



Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES



inovex

AI Labor - Sommersemester 2019

1. Termin

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, Stanislav Frolov
- 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. AI auf Android Smartphone

Inhalt

Einführung in

- Python
- Jupyter Notebooks





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

Python

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

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:

```
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:



```
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)

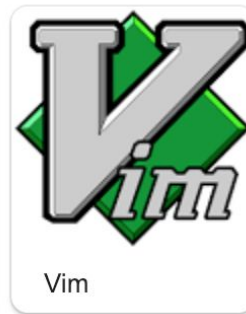
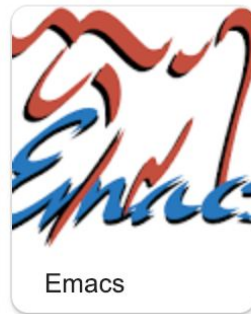
Python

Anaconda

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

Python

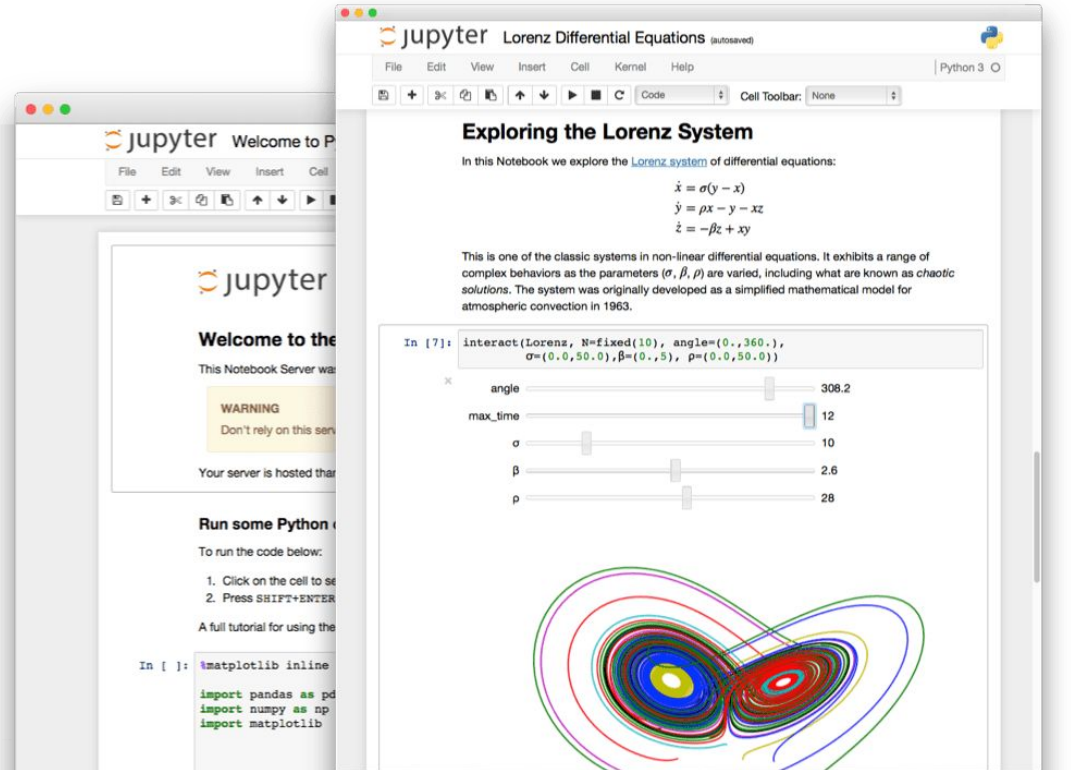
IDEs





Jupyter

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



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

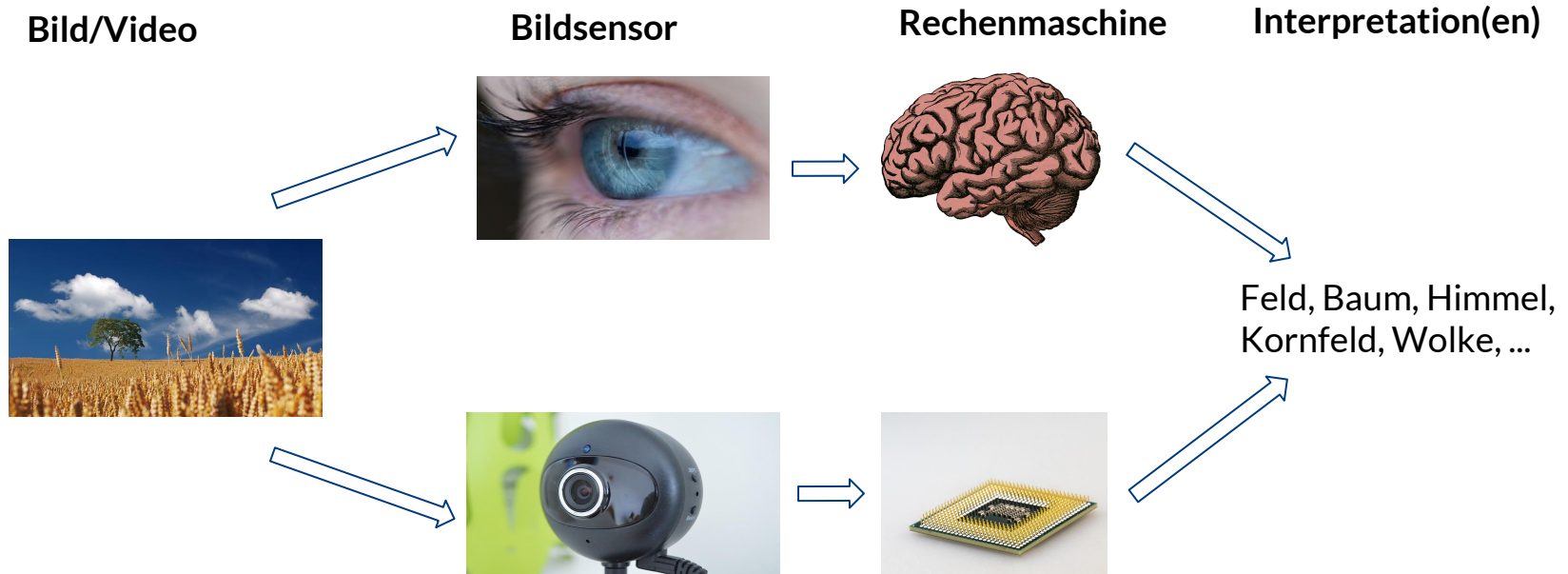
<http://localhost:8888>

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?



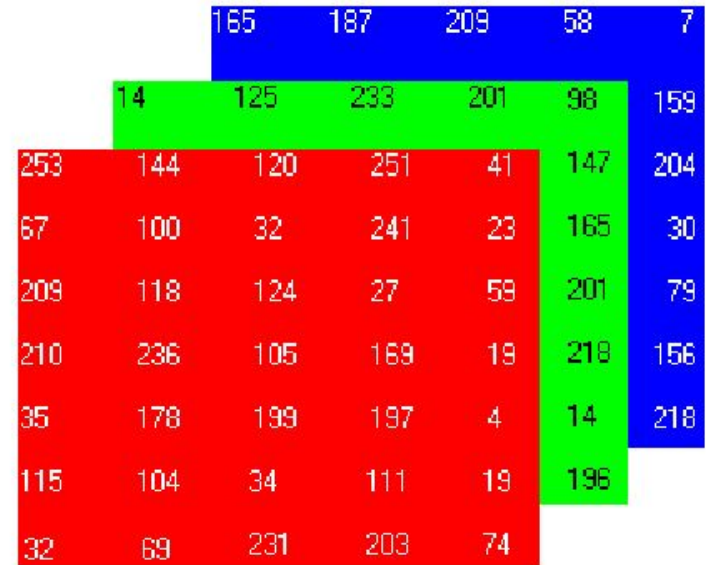
Worin liegt die Herausforderung?

Was wir sehen:

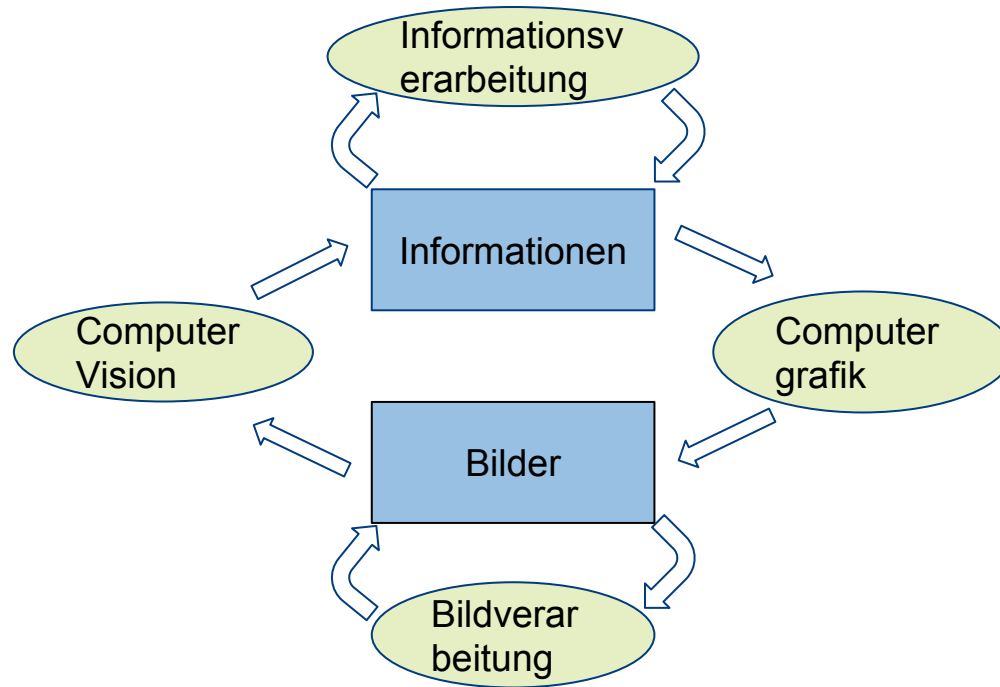


Was die Maschine "sieht":

vs.



Abgrenzung

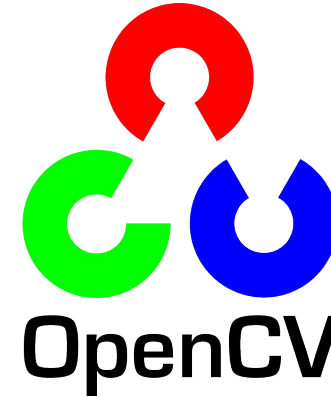


Bildverarbeitung

- Datenbereinigung
- Vorverarbeitung für Klassifikation
- Augmentation
- Merkmalsextraktion

OpenCV

(<https://opencv.org/>)



- Open Source unter BSD Lizenz
- Aktuell in der Version 4.0.1
- C++, Java und Python APIs
- Hocheffiziente C++11 Implementierung gebräuchlicher Algorithmen aus dem Bereich CV
- Hardwarebeschleunigung über z.B. OpenCL möglich

OpenCV

Windows	Linux	OSX	Android	iOS
---------	-------	-----	---------	-----

	Bindings: Python, Java	Samples, Apps, Solutions
OpenCV Contrib	face, text, rgbd, ...	
OpenCV	core, imgproc, objdetect, ...	
OpenCV HAL	SSE, NEON, IPP, OpenCL, CUDA, OpenCV4Tegra, ...	

Faltung (Convolution)

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

Diskret: $F(x, y) = f(x, y) \otimes g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b f(s, t)g(x - s, y - t)$



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


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



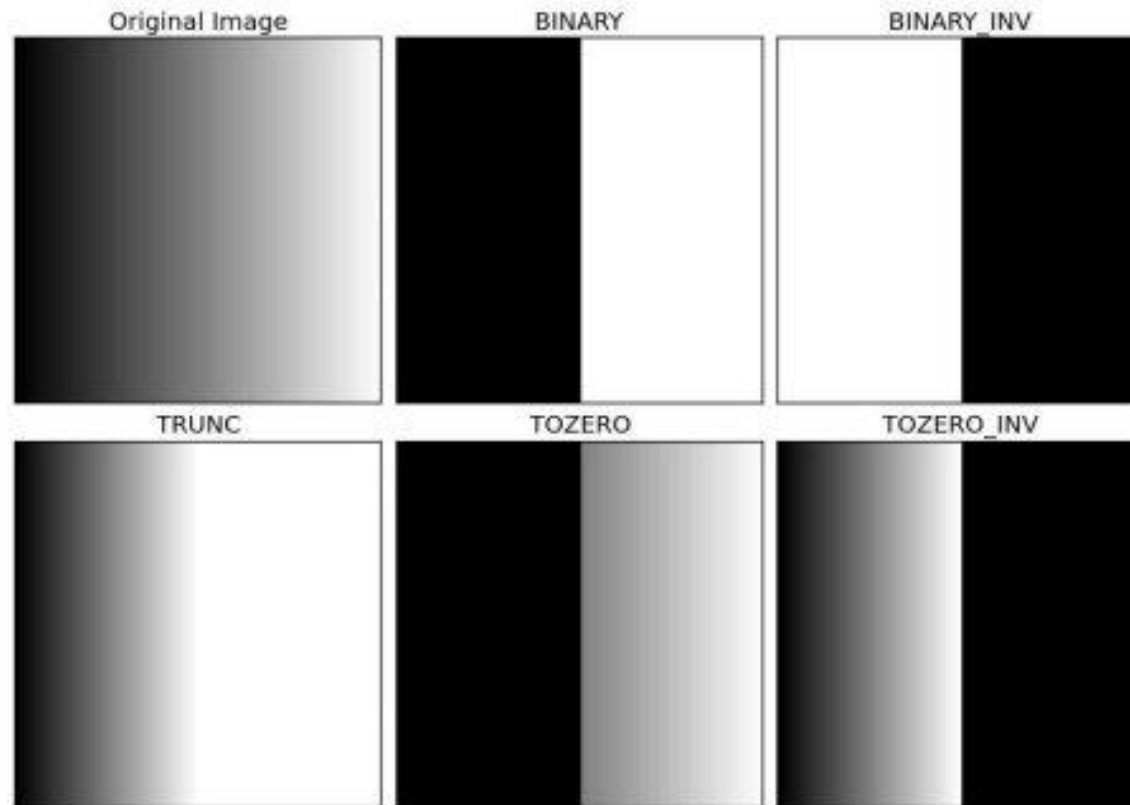
`cv2.threshold(img, thresh, maxVal, type)`

- adaptives Thresholding



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

Threshold Types



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
- Ethernet auf `hskaopen` umstellen
- Wired Settings:
 - `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://goo.gl/forms/1JRof8CgFAHSASKk2>



Vielen Dank

Robin Baumann

rbaumann@inovex.de

Stanislav Frolov

sfrolov@inovex.de

inovex GmbH

Ludwig-Erhard-Allee 6

76131 Karlsruhe

