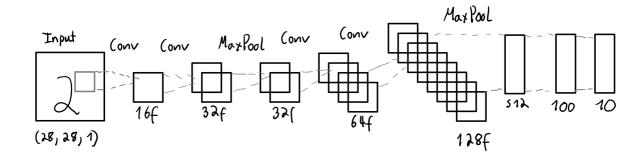
Introduction 000000

Convolutional Neural Networks(CNN) 000000000

Self-trained CNN •000

## **Architecture**





Introduction

Convolutional Neural Networks(CNN)

Self-trained CNN ○●○○

## **Trainings-Details**



Dataset: MNIST(60,000 Images)

Trainings time: 5 Epochs

Batch size: 1024Optimizer: Adam

Learning-rate: 0.001

Trained on Google Colab with GPU



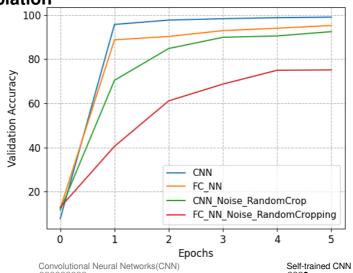
Example Data from MNIST dataset

Introduction 000000 Convolutional Neural Networks(CNN)

Self-trained CNN ○○●○

## **Results & Ablation**





Introduction

0000

Introduction 000000

Convolutional Neural Networks(CNN) 000000000

Self-trained CNN 0000



Teaching a Computer the semantic meaning of Images is hard

Introduction

Convolutional Neural Networks(CNN)

Self-trained CNN

NN Summary

Proseminar: Grundlagen des maschinellen



- Teaching a Computer the semantic meaning of Images is hard
- But it's possible with the help of CNN

24.07.23



- Teaching a Computer the semantic meaning of Images is hard
- But it's possible with the help of CNN
- Nowdays its rather easy to get good performance on simple datasets

### Literature



- [1] Alex Krizhevsky, Ilya Sutskever und Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: Advances in Neural Information Processing Systems. Hrsg. von F. Pereira u. a. Bd. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper\_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.
- [2] Y. LeCun u. a. "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation* 1.4 (1989), S. 541–551. DOI: 10.1162/neco.1989.1.4.541.
- [3] Y. Lecun u. a. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), S. 2278–2324. DOI: 10.1109/5.726791.

Literatur