Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES

inovex

AI Labor - Wintersemester 2019/20

Computer Vision
1. Termin

Robin Baumann, Matthias Richter

Karlsruhe, 04. Oktober 2019

Agenda für Heute

- Allgemeines
- Einführung Python, Jupyter
- Theorie zu den Übungen des 1. Termins
- Gruppenfindung
- Praktischer Teil

Modus

- 14 Termine, jeweils 2 Blöcke
- 4 ECTS, 4 SWS

- Gruppenarbeit (3 Personen)

- Bewertung
  - Diskussion und Vorstellung der Lösungen + Fragen

Teamvorstellung

- Computer Vision (4 Termine)
  - Robin Baumann, Matthias Richter
- Natural Language Processing (5 Termine)
  - Anna Weisshaar, Tilman Wittl
- Reinforcement Learning (5 Termine)
  - Benedikt Hagen, Frederik Martin

Computer Vision

1. Python, Jupyter, Grundlagen Computer Vision
2. Texterkennung
3. Objektklassifikation
4. Transfer Learning, 3D Deep Learning

Inhalt

Einführung in

- Python
- Jupyter Notebooks


python

Python

“Python is powerful... and fast;
plays well with others;
runs everywhere;
is friendly & easy to learn;
is Open.”

- Anfang der 1990er von Guido van Rossum entwickelt

https://www.python.org/about/

Python

- sehr universell
- interpretierte Sprache
- multiparadigmatisch
- dynamische Typisierung
- wird oft als Skriptsprache benutzt
- sehr gute Lesbarkeit
- klare und übersichtliche Syntax

https://www.python.org/about/

Python

Zen of Python : 19 Prinzipien

- Beautiful is better than ugly
- Explicit is better than implicit
- Simple is better than complex
- Complex is better than complicated
- Readability counts
- …

Class & Inheritance in Java:

```java
class Animal{
    private String name;
    public Animal(String name){
        this.name = name;
    }
    public void saySomething(){
    System.out.println("I am" + name);
    }
}

class Dog extends Animal{
    public Dog(String name) {
        super(name);
    }
    public void saySomething(){
    System.out.println("I can bark");
    }
}

public class Main {
public static void main(String[] args)
    {
    Dog dog = new Dog("Chiwawa");
        dog.saySomething();
    }
}
```

Class & Inheritance in Python:

```python
class Animal():

    def __init__(self, name):
        self.name = name

    def saySomething(self):
        print "I am " + self.name

class Dog(Animal):
    def saySomething(self):
        print "I am "+ self.name\
        + ", and I can bark"

dog = Dog("Chiwawa")
dog.saySomething()
```

Python

Wer benutzt Python?

- Youtube
- Dropbox
- Google
- Quora
- Instagram
- BitTorrent
- Spotify
- Reddit
- Pinterest
- BitBucket

Python

Aber

- keine statische Typprüfung!
- whitespace sensitive!

Python

Python 2 vs Python 3

- Python 3 erschien 2008 nach längere Entwicklungszeit
- tiefgreifende Änderungen an der Sprache
- teilweise inkompatibel zu früheren Versionen
- parallele Unterstützung bis Ende 2019
- ab 2020 wird Python 2 nicht mehr unterstützt

Python

Paketverwaltung

- Pakete können u.a. mit *distutils* oder *setuptools* erstellt werden
- Paketmanager üblicherweise *pip*
- Offizielles Repository für 3rd-party Pakete *PyPI* hat inzwischen über 167k Projekte (Stand: 06.02.19)
- Virtuelle Umgebungen mit *virtualenv* oder *venv* zur sauberen Isolierung von Applikationen und Vermeidung von Versionschaos

Python

Paketverwaltung

- ist generell etwas langsamer als kompilierte Sprachen
- Fokus auf Lesbarkeit, statt verfrühte Optimierung
- *Cython* Erweiterung erlaubt Übersetzung in oder Einbindung von C/C++ Code

Python

Bibliotheken

- *NumPy* scientific computing
- *pandas* data analysis
- *matplotlib* plotting
- *SciPy* contains core packages for maths, science, engineering

