

Gesichtsverfolgung mit dem Raspberry Pi

VERTEIDIGUNG DER SEMINARARBEIT
VON LEONARDO HÜBSCHER

1. Einstieg

Zielstellung

Demonstrierung der Funktionsweise der Gesichtserkennung mit Hilfe des Raspberry Pis

Konkreter: Verfolgung eines Gesichtes durch eine vom Raspberry Pi gesteuerte Webcam, welche in geeigneter Weise gesteuert werden kann

Motivation

Bereits seit längerem Interesse für die vielseitigen Einsatzmöglichkeiten des Raspberry Pis

Gliederung

1. Einstieg
2. Theoretische Grundlagen
 1. Die Biologische Gesichtserkennung des Menschen
 2. Viola-Jones-Algorithmus
 3. OpenCV
 4. Servomotor
3. Praktische Umsetzung
 1. Echtzeit-Gesichtserkennung
 2. Versuch des Trainierens eigener Haarcascades
 3. Lego-Konstruktion
4. Ansteuerung eines Servos
5. Verbindung der Module zu einem finalen Produkt
4. Fazit
5. Interessante Pi-Projekte Anderer
6. Diskussion:
Gesichtserkennung und Privatsphäre

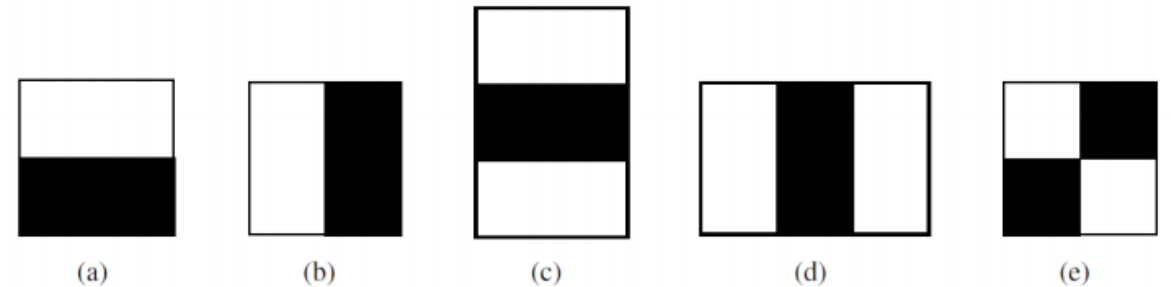
Theoretische Grundlagen

Die biologische Gesichtserkennung des Menschen

- Abstraktion altersunabhängige Merkmale
- Gesicht enthält Informationen zu:
 - Identität
 - Alter
 - Geschlecht
 - Gefühlslage
 - Aufmerksamkeit
- spielt bei sozialer Interaktion eine große Rolle
- Funktionsweise bis jetzt unerforscht

Viola-Jones-Algorithmus

- erster Objekterkennungsalgorithmus, welcher in Echtzeit Objekte erkennt
 - ursprüngliches Ziel: performanter Weg für die Gesichtserkennung
1. Graustufenbild
 2. Integralbild
 3. Analyse des Bildes durch Klassifikation der Helligkeitsdifferenzen
- Beschleunigung durch ADA-Boost-Verfahren
 - Cascade – Kaskade: Wasserfall
 - Nachteil: Helligkeitsabhängig



Basis-Klassifikatoren

OpenCV – Was ist das?

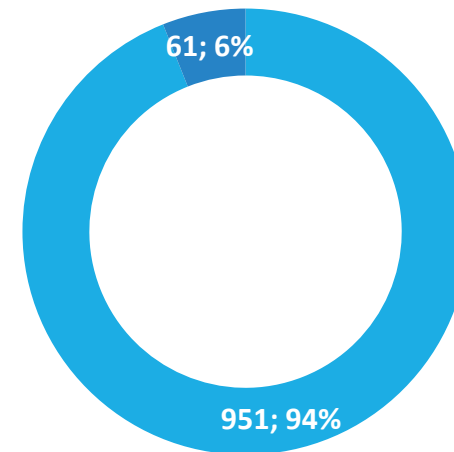
- ist eine Softwarebibliothek für die Verarbeitung von Bildern, sowie für maschinelle Lernprozesse
- besteht aus 2500 optimierten Algorithmen
- finanziert durch renommierte Unternehmen (Google, Yahoo, Microsoft, Intel)
- Einsatz:
 - StreetView
 - Sprengstofferkennung
 - Warnsystem für ertrinkende Personen



OpenCV - Test

- Programm zur Gesichtserkennung in einem Bild
- Anwender bestätigt Korrektheit bzw. korrigiert
(ohne Beeinflussung Cascade)
- abschließende Zusammenfassung der Ergebnisse
- es wurden von mir ca. 1000 Bilder untersucht
 - Motive: Gesichter (versch. Perspektiven), Auto, Haus

Testergebnisse



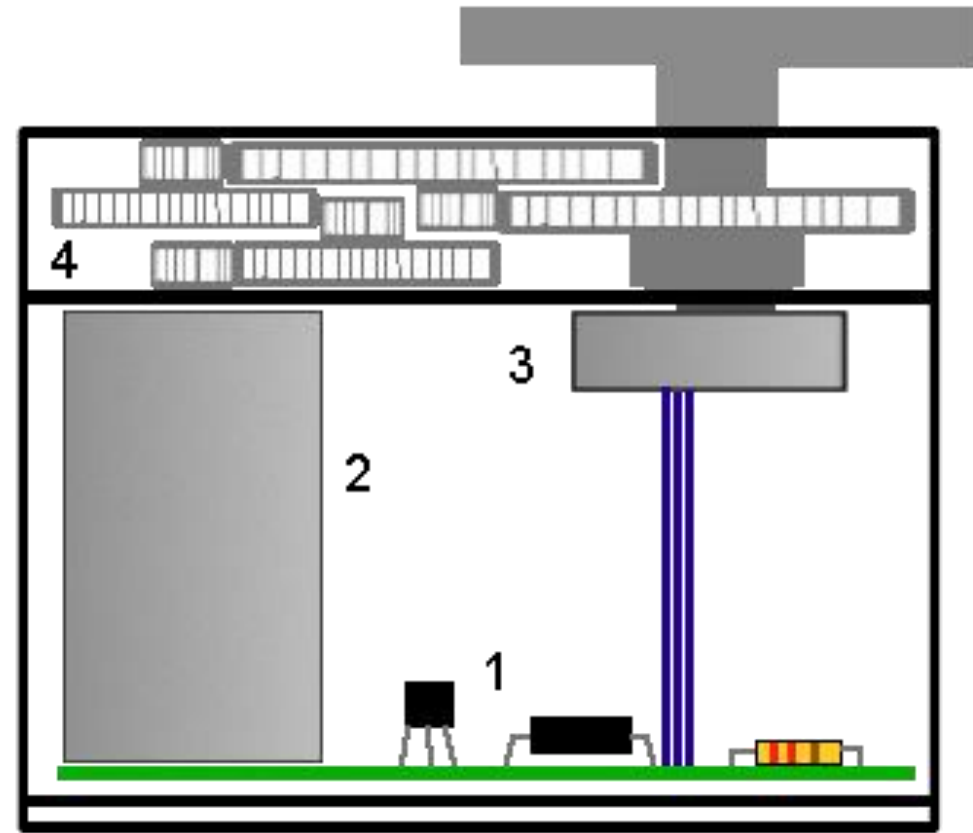
■ richtig erkannt ■ fehlerhaft

Programm ausführen

Servomotor

- ist ein positionsgeregelter Elektromotor
- Bedienung durch Steuerelektronik mittels PWM
- Drehung in die Richtung, welche geringere Entfernung hat
- anschließend Haltung der Position

1. Steuerelektronik
2. Elektromotor
3. Potentiometer
4. Getriebe



Praktische Umsetzungen

Echtzeit-Gesichtserkennung

■ Ziel: Erste Erfahrung mit OpenCV sammeln

1. Kamera-Stream einlesen
2. Gesichtserkennung anwenden
3. Rechteck auf Eingabebild um Gesicht zeichnen
4. Bild ausgeben

Programm ausführen

Trainieren eines eigenen Cascades

- Ziel: bei mehreren Gesichtern, das zu verfolgende markieren
- Beispiel: Daumen-hoch
- Trainingsprozess sehr fordernd
 - Zeit (Trainingprozess lief mehrere Tage bis Fehlschlag)
 - Rechenleistung
 - Konzentration (3 Stunden, für das markieren des Datensatzes benötigt)
- Vorbereitung:
 - Datensatz positiver (ca. 1000) und negativer Bilder (ca. 2500)
- Umsetzung:
 - Videoaufnahme in einzelne Bilder zerlegen
 - anschließend Daumen-hoch manuell markiert

Programm ausführen

Trainieren eines eigenen Cascades

- Programm legt Dateien aus Markierungen an:

Aufbau der infos_neg.txt

rel. Pfad zur Bilddatei

neg/003_frame.bmp

Aufbau der infos_pos.txt

rel. Pfad zur Bilddatei *Anzahl der Objektfunde* *Koordinaten der Funde (x, y, width, height)*

neg/003_frame.bmp 1 264 78 214 345

...

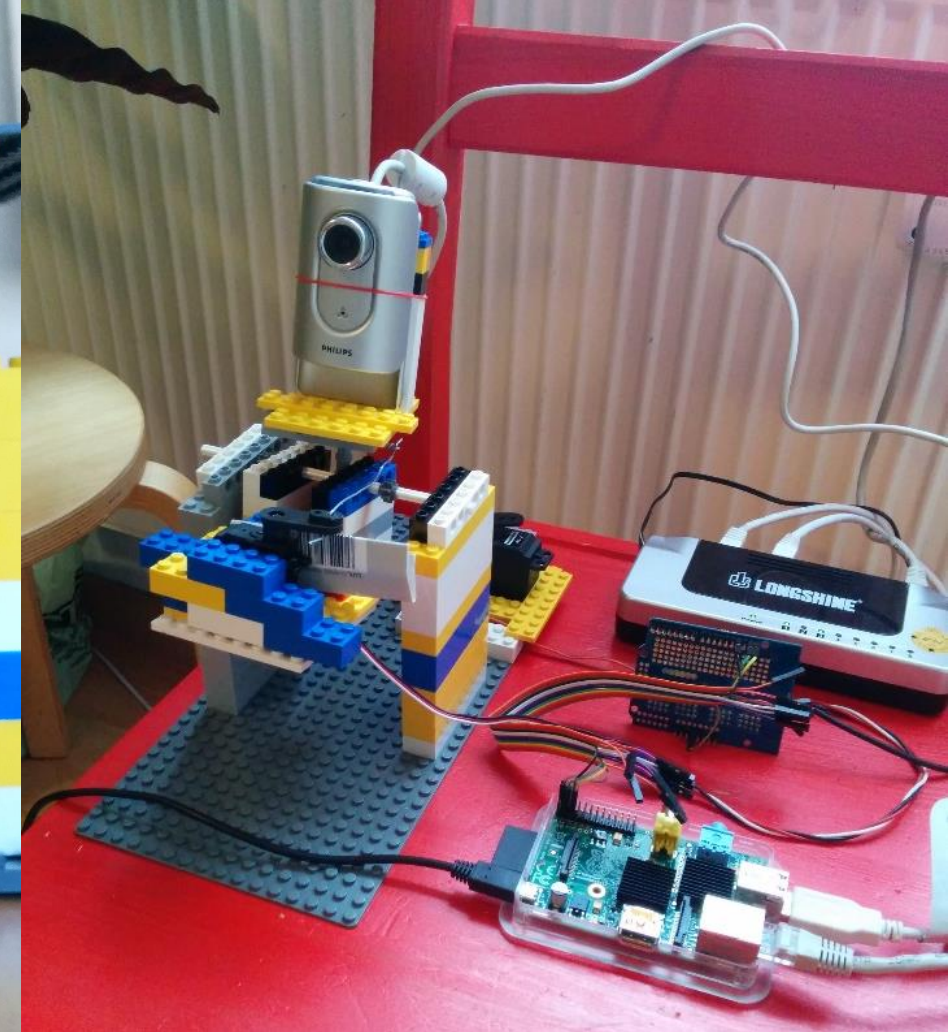
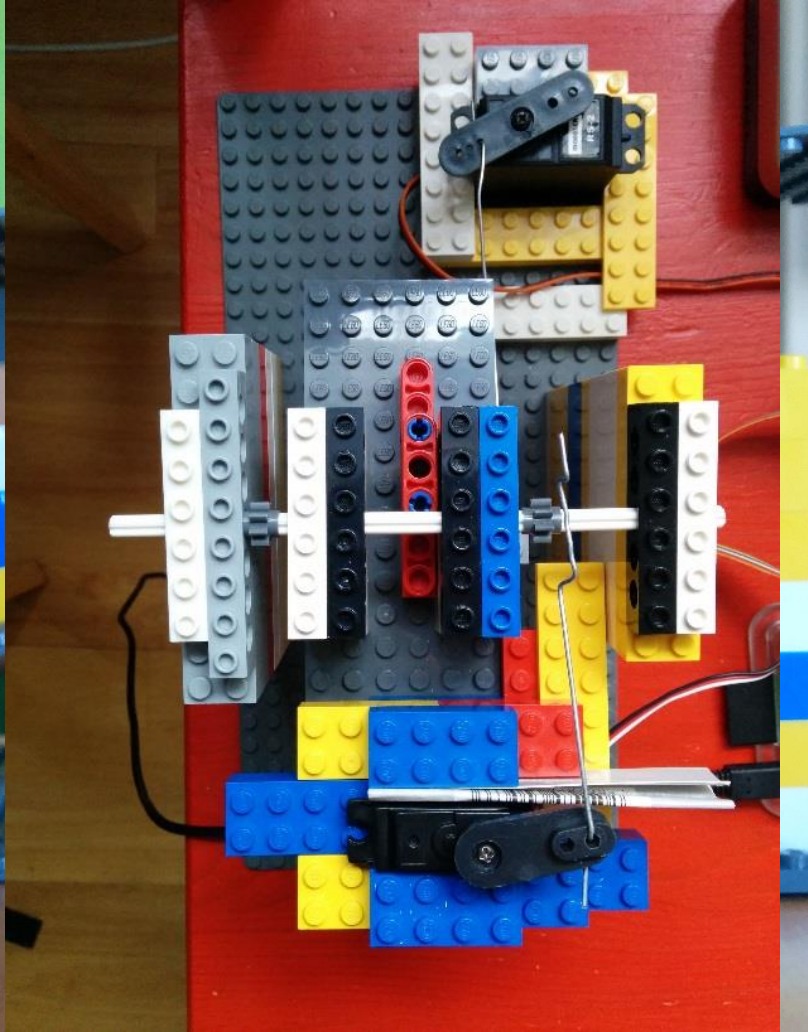
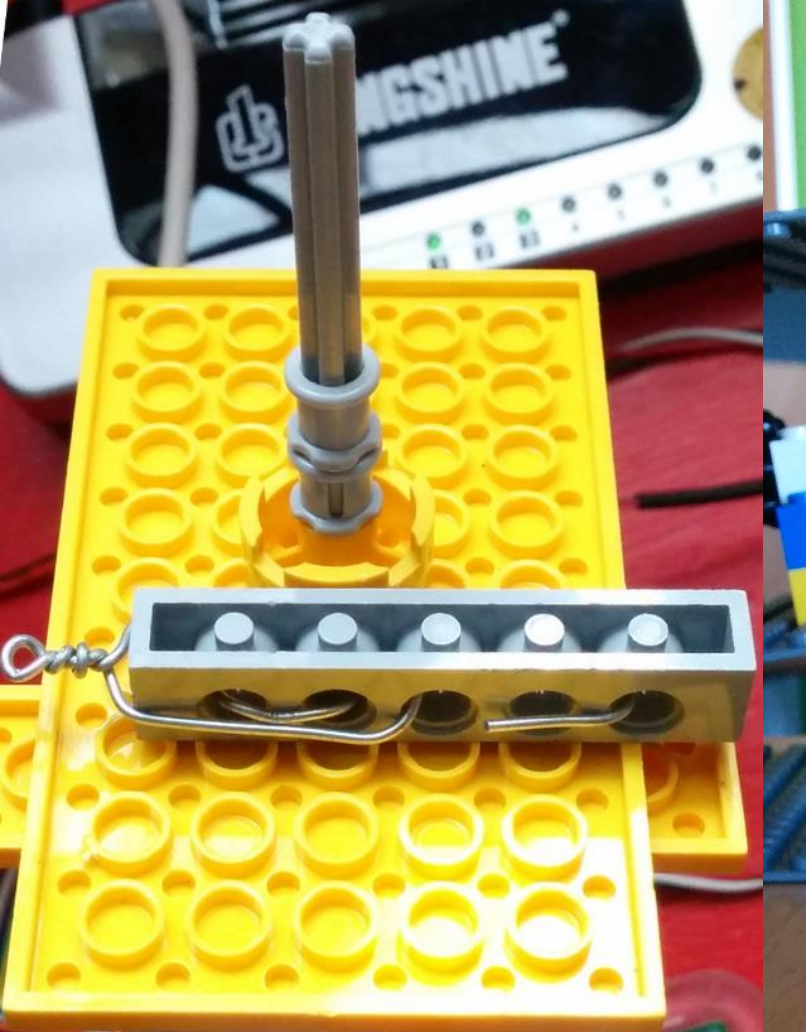
Trainieren eines eigenen Cascades

■ Probleme:

- Aufteilen des Video nicht mit OpenCV möglich (obwohl es gehen sollte)
→ Rückgriff auf JavaCV – Bibliothek
- mehrmaliges Abbrechen des Trainings aufgrund unbekannten Fehlers
- Dokumentation unvollständig/ veraltet

■ Ergebnisse:

- Frustration
- Umstieg auf Median aller Gesichter



Lego-Konstruktion für Endprodukt

- komplizierter als erwartet
- erfordert Grundkenntnisse im Umgang mit Lego und Servodrähten

Ansteuerung eines Servos

- auf Raspberry wird Programmiersprache Python verwendet
 - grundlegende Kenntnisse erforderlich
- Nach Einrichtung des Raspberrys relativ simpel
- Beispiel-Datei wurde umprogrammiert
- Servo-Demo fährt mehrere Positionen ab
- für Hauptprogramm wurde Interface programmiert
 - moveLeft, moveRight, moveUp, moveDown, moveToDefault
 - 2 Geschwindigkeitsstufen

Finales Produkt

- Verknüpfung der einzelnen Module
 - Echtzeit-Gesichtserkennung
 - Ansteuerung eines Servos
- Endprodukt verfolgt Gesicht in Frontalaufnahme
- Probleme:
 - Raspberry – Hardware zu schwach -> nur langsame Bewegungen nachvollziehbar
 - Performance auf Kosten der Qualität der Gesichtserkennung
- dennoch Erfolg, da Grundprinzip funktioniert

Fazit

„Die Arbeit mit dem Raspberry Pi war interessant und hat die Freude auf weitere Projekte solcher Art geweckt. Die Arbeit mit der OpenCV – Bibliothek hingegen hat aufgrund der mangelnden bzw. veralteten Dokumentation überhaupt keinen Spaß gemacht. Man musste ständig auf anderweitige Forenbeiträge im Internet zu den gleichen Fehlern und Problemen zurückgreifen, um zu einem Ergebnis zu kommen.“

Interessante Pi-Projekte Anderer

Sprachgesteuerte Hausautomation

Bild-Quellen

https://www.youtube.com/watch?v=gZkwvSX0_Os

<http://www.electronicsplanet.ch/Roboter/Servo/intern/querschn.png>

<https://www12.informatik.uni-erlangen.de/edu/mpa/ss11/talks/violajones.pdf>

Vielen Dank für Eure Aufmerksamkeit