

## Computer Vision Frameworks

Es gibt diverse OpenSource Frameworks:



python

lib C++ Library

... runs best on ...



scikit-image

SimpleCV

Mahotas

Other Frameworks:  
VisionWorks / nvidia

## Projekt CV Drone

<http://cvdrone.de> gestartet Anfang 2017

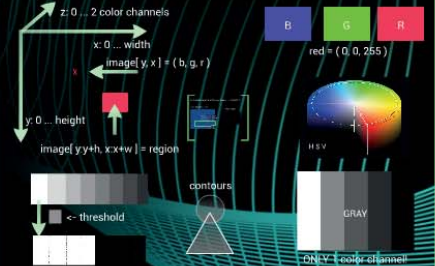


Projekt gestartet in der 20% bzw. "+1" Zeit

- Computer Vision Fähigkeiten
- Audio Analyse
- autonomes Fliegen

... show me a DEMO

## Computer Vision Basics



## Algorithmen

### Background Subtraction

... nicht genau den selben Hintergrund überabstrahieren  
 Falschung seit Jahrzehnten  
 Immer noch keine One Fits All Algorithmen



### Convolution



Feature Extraction / Classifiers

## klassische

Einfach: Frame Differenz  
 Common: GMM / MoG2  
 Advanced: SuBSENSE

Probleme:  
 - nur statische Kamera  
 - Lichtänderungen  
 - Schatten  
 - Rechenlastung

Matrix-Multiplikation + Summe  
 Verwischen, Schärfen, Detektieren ... damit kann man extrem viel machen  
 Haar Cascade Classifiers  
 HOG  
 Local Binary Patterns

## einfache Detektoren

Sliding Window



Multi Scale



Gesichts-Erkennung

Haar Cascade Classifier

- heißt oft Viola Jones (2004)  
 - ziemlich schnell läuft auf Raspberry Pi  
 - viele Fehler in der Erkennung

(Default) HOG Detector

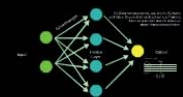
- weniger falsche Detektionen  
 - braucht viel mehr Rechenleistung  
 - optimieren für Regions of Interest (ROI)

## Neuronale Netze

Alle gezeigten klassischen Methoden haben ein Problem:  
 - der Computer versteht nicht wirklich was im Bild passiert  
 - alles ist ein "handcrafted overfitting" - passend für eine Szene

Mit ImageNet kam die Wende: Heute erreichen Algorithmen "Super-Human-Performance"

Neuronale Netze ermöglichen stetiges Lernen aus Daten.



## Computer Vision ... finally takes off!

- Disziplin der Informatik gibt es seit Jahrzehnten
- Algorithmen und Methodologien sind anfechtbar  
 - aber erst durch Cloud, Mehr DATEN und GPUs gibt es neue Möglichkeiten
- der Vortrag zeigt den Pfad von einfachen Werkzeugen und deren Vor- und Nachteilen hin zu Neuronalen Netzen

Follow us on <https://youtube.com/c/codecentricAI>

Team codecentric - Karlsruhe

Macht uns auf

DAS FEST



Fragen ???

Oliver Moser - 2018

@moseroli

oliver.moser@codecentric.de

## Deep Learning

Damit werden beeindruckende neue Applikationen möglich ...  
 Aber: Man braucht mehr POWER (und mehr Daten).



Es gibt viele weitere beeindruckende Beispiele ...

Nach Jahrzehnten der Forschung beginnen Computer zu verstehen, was in Bildern zu sehen ist. Die cambrische Explosion resultierte unter anderem aus der Entwicklung des Sehens. Die heutige Entwicklung läuft aber um ein Vielfaches schneller als in der Evolution ...

How we interact with machines will soon change ...

## Computer Vision Frameworks

Es gibt diverse OpenSource Frameworks:



python

scikit-image

SimpleCV

Mahotas

Lib C++ Library



Other Frameworks:  
VisionWorks / nvidia

... runs best on ...



## Projekt CV Drone

<http://cvdrone.de> gestartet Anfang 2017

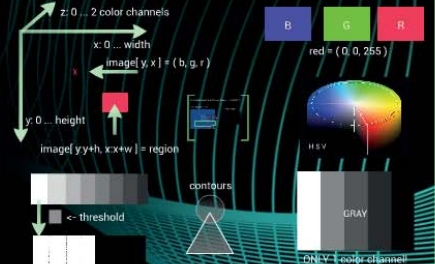


Projekt gestartet in der 20% bzw. "+1" Zeit

- Computer Vision Fähigkeiten
- Audio Analyse
- autonomes Fliegen

... show me a DEMO

## Computer Vision Basics



## Algorithmen

klassische

### Background Subtraction

• meist gegen den aktuellen Video-Überwachungs-Stream  
• Falschmeldung seit Jahrzehnten  
• Immer noch keine One-Fits-All-Algorithmen



Einfach: Frame Differenz  
Common: GMM / MoG2  
Advanced: SuBSENSE

Probleme:  
- nur statische Kamera  
- Lichtänderungen  
- Schatten  
- Rechenlastung

### Convolution



Feature Extraction / Classifiers

Matrix-Multiplikation + Summe  
Verwischen, Schärfen, Detektieren ... damit kann man extrem viel machen  
Haar Cascade Classifiers  
HOG  
Local Binary Patterns

## Detektoren

einfache

Sliding Window



Multi Scale



Gesichts-Erkennung

Haar Cascade Classifier

- beliebt um 2004 bis ~2006  
- ziemlich schnell läuft auf Raspberry Pi  
- viele Fehler in der Erkennung

(Default) HOG Detector

- weniger falsche Detektionen  
- braucht viel mehr Rechenleistung  
- optimieren für Regions of Interest (ROI)

## Neuronale Netze

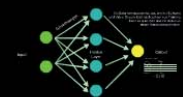
Alle gezeigten klassischen Methoden haben ein Problem:

- der Computer versteht nicht wirklich was im Bild passiert
- alles ist ein "handcrafted overfitting" - passend für eine Szene

Mit ImageNet kam die Wende:

Heute erreichen Algorithmen "Super-Human-Performance"

Neuronale Netze ermöglichen stetiges Lernen aus Daten.



## Computer Vision

... finally takes off!

- Disziplin der Informatik gibt es seit Jahrzehnten
- Algorithmen und Methodologien sind anfechtbar  
→ Aber erst durch Cloud, Mehr DATEN und GPUs gibt es neue Möglichkeiten
- der Vortrag zeigt den Pfad von einfachen Werkzeugen und deren Vor- und Nachteilen hin zu Neuronalen Netzen

Follow us on <https://youtube.com/c/codecentricAI>

Team codecentric - Karlsruhe

Macht uns auf

DAS FEST



Fragen ???

Oliver Moser - 2018

@moseroli

oliver.moser@codecentric.de

## Deep Learning

Damit werden beeindruckende neue Applikationen möglich ...

Aber: Man braucht mehr POWER (und mehr Daten).



Es gibt viele weitere beeindruckende Beispiele ...

Nach Jahrzehnten der Forschung beginnen Computer zu verstehen, was in Bildern zu sehen ist. Die cambrische Explosion resultierte unter anderem aus der Entwicklung des Sehens. Die heutige Entwicklung läuft aber um ein Vielfaches schneller als in der Evolution ...

How we interact with machines will soon change ...