Python

Bibliotheken

- *scikit-learn* traditional machine learning
- *TensorFlow* primarily neural networks (from Google)
- *Keras* deep learning library (high-level api)

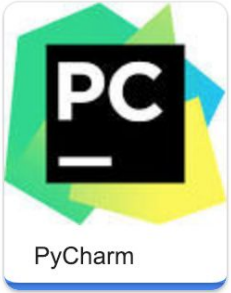
Python

Anaconda

- “data science platform”
- Open-source Distribution von Python/R und vieler Bibliotheken
- *conda* Paket- und Umgebungsverwaltung

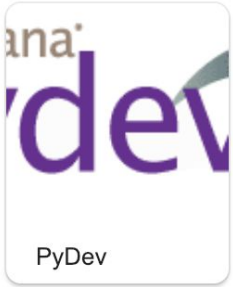
Python

IDEs


PC
PyCharm

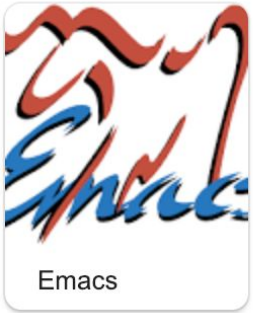

NT ENVIRONMEN
Wing IDE


eric


dev
PyDev


Eclipse


SPYDER
Spyder

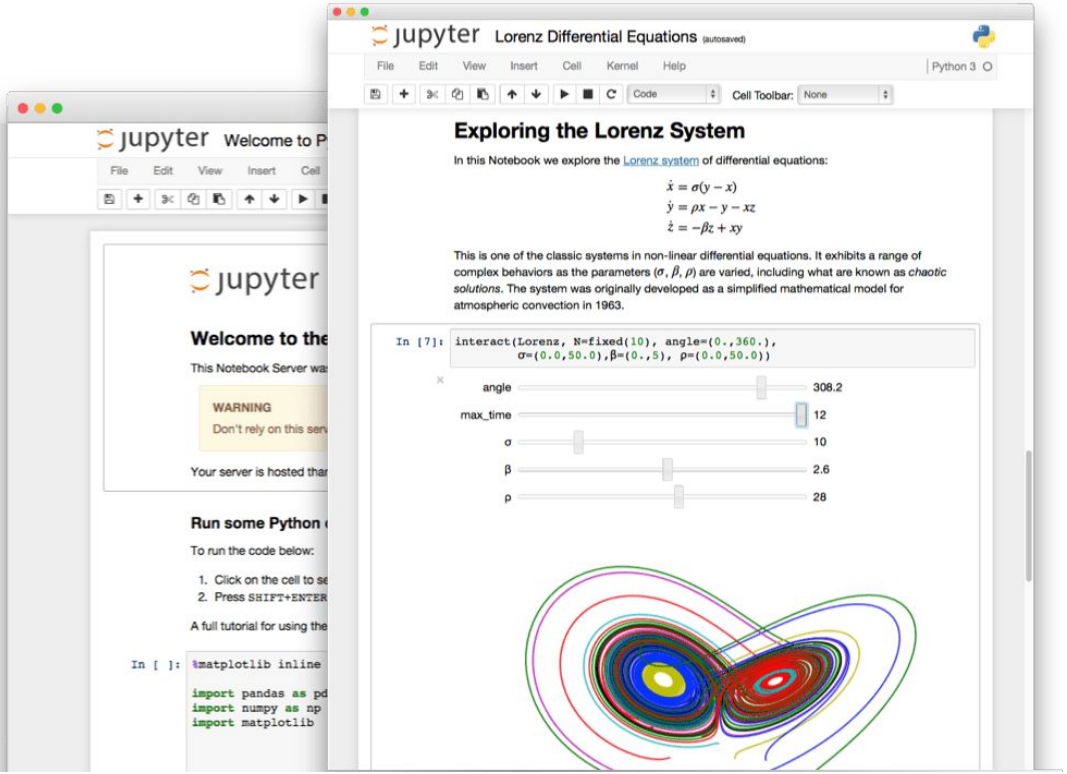

Emacs


Vim


jupyter

Jupyter

- Jupyter Notebook
- Open-source Webapplikation zum Erstellen und Teilen von (Code-) Dokumenten



inovex

Jupyter

- unterstützt über 40 verschiedene Programmiersprachen (Python, R, Scala, Julia, …)
- können einfach mit anderen geteilt werden
- sehr interaktiv; output wird direkt gezeigt
- big data Integration (z.B. mit Spark)

- Installation über *Anaconda* oder *pip*

Jupyter

kleine Live-Demo

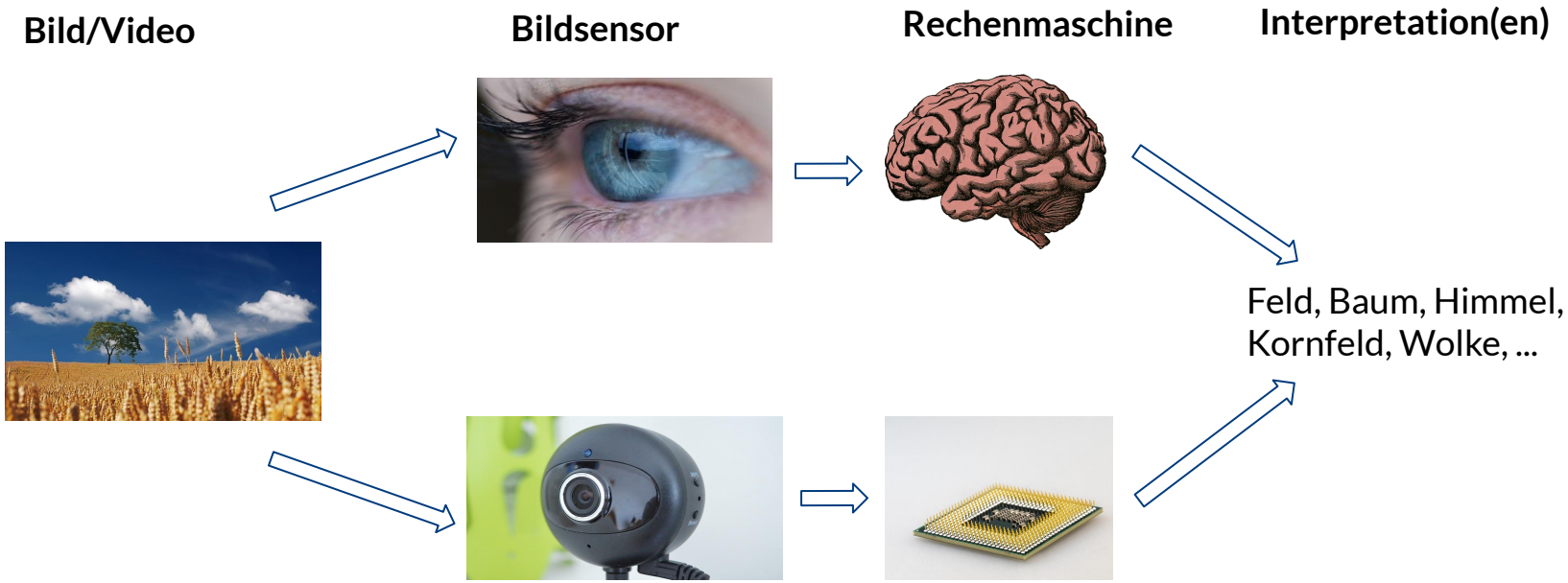
http://localhost:8888

Was ist Computer Vision?

Definition von Microsoft Research:

*"Computer Vision is an research area that studies how to **make computers efficiently perceive, process, and understand visual data** [...]. The ultimate goal is for computers to **emulate the striking perceptual capability of human eyes and brains**, or even to surpass and assist the human in certain ways".*

Was ist Computer Vision?



inovex

Worin liegt die Herausforderung?

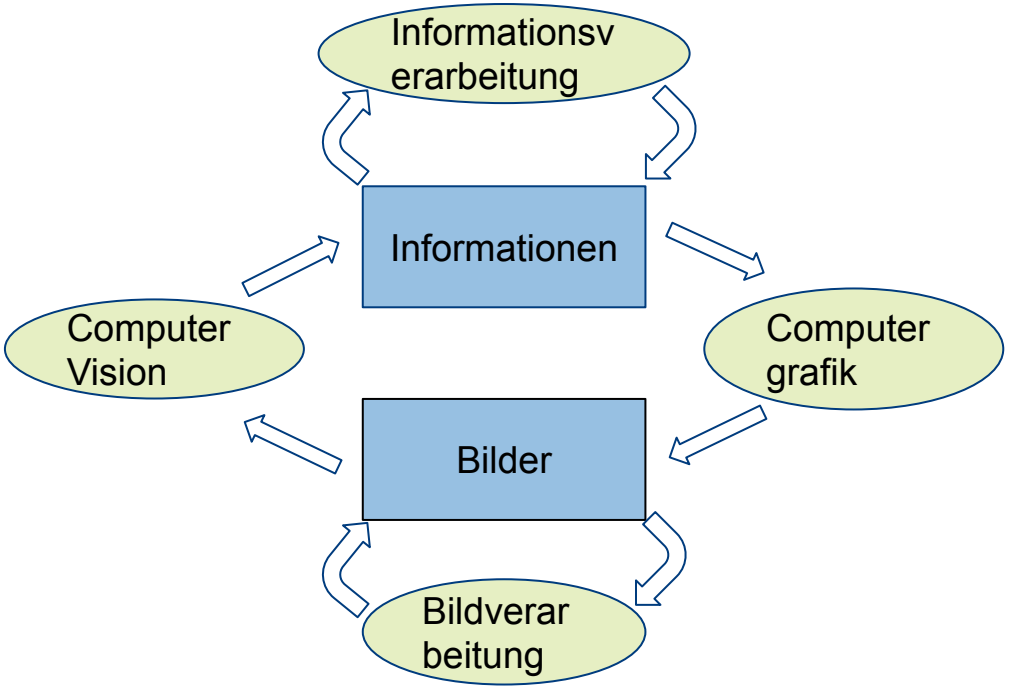
Was wir sehen:

vs.

Was die Maschine “sieht”:

		165	187	209	58	7
	14	125	233	201	98	159
253	144	120	251	41	147	204
67	100	32	241	23	165	30
209	118	124	27	59	201	79
210	236	105	169	19	218	156
35	178	199	197	4	14	218
115	104	34	111	19	196	
32	69	231	203	74		

Abgrenzung

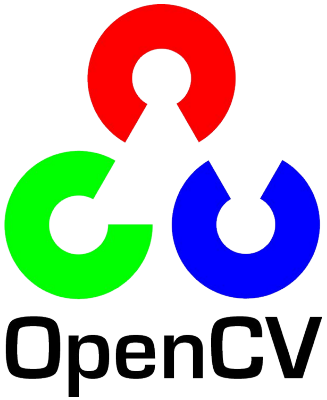

Informationsv
erarbeitung
Informationen
Computer
Vision
Computer
grafik
Bilder
Bildverar
beitung

Bildverarbeitung

- Datenbereinigung
- Vorverarbeitung für Klassifikation
- Augmentation
- Merkmalsextraktion

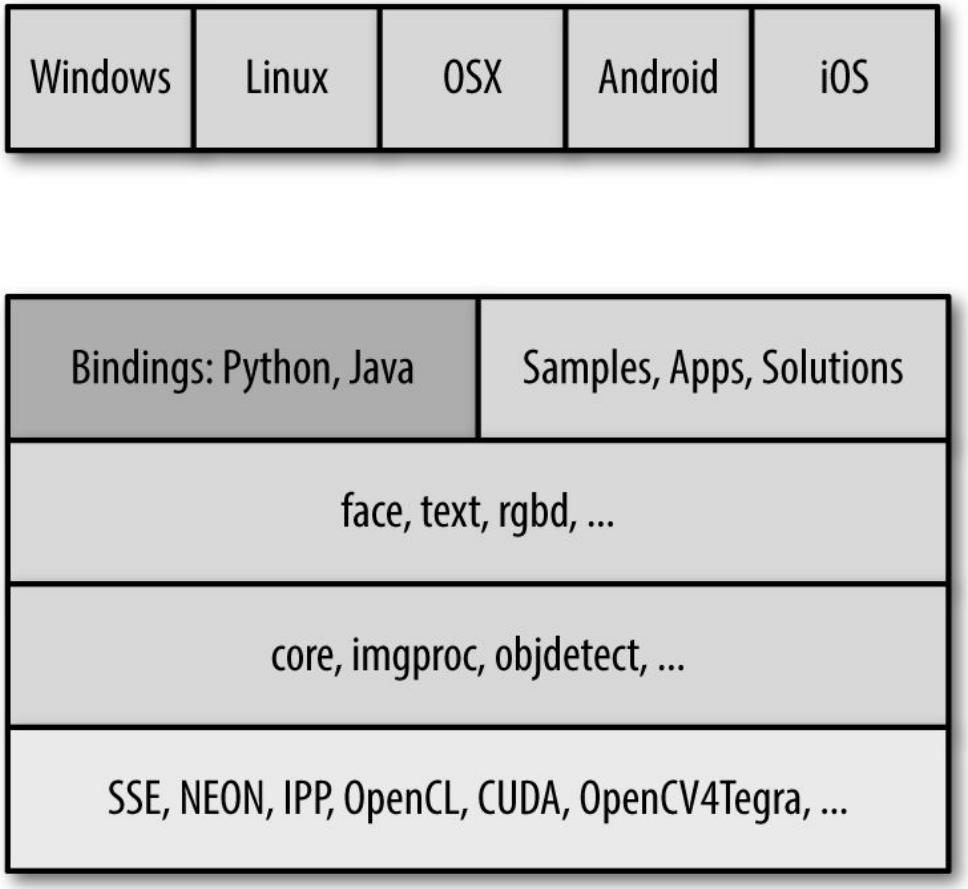
OpenCV

(https://opencv.org/)



- Open Source unter BSD Lizenz
- Aktuell in der Version 4.1.1
- C++, Java und Python APIs
- Hocheffiziente C++11 Implementierung gebräuchlicher Algorithmen aus dem Bereich CV
- Hardwarebeschleunigung über z.B. OpenCL möglich

OpenCV



Faltung (Convolution)

Kontinuierlich, 1D: $(f*g)(x) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$

Diskret, 2D: $F(x,y) = (f**g)(x,y) = \sum_{s=-\infty}^{\infty}\sum_{t=-\infty}^{\infty} f(s,t)g(x-s,y-t)$

OpenCV

```
dst = cv2.filter2D(src, ddepth, kernel, [...])
```


inovex

Faltung (Convolution)

Faltungskernel

- Größe des Kernels beeinflusst Größe der Ausgabe
  => “Padding” der Eingabe, damit Ausgabebild dieselbe Dimensionalität behält

- Nützliche Visualisierungen:
  http://setosa.io/ev/image-kernels/

Thresholding

- Für viele Aufgaben werden Binärbilder benötigt
  - Schwarz: Hintergrund
  - Weiß: Vordergrund
- Unterscheidung in:
  - globales Thresholding

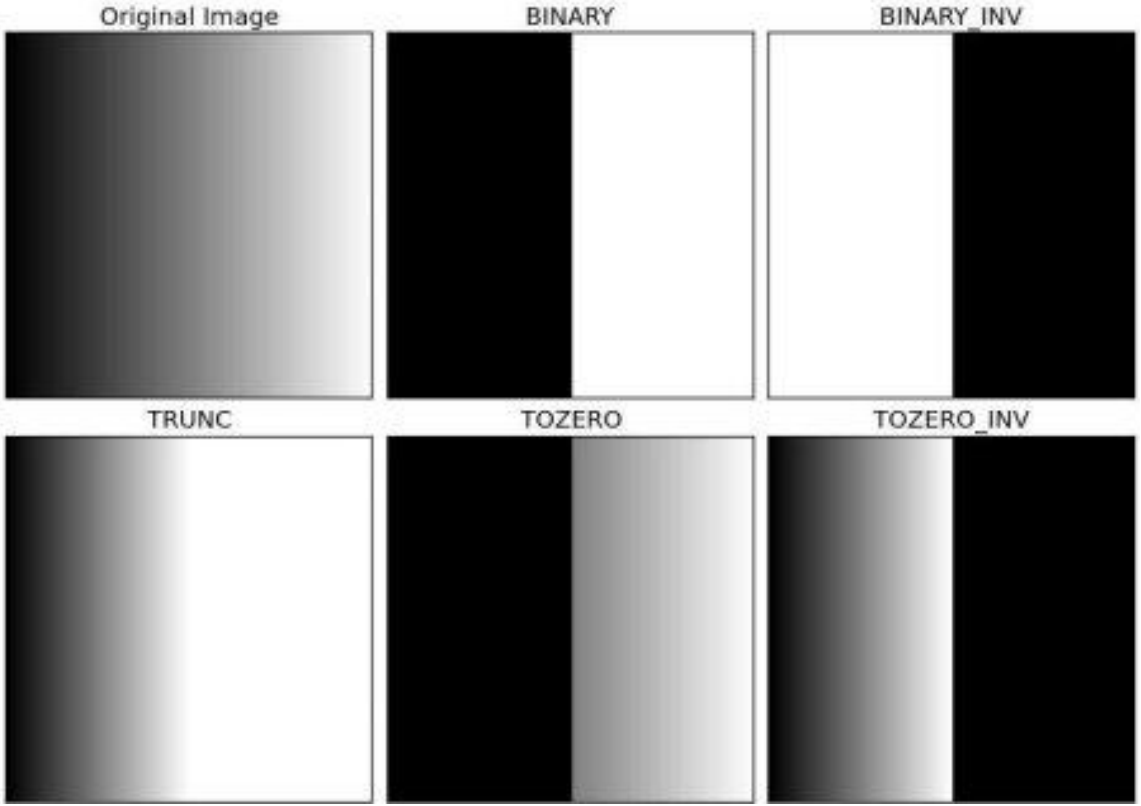
    


    ```
    cv2.threshold(img, thresh, maxVal, type)
    ```
  - adaptives Thresholding


    ```
    cv2.adaptiveThreshold(src, maxValue,
    adaptiveMethod, thresholdType, blockSize, C)
    ```

Threshold Types



Quelle: https://docs.opencv.org/4.0.1/d7/d4d/tutorial_py_thresholding.html (21.03.2019)

Konturen

- Können als eine Menge an interessanten Punkten gleicher Intensität entlang einer Objektkante beschrieben werden
- Am besten können sie mit OpenCV auf Binärbildern erkannt werden.
- Mit ihrer Hilfe lassen sich Merkmale zur Identifikation von Objekten definieren (Contour Properties)


```
cnts, hier = cv2.filter2D(src, mode, method, [...])
```

Konturmerkmale

Eingebaut in OpenCV

- Fläche der Kontur:
  - `area = cv2.contourArea(cnt)`
- Konvexe Hülle:
  - `hull = cv2.convexHull(cnt)`
- Bounding box:
  - `x,y,w,h = cv2.boundingRect(cnt)`
- Approximierte Linie durch alle Punkte:
  - `line = cv2.fitLine(points, distType, param, reps, aeps)`

Konturmerkmale

… zum selbst berechnen ;)

- Aspect Ratio = Breite / Höhe
- Extent: Anteil an Konturfläche zur Bounding box
- Solidity: Anteil der konvexen Hülle zur Bounding box
- … und viele andere.

- Referenz: OpenCV Dokumentation (21.03.2019)

Gruppenfindung

- Dreierteams bilden
- Name und Gruppe auf Zettel schreiben

Setup

- Anmeldung am Rechner
  - User: i-lfm-docker
  - Passwort: garten::obst

- Ethernet auf `hskaopen` umstellen
- Wired Settings > Security:
  - [x] No CA certificate is required
  - `testXYZ` entfernen und mit Studentenkürzel ersetzen
  - Passwort eingeben

Setup

- Terminal öffnen
  - `git clone https://github.com/hskaailabcv/source.git`
  - `cd source`
  - `docker pull hskaailabcv/image:1.0`
  - `docker-compose up`
- Jupyter: *http://localhost:8888*

Feedback

- Google Forms

  https://forms.gle/nikxym47jiUf8v196

Vielen Dank

Robin Baumann
rbaumann@inovex.de

Matthias Richter
mrichter@inovex.de

inovex GmbH
Ludwig-Erhard-Allee 6
76131 Karlsruhe