

Machine Learning

Gaze Tracking Project

Alexander Rubinov, Stephan Sigel

24.02.2017

M.Sc. Informatik

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

Verfolgter Ansatz

Datengrundlage

Messungen und Vorhersagen

Fazit & Ausblick

Verfolgter Ansatz

Ziel

- Prototyp mit folgenden Eigenschaften und Funktionalitäten
 - Aufnahme von Bildern/Videos über eine Laptop-Webcam
 - Abbildung des geschätzten Blickpunkts des Benutzers
 - Clientseitig
 - GUI (Auf- und Entgegennahme von Daten)
 - Image Processing
 - Serverseitig
 - CNN (Training, Kalibrierung, Vorhersagen)

Betrachtung als Classification-Problem

⇒ Einteilung des Bildschirms in 9 Felder (3x3-Matrix)

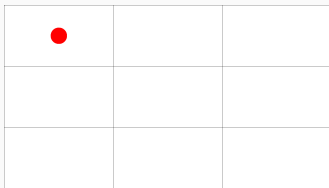


Abbildung 1: Kalibrierung

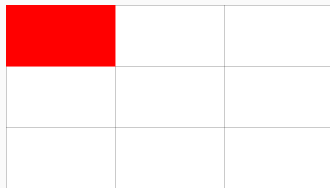


Abbildung 2: Gaze Estimation

Verwendete Datenquellen

- MPIIGaze-Datensatz (213.659 Bilder, 15 Personen)
 - Betrachtung der Daten und der Struktur
 - Erste CNN-Realisierung
- Erzeugung eines eigenen Datasets
 - Ca. 27.000 virtualisierte Augen-Bilder von zwei Personen (darunter ca. 1400 Originale)
 - Ausmaße der Augenbilder: 36x60
 - Zusätzlich serialisierte Ground Truth (0-8) mit Kopfwinkeln
 - Aufgenommen...
 - ...mit der selben Laptop Webcam
 - ...unter guten als auch schlechten Lichtverhältnissen
 - ...mit geringen Abweichungen bei der Kopfdrehung (head pose)

Image Processing

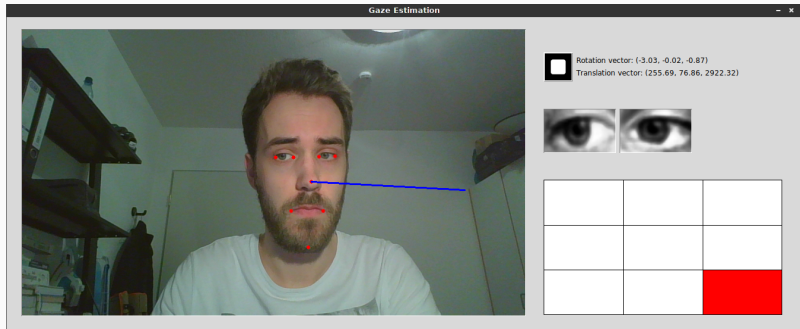


Abbildung 3: GUI mit allen Image Processing Schritten

- **Face & Landmark Detection** (Dlib)
 - Landmarks
- **PnP (Perspective-n-Point)** (OpenCV)
 - Rotation vector
- **Normalisierung** (OpenCV)
 - Rotation um Z-Achse
 - Ausschneiden der Augen nach fester Größe
 - Perspective Transformation
 - Konvertierung zu Graustufen
 - Histogram Equalization
 - ⇒ Höherer Kontrast & Minimierung ungleicher Lichtbedingungen

Classifier

- (Multimodal) Convolutional Neural Network (CNN)
 - 2 convolutional layer
 - 5x5 Filtermasken für die Gewichtungen
 - ReLU-Aktivierungen über alle Schichten
 - max pooling mit 2x2-Blöcken
- Ablauf
 - 1. Conv layer - 32 Neuronen - Ausgabe 18x30 px
 - 2. Conv layer - 64 Neuronen - Ausgabe 9x15 px
 - Densely connected layer - 1024 Neuronen
 - (Konkatenation mit der head pose)
 - Dropout
 - Readout layer
 - L2-Regularisierung der Gewichtungen
 - Softmax für Wahrscheinlichkeitseinteilung der Klassen

Datengrundlage

- **Anzahl der Trainingssamples**

- 4/5 des Datensatzes - ca. 22.000 Bilder
- Weitergabe in zufälligen generierten Batches

- **Anzahl der Testsamples**

- 1/5 des Datensatzes - ca. 6.000 Bilder
- 100 davon als Grundlage für Genauigkeit auf Testdaten

Messungen und Vorhersagen

CNN-Parameter

- Learning rate - 0.0001
- Std.abweichungen der Gewichtungen für C1, C2, FC1, FC2 - 0.1
- Biases für C1, C2, FC1, FC2 - 0.1
- Anzahl Neuronen für C1, C2, FC1 - 32, 64, 1024
- Stärke der Regularisierung - 0.01
- Wahrscheinlichkeit für Dropout der Neuronen - 0.5
- Anzahl Iterationen für Kalibrierung - 100

- **Messungen:** Trainingsverlauf ohne head pose, Trainingsverlauf mit head pose
- **Ohne head pose**
 - Genauigkeit auf Trainingsdaten: 0.99
 - Genauigkeit auf allen Testdaten: 0.97
 - Vorhersagen recht genau
- **Mit head pose**
 - Genauigkeit auf Trainingsdaten: 0.8
 - Genauigkeit auf allen Testdaten: 0.72
 - Vorhersagen ungenau