# Computer Vision ... finally takes off!

- Disziplin der Informatik gibt es seit Jahrzenten
- Algorithmen und Methodologien sind altbekannt
- Aber erst durch Cloud, MEHR DATEN und GPUs gibt es neue Möglichkeiten
- der Vortrag zeigt den Pfad von einfachen Werkzeugen und deren Vor- und Nachteilen hin zu Neuronalen Netzen



Follow us on <https://youtube.com/c/codecentricAI>

Team codecentric - Karlsruhe

Meet us auf:

DAS FEST

## Fragen ???



### FAZIT

Klassische Methoden funktionieren "unter engen Labor-Bedingungen" ... sie lernen nicht dazu.

Deep Learning ist keine Magie

Man muss nicht Mathematik studiert haben, um es anzuwenden

... Detektoren werden immer besser und besser.

Neue Möglichkeiten:

- smarte Sensoren
- Überwachung UND Datenschutz
- neue Formen der Interaktion
- Automatische Auswertungen

Oliver Moser - 2018



@moseroli

[oliver.moser@codecentric.de](mailto:oliver.moser@codecentric.de)

# Computer Vision

## ... finally takes off!

- Disziplin der Informatik gibt es seit Jahrzehnten
  - Algorithmen und Methodologien sind altbekannt
    - Aber erst durch Cloud, MEHR DATEN und GPUs gibt es neue Möglichkeiten
      - der Vortrag zeigt den Pfad von einfachen Werkzeugen und deren Vor- und Nachteilen hin zu Neuronalen Netzen



Follow us on <https://youtube.com/c/codecentricAI>

Team codecentric - Karlsruhe



## Computer Vision Frameworks

Es gibt diverse OpenSource Frameworks:



python

lib C++ Library

... runs best on ...



scikit-image

SimpleCV

Mahotas

Other Frameworks:  
VisionWorks / nvidia

## Projekt CV Drone

<http://cvdrone.de> gestartet Anfang 2017

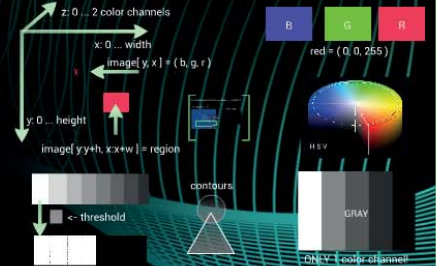


Projekt gestartet in der 20% bzw. "+1" Zeit

- Computer Vision Fähigkeiten
- Audio Analyse
- autonomes Fliegen

... show me a DEMO

## Computer Vision Basics



## Algorithmen

### Background Subtraction

... nicht genau den selben Hintergrund überabstrahieren + trennen  
Falschung seit Jahrzehnten  
Immer noch keine One Fits All Algorithmen



### Convolution



Feature Extraction / Classifiers

klassische

Einfach: Frame Differenz  
Common: GMM / MoG2  
Advanced: SuBSENSE

Probleme:  
- nur statische Kamera  
- Lichtänderungen  
- Schatten  
- Rechenlastung

Matrix-Multiplikation + Summe  
Verwischen, Schärfen, Detektieren ... damit kann man extrem viel machen  
Haar Cascade Classifiers  
HOG  
Local Binary Patterns

## einfache Detektoren

Sliding Window

Multi Scale

Gesichts-Erkennung



Haar Cascade Classifier

- relativ gut (Dmitry Jorjic ~2006)  
- ziemlich schnell läuft auf Raspberry Pi  
- viele Fehler in der Erkennung

(Default) HOG Detector

- weniger falsche Detektionen  
- braucht viel mehr Rechenleistung  
-> optimieren für Regions of Interest (ROI)

## Neuronale Netze

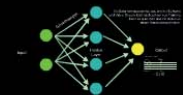
Alle gezeigten klassischen Methoden haben ein Problem:

- der Computer versteht nicht wirklich was im Bild passiert
- alles ist ein "handcrafted overfitting" - passend für eine Szene

Mit ImageNet kam die Wende:

Heute erreichen Algorithmen "Super-Human-Performance"

Neuronale Netze ermöglichen stetiges Lernen aus Daten.



## Computer Vision

... finally takes off!

- Disziplin der Informatik gibt es seit Jahrzehnten
- Algorithmen und Methodologien sind anfechtbar  
-> Aber erst durch Cloud, Mehr DATEN und GPUs gibt es neue Möglichkeiten
- der Vortrag zeigt den Pfad von einfachen Werkzeugen und deren Vor- und Nachteilen hin zu Neuronalen Netzen

Follow us on <https://youtube.com/c/codecentricAI>

Team codecentric - Karlsruhe

Macht uns auf

DAS FEST



Fragen ???

Oliver Moser - 2018

@moseroli

oliver.moser@codecentric.de

## Deep Learning

Damit werden beeindruckende neue Applikationen möglich ...

Aber: Man braucht mehr POWER (und mehr Daten).



Es gibt viele weitere beeindruckende Beispiele ...

Nach Jahrzehnten der Forschung beginnen Computer zu verstehen, was in Bildern zu sehen ist. Die cambrische Explosion resultierte unter anderem aus der Entwicklung des Sehens. Die heutige Entwicklung läuft aber um ein Vielfaches schneller als in der Evolution ...

How we interact with machines will soon change ...

# Computer Vision Frameworks

Es gibt diverse OpenSource Frameworks:



python



Dlib C++ Library

... runs best on ...



ubuntu



scikit-image

SimpleCV

Mahotas

Other Frameworks:  
VisionWorks / nvidia

# Projekt CV Drone

<http://cvdrone.de> gestartet Anfang 2017



Projekt gestartet in der 20% bzw. "+1" Zeit

- Computer Vision Fähigkeiten
- Audio Analyse
- autonomes Fliegen

... show me a DEMO





Folgendes Video demonstriert die Verknüpfung von einfachen Methoden. Es ist in Echtzeit auf einfacher Hardware lauffähig.



# a DEMO

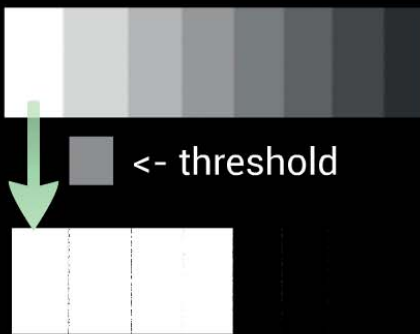
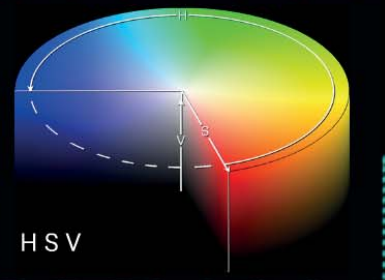
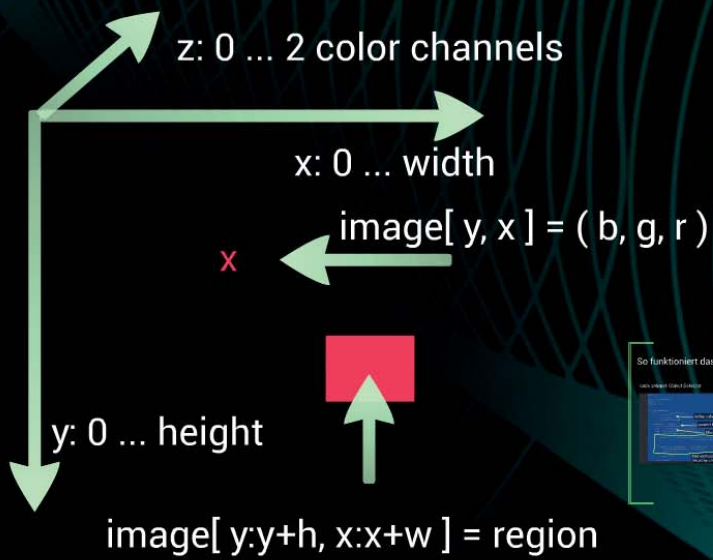
... im zweiten Schritt zeigen wir die Schwächen dieser einfachen Methoden.

evdrone.de - pydrone version v. 0.1



2017-03-24 15:32:29/641 - 2021

# Computer Vision Basics



contours



# So funktioniert das Demo Video ... Schritt 1

code snippet: Object Detector

```
8 import cv2
9
10 cam = cv2.VideoCapture(0)
11 margin_x = 60
12 margin_y = 160
13
14
15 class ObjectDetector:
16     def __init__(self, min_area=10):
17         self.color_lower = (15, 150, 150)
18         self.color_upper = (15, 240, 250)
19         self.min_area = min_area
20
21     def detect(self, frame):
22         hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
23         mask = cv2.inRange(hsv, self.color_lower, self.color_upper)
24         mask = cv2.erode(mask, None, iterations=2)
25         mask = cv2.dilate(mask, None, iterations=2)
26
27         _, contours, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL,
28                                         cv2.CHAIN_APPROX_SIMPLE)
29
30         try:
31             contour = sorted(contours, key=cv2.contourArea, reverse=True)[0]
32             if cv2.contourArea(contour) < self.min_area:
33                 contour = None
34         except:
35             contour = None
36
37         return contour
38
```

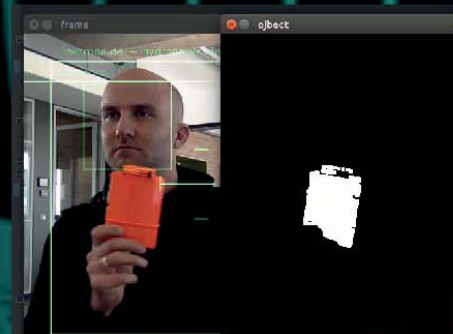
define color boundaries

convert frame to HSV

filter only orange

find contours, filter noise (must be > min\_area)

video output:



## code snippet: Object Detector

```
8 import cv2
9
10 cam = cv2.VideoCapture(0)
11 margin_x = 60
12 margin_y = 160
13
14
15 class ObjectDetector:
16     def __init__(self, min_area=10):
17         self.color_lower = (1, 150, 150)
18         self.color_upper = (40, 240, 255)
19         self.min_area = min_area
20
21     def detect(self, frame):
22         hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
23         mask = cv2.inRange(hsv, self.color_lower, self.color_upper)
24         mask = cv2.erode(mask, None, iterations=2)
25         mask = cv2.dilate(mask, None, iterations=2)
26
27         _, contours, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL,
28                                         cv2.CHAIN_APPROX_SIMPLE)
29
30         try:
31             contour = sorted(contours, key=cv2.contourArea, reverse=True)[0]
32             if cv2.contourArea(contour) < self.min_area:
33                 contour = None
34         except IndexError:
35             contour = None
36
37         return contour
38
```

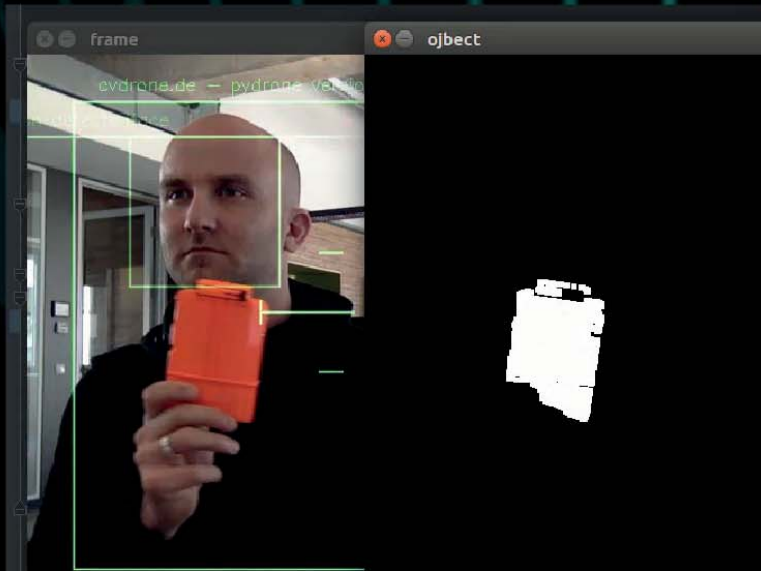
define color boundaries

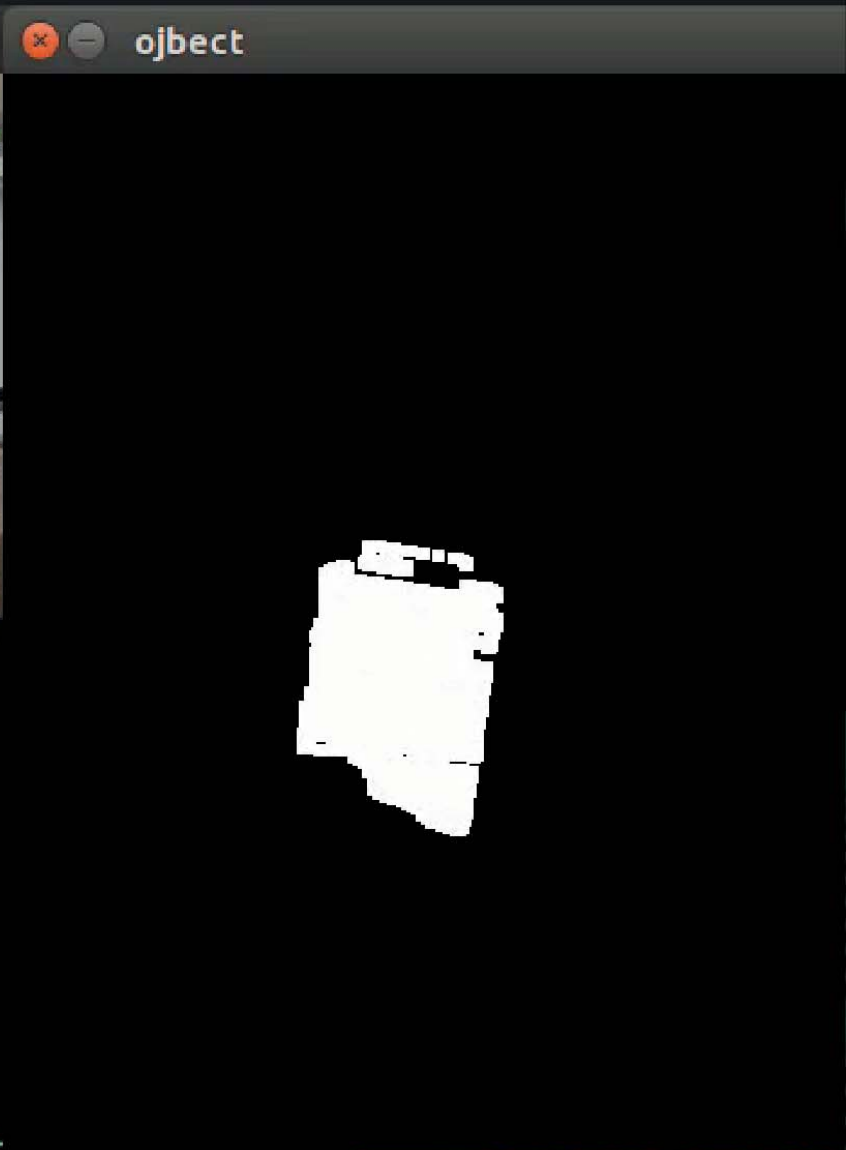
convert frame to HSV

filter only orange

find contours, filter noise  
(must be > min\_area)

video output:





# klassische Algorithmen

nicht ganz so smarte Video-Überwachung verwendet  
**Background Subtraction**

Forschung seit Jahrzehnten  
Immer noch kein One Fits  
All Algorithm

<http://tommessani.com/index.php/video/comparing-background-subtraction-algorithms.html>



was ist eigentlich  
**Convolution**

$$\begin{bmatrix} 35 & 40 & 41 & 45 & 50 \\ 40 & 46 & 42 & 46 & 52 \\ 42 & 46 & 50 & 55 & 55 \\ 48 & 52 & 56 & 58 & 60 \\ 56 & 60 & 65 & 70 & 75 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 40 & 46 & 42 & 46 & 52 \\ 42 & 46 & 50 & 55 & 55 \\ 48 & 52 & 56 & 58 & 60 \\ 56 & 60 & 65 & 70 & 75 \end{bmatrix}$$

source: gimp.org

**Feature Extraction /  
Classifiers**

**Einfach: Frame Differenz**  
**Common: GMM / MoG2**  
**Advanced: SuBSENSE**

Probleme:

- nur statische Kamera
- Lichtänderungen
- Schatten
- Rechenleistung

**Matrix-Multiplikation + Summe**

Verwischen, Schärfen,  
Detektieren ... damit kann  
man extrem viel machen

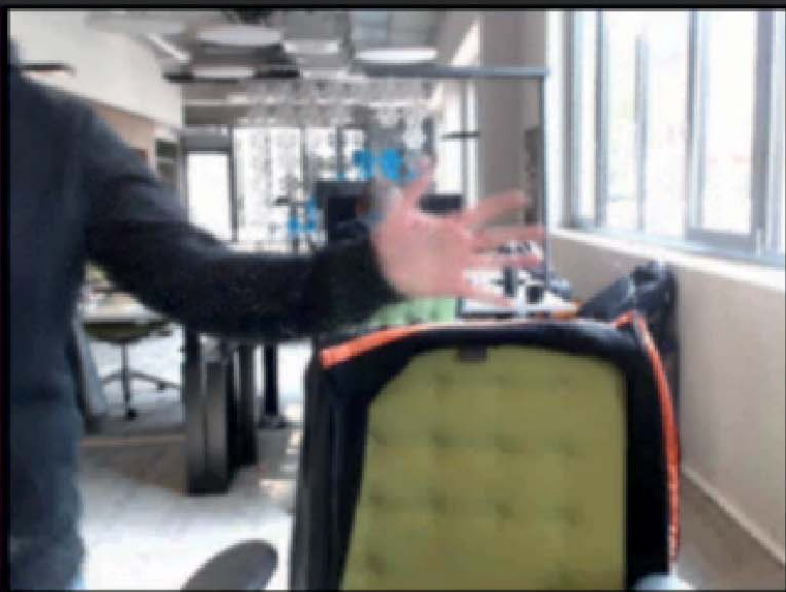
**Haar Cascade Classifiers**

**HOG**

**Local Binary Patterns**



user@ubuntu: ~/opencv/bgslibrary



R - Repeat



using: <https://github.com/andrewssobral/bgslibrary>

# klassische Algorithmen

nicht ganz so smarte Video-Überwachung verwendet  
**Background Subtraction**

Forschung seit Jahrzehnten  
Immer noch kein One Fits  
All Algorithm

<http://tommessani.com/index.php/video/comparing-background-subtraction-algorithms.html>



was ist eigentlich  
**Convolution**

$$\begin{bmatrix} 35 & 40 & 41 & 45 & 50 \\ 40 & 46 & 42 & 46 & 52 \\ 42 & 46 & 50 & 55 & 55 \\ 48 & 52 & 56 & 58 & 60 \\ 56 & 60 & 65 & 70 & 75 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 40 & 46 & 42 & 46 & 52 \\ 42 & 46 & 50 & 55 & 55 \\ 48 & 52 & 56 & 58 & 60 \\ 56 & 60 & 65 & 70 & 75 \end{bmatrix}$$

source: gimp.org

**Feature Extraction /  
Classifiers**

**Einfach: Frame Differenz**  
**Common: GMM / MoG2**  
**Advanced: SuBSENSE**

Probleme:

- nur statische Kamera
- Lichtänderungen
- Schatten
- Rechenleistung

**Matrix-Multiplikation + Summe**

Verwischen, Schärfen,  
Detektieren ... damit kann  
man extrem viel machen

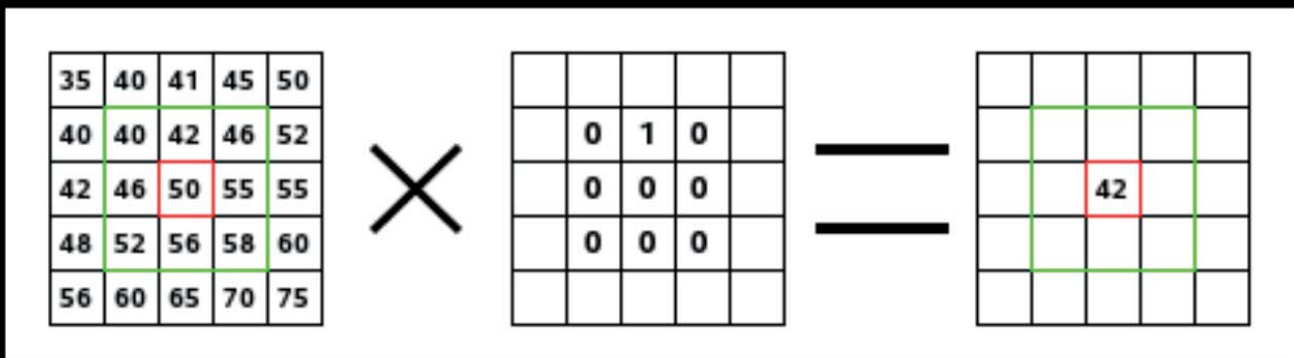
**Haar Cascade Classifiers**

**HOG**

**Local Binary Patterns**

was ist eigentlich

# Convolution



source: gimp.org

# Feature Ex

# klassische Algorithmen

nicht ganz so smarte Video-Überwachung verwendet  
**Background Subtraction**

Forschung seit Jahrzehnten  
Immer noch kein One Fits  
All Algorithm

<http://tommessani.com/index.php/video/comparing-background-subtraction-algorithms.html>



was ist eigentlich  
**Convolution**

$$\begin{bmatrix} 35 & 40 & 41 & 45 & 50 \\ 40 & 46 & 42 & 46 & 52 \\ 42 & 46 & 50 & 55 & 55 \\ 48 & 52 & 56 & 58 & 60 \\ 56 & 60 & 65 & 70 & 75 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 40 & 46 & 42 & 46 & 52 \\ 42 & 46 & 50 & 55 & 55 \\ 48 & 52 & 56 & 58 & 60 \\ 56 & 60 & 65 & 70 & 75 \end{bmatrix}$$

source: gimp.org

**Feature Extraction /  
Classifiers**

**Einfach: Frame Differenz**  
**Common: GMM / MoG2**  
**Advanced: SuBSENSE**

Probleme:

- nur statische Kamera
- Lichtänderungen
- Schatten
- Rechenleistung

**Matrix-Multiplikation + Summe**

Verwischen, Schärfen,  
Detektieren ... damit kann  
man extrem viel machen

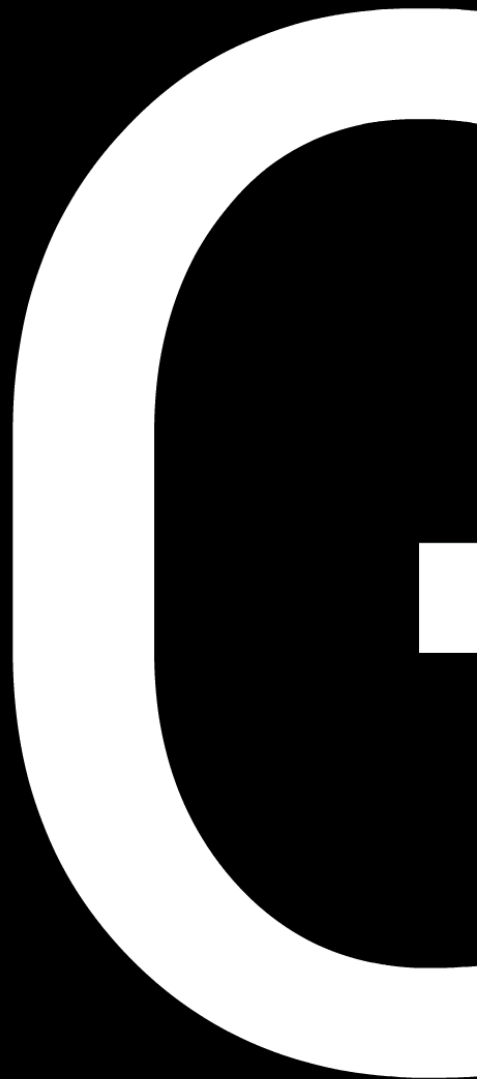
**Haar Cascade Classifiers**

**HOG**

**Local Binary Patterns**



source: dlib.net



# Detektoren einfache

Sliding Window



Multi Scale



Gesichts-Erkennung

## Haar Cascade Classifier

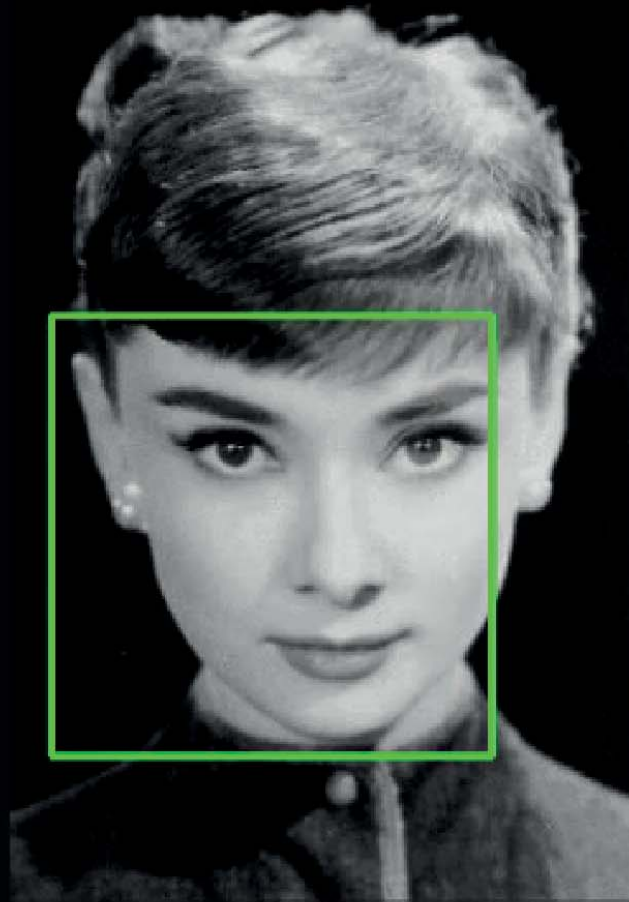
- relativ alt (Viola Jones ~2000)
- ziemlich schnell (läuft auf Raspberry Pi)
- viele Fehler in der Erkennung

## (Default) HOG Detector

- weniger falsche Detektionen
- braucht viel mehr Rechenleistung
- > optimieren für Regions of Interest (ROI)



source: [pyimagesearch.com](http://pyimagesearch.com)



# Detektoren einfache

Sliding Window



Multi Scale



Gesichts-Erkennung

## Haar Cascade Classifier

- relativ alt (Viola Jones ~2000)
- ziemlich schnell (läuft auf Raspberry Pi)
- viele Fehler in der Erkennung

## (Default) HOG Detector

- weniger falsche Detektionen
- braucht viel mehr Rechenleistung
- > optimieren für Regions of Interest (ROI)





# Demo Video erklärt - Schritt 2

- ... Gesichts-Erkennung nur in "Region of Interest"
- ... das spart Rechenleistung
- ... und führt zu weniger False Detections

## Looking for faces, near detected marker object

```
class FaceDetector:
    def __init__(self, scale=1.25, num=3):
        self.cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
        self.scale = scale
        self.num = num

    def detect(self, frame):
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        rois = self.cascade.detectMultiScale(gray, self.scale, self.num)
        return rois

    def get_region_in_rect(self, x1, y1, x2, y2):
        margin_x = 2
        margin_y = 2
        x1 -= margin_x
        y1 -= margin_y
        x2 = x1 + w + margin_x
        y2 = y1 + h + margin_y
        return x1, y1, x2, y2

    def in_region_in_rect(self, face):
        x1, y1, x2, y2 = get_region(marker)
        tx1, ty1, tx2, ty2 = face
        logging.debug('face')
        if (x2 < tx1) and ((tx1 + ty) < x2) and (y1 < ty1) and ((ty1 + 70) < y2):
            return True
        return False

while True:
    ret, frame = cap.read()
    markers = marker_detector.detect(frame)
    if marker is not None:
        marker = cv2.boundingRect(marker)
        faces = face_detector.detect(frame)
        for face in faces:
            if in_region(marker, face):
                hud.mark_rect(frame, face)
                x1, y1, w, h = face
                hud.mark_rect(frame, [(x1, y1, x1+w, y1+h)])
cv2.imshow('video', frame)
```

video output:



# Looking for faces, near detected marker object

```
14
15 class FaceDetector:
16     def __init__(self, scale=1.25, num=3):
17         self.cascade = cv2.CascadeClassifier('/usr/share/opencv/haarcascades/haarcascade_frontalface
18         self.scale = scale
19         self.num = num
20
21     def detect(self, frame):
22         gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
23         rois = self.cascade.detectMultiScale(gray, self.scale, self.num)
24         return rois
25
26     def get_region(marker):
27         x1, y1, w, h = marker
28         x1 -= margin_x
29         y1 -= margin_y
30         x2 = x1 + w + 2 * margin_x
31         y2 = y1 + h + margin_y
32         return x1, y1, x2, y2
33
34     def in_region(marker, face):
35         x1, y1, x2, y2 = get_region(marker)
36         fx1, fy1, fw, fh = face
37         logging.debug(face)
38         if (x1 < fx1) and ((fx1 + fw) < x2) and (y1 < fy1) and ((fy1 + fh) < y2):
39             return True
40         return False
41
42 while True:
43     ret, frame = cam.read()
44     marker = marker_detector.detect(frame)
45     if marker is not None:
46         marker = cv2.boundingRect(marker)
47         faces = face_detector.detect(frame)
48         for face in faces:
49             if in_region(marker, face):
50                 hud.mark_rois(frame, [face, time])
51                 x1, y1, w, h = face
52                 hud.mark_rois(frame, [(p_x1, p_y1, p_x2, p_y2)])
53
54 cv2.imshow("frame", frame)
55
```

init face detector

read frame from video source

look for marker objects

if marker found, search for faces

if face near marker, mark person

video output:





# Neuronale Netze

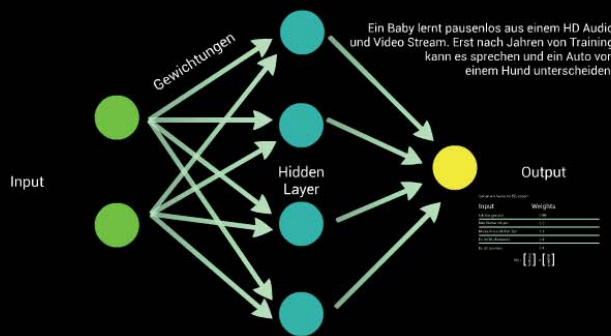
Alle gezeigten klassischen Methoden haben ein Problem:

- der Computer versteht nicht wirklich was im Bild passiert
- alles ist ein "handcrafted overfitting" - passend für eine Szene

Mit ImageNet kam die Wende:

Heute erreichen Algorithmen  
"Super-Human-Performance"

Neuronale Netze ermöglichen stetiges Lernen aus Daten.

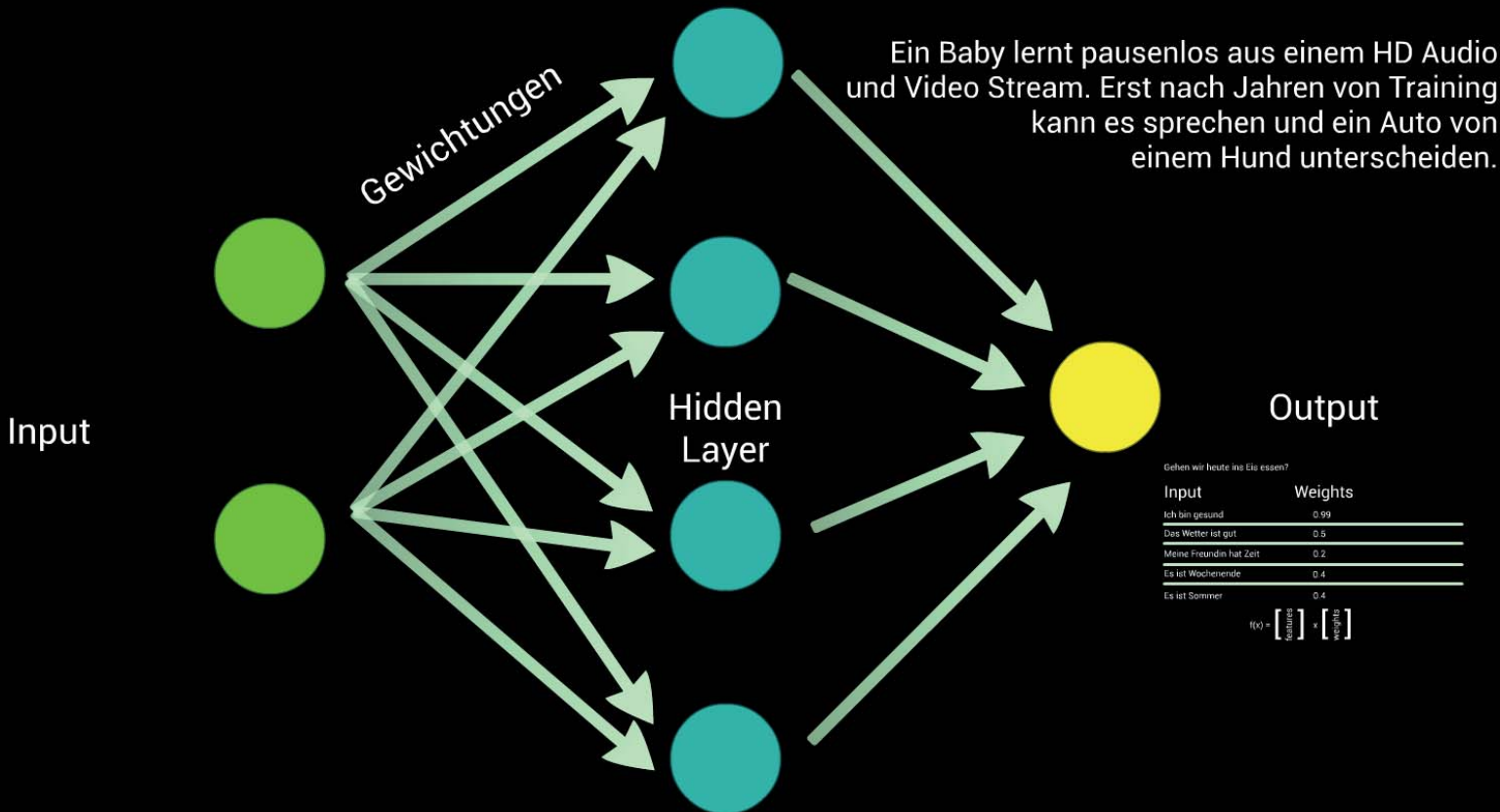


Ein Baby lernt pausenlos aus einem HD Audio und Video Stream. Erst nach Jahren von Training kann es sprechen und ein Auto von einem Hund unterscheiden.

• ein neuronales Netz nähert beliebige mathematische Funktionen an  
automatisch & iterativ

• es kann Milliarden von Parametern LERNEN

• das ist wie Milliarden von if-else-Bedingungen zu lernen



- ein neuronales Netz nähert beliebige mathematische Funktionen an  
**automatisch & iterativ**
- es kann Milliarden von Parametern **LERNEN**
  - das ist wie Milliarden von if-/else Bedingungen zu lernen

# Output

Gehen wir heute ins Eis essen?

## Input

## Weights

Ich bin gesund

0.99

Das Wetter ist gut

0.5

Meine Freundin hat Zeit

0.2

Es ist Wochenende

0.4

Es ist Sommer

0.4

$$f(x) = \begin{bmatrix} \text{features} \end{bmatrix} \times \begin{bmatrix} \text{weights} \end{bmatrix}$$

# Deep Learning

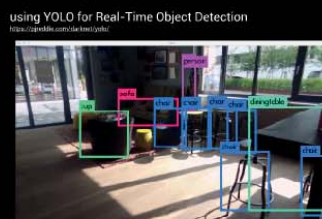
Damit werden beeindruckende neue Applikationen möglich ...

Aber: Man braucht mehr **POWER** (und mehr **Daten**).

## Style Transfer



## Object Detection



## Pose Estimation



Es gibt viele weitere beeindruckende Beispiele ...

Nach Jahrzehnten der Forschung beginnen Computer zu verstehen, was in Bildern zu sehen ist. Die cambrische Explosion resultierte unter anderem aus der Entwicklung des Sehens. Die heutige Entwicklung läuft aber um ein Vielfaches schneller als in der Evolution ...

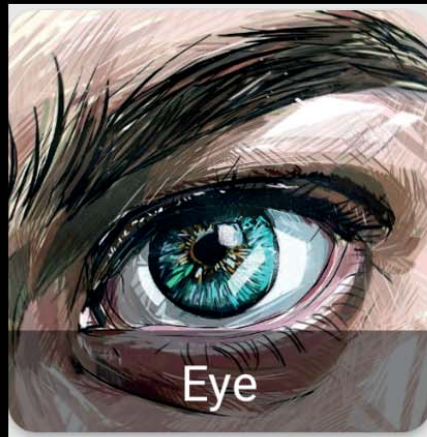
How we interact with machines will soon change ...



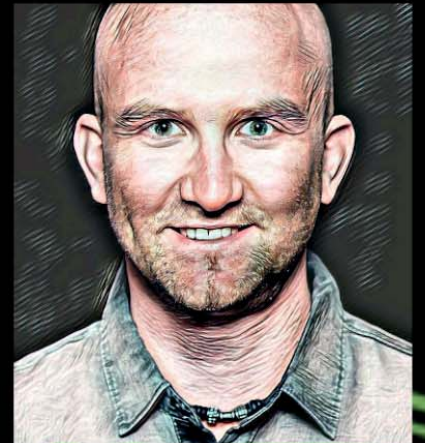
# Style Transfer



+



=



Source: [Ruder et al. 2016]

Star Wars



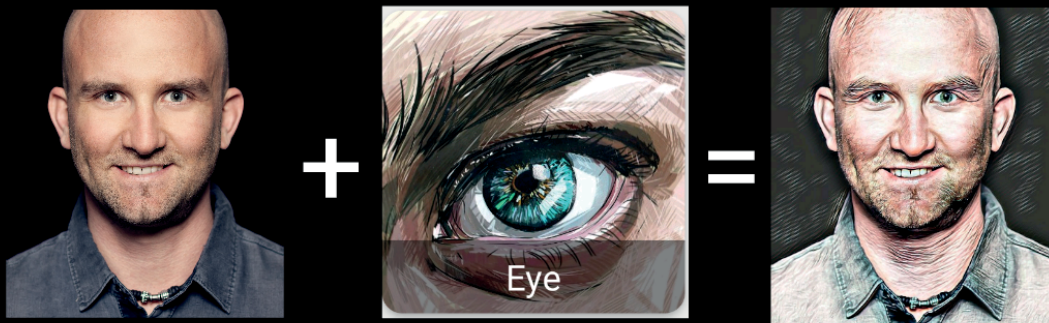
... also works with music  
... and with text!  
... maybe soon with code?!?

Source: [Ruder et al. 2016]

## Star Wars



# Style Transfer



Source: (Fidler et al. 2016)

Star Wars



... also works with music  
... and with text!  
... maybe soon with code!?

Vergleich Performance: Klassisch -> Deep Learning

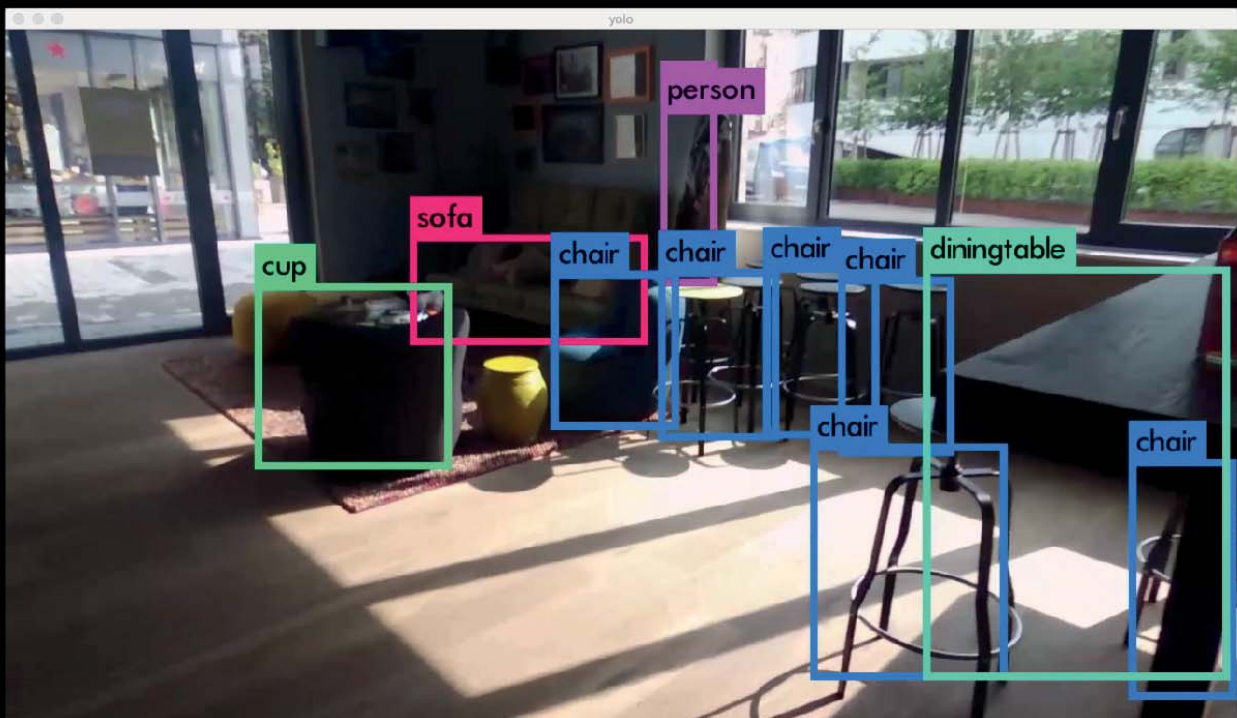


# Es gibt viele weitere beei

# Object Detection

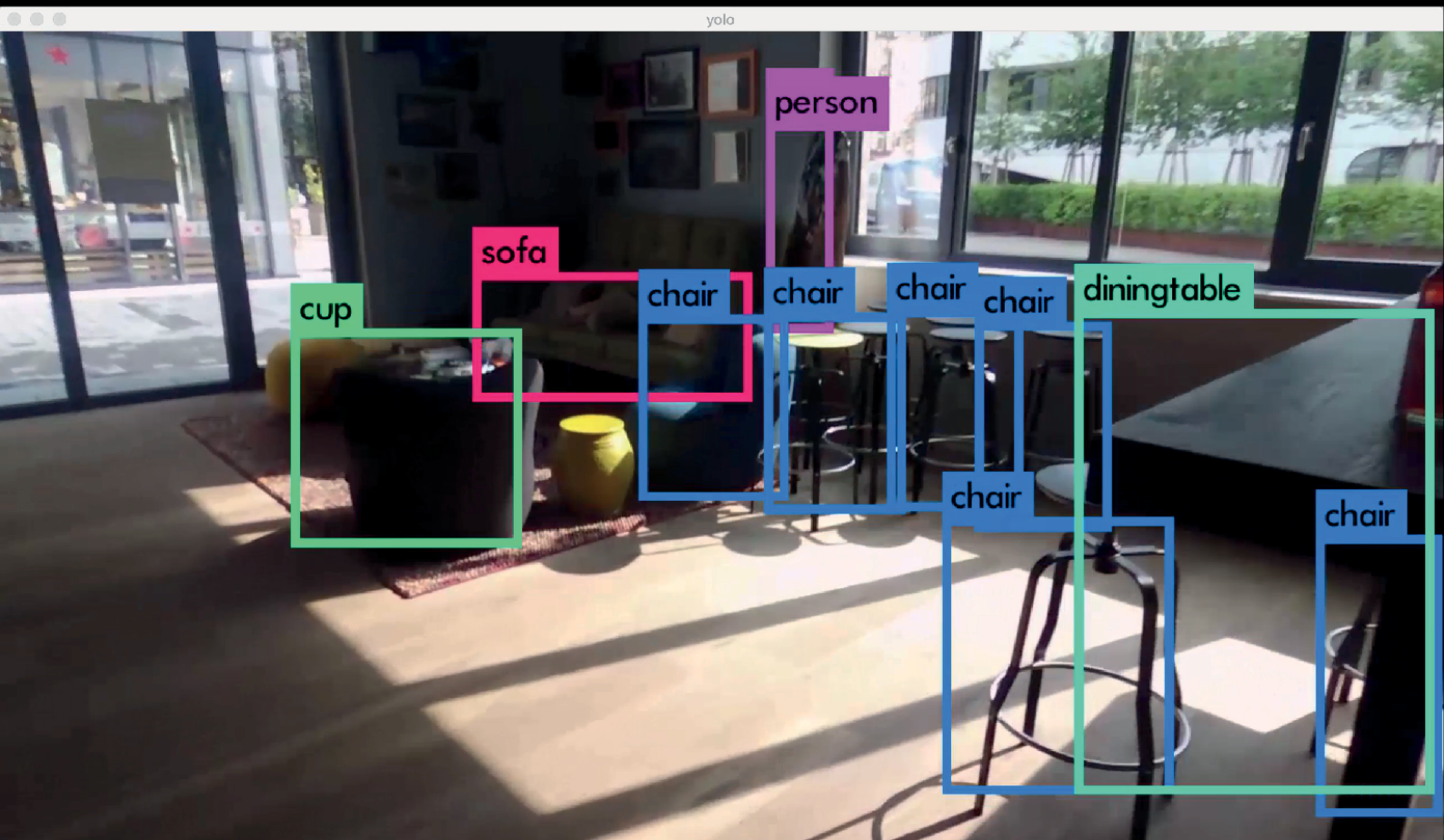
using YOLO for Real-Time Object Detection

<https://pjreddie.com/darknet/yolo/>



# Using YOLO for Real-Time Object Detection

<https://pjreddie.com/darknet/yolo/>



# Pose Estimation

## Demo: Realtime Multi Person Pose Estimation

[https://github.com/ZheC/Realtime\\_Multi-Person\\_Pose\\_Estimation](https://github.com/ZheC/Realtime_Multi-Person_Pose_Estimation)



Source: <https://www.youtube.com/watch?v=2DlQUX11YaY>

# Demo: Realtime Multi Person Pose Estimation

[https://github.com/ZheC/Realtime\\_Multi-Person\\_Pose\\_Estimation](https://github.com/ZheC/Realtime_Multi-Person_Pose_Estimation)



Source: <https://www.youtube.com/watch?v=2DiQUX11YaY>



# Deep Learning

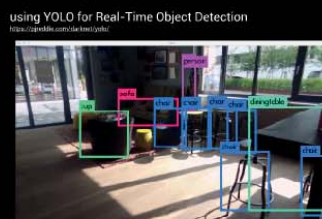
Damit werden beeindruckende neue Applikationen möglich ...

**Aber: Man braucht mehr POWER (und mehr Daten).**

## Style Transfer



## Object Detection



## Pose Estimation



Es gibt viele weitere beeindruckende Beispiele ...

Nach Jahrzehnten der Forschung beginnen Computer zu verstehen, was in Bildern zu sehen ist. Die cambrische Explosion resultierte unter anderem aus der Entwicklung des Sehens. Die heutige Entwicklung läuft aber um ein Vielfaches schneller als in der Evolution ...

**How we interact with machines will soon change ...**



## Vergleich Performance: Klassisch -> Deep Learning



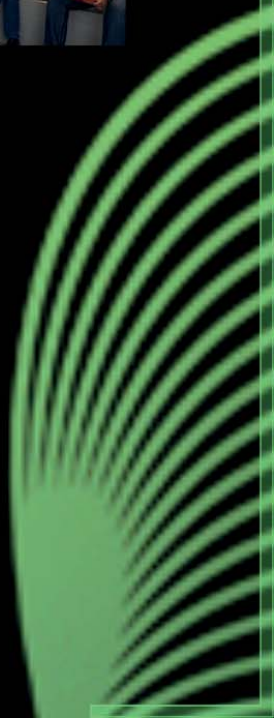
Haar Cascades (~2001)



HOG (~2005)



MTCNN (~2016)





Haar Cascades (~2001)



HOG (~2005)



MTCNN (~2016)

# Deep Learning

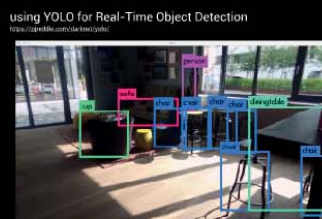
Damit werden beeindruckende neue Applikationen möglich ...

Aber: Man braucht mehr **POWER** (und mehr **Daten**).

## Style Transfer



## Object Detection



## Pose Estimation



Es gibt viele weitere beeindruckende Beispiele ...

Nach Jahrzehnten der Forschung beginnen Computer zu verstehen, was in Bildern zu sehen ist. Die cambrische Explosion resultierte unter anderem aus der Entwicklung des Sehens. Die heutige Entwicklung läuft aber um ein Vielfaches schneller als in der Evolution ...

How we interact with machines will soon change ...

# Computer Vision ... finally takes off!

- Disziplin der Informatik gibt es seit Jahrzenten
- Algorithmen und Methodologien sind altbekannt
- Aber erst durch Cloud, MEHR DATEN und GPUs gibt es neue Möglichkeiten
- der Vortrag zeigt den Pfad von einfachen Werkzeugen und deren Vor- und Nachteilen hin zu Neuronalen Netzen



Follow us on <https://youtube.com/c/codecentricAI>

Team codecentric - Karlsruhe

Meet us auf:

DAS FEST

## Fragen ???



### FAZIT

Klassische Methoden funktionieren "unter engen Labor-Bedingungen"  
... sie lernen nicht dazu.

Deep Learning ist keine Magie  
Man muss nicht Mathematik studiert haben, um es anzuwenden  
... Detektoren werden immer besser und besser.

Neue Möglichkeiten:

- smarte Sensoren
- Überwachung UND Datenschutz
- neue Formen der Interaktion
- Automatische Auswertungen

Oliver Moser - 2018



@moseroli

[oliver.moser@codecentric.de](mailto:oliver.moser@codecentric.de)

# FAZIT

Klassische Methoden funktionieren "unter engen Labor-Bedingungen"

... sie lernen nicht dazu.

Deep Learning ist keine Magie

Man muss nicht Mathematik studiert haben, um es anzuwenden

... **Detektoren werden immer besser und besser.**

Neue Möglichkeiten:

- smarte Sensoren
  - Überwachung UND Datenschutz
    - neue Formen der Interaktion
      - Automatische Auswertungen

- der Vortrag zeigt den Pfad von einfachen und deren Vor- und Nachteilen hin zu Netzen

Follow us on <https://youtube.com/c/codecentricAI>

Team codecentric - Karlsruhe

Meet us auf:

**DAS FEST**

# Fragen ????





# Computer Vision ... finally takes off!

- Disziplin der Informatik gibt es seit Jahrzenten
- Algorithmen und Methodologien sind altbekannt
- Aber erst durch Cloud, MEHR DATEN und GPUs gibt es neue Möglichkeiten
- der Vortrag zeigt den Pfad von einfachen Werkzeugen und deren Vor- und Nachteilen hin zu Neuronalen Netzen



Follow us on <https://youtube.com/c/codecentricAI>

Team codecentric - Karlsruhe

Meet us auf:

DAS FEST

## Fragen ???



### FAZIT

Klassische Methoden funktionieren "unter engen Labor-Bedingungen"  
... sie lernen nicht dazu.

Deep Learning ist keine Magie  
Man muss nicht Mathematik studiert haben, um es anzuwenden  
... Detektoren werden immer besser und besser.

Neue Möglichkeiten:

- smarte Sensoren
- Überwachung UND Datenschutz
- neue Formen der Interaktion
- Automatische Auswertungen

Oliver Moser - 2018



@moseroli

oliver.moser@codecentric.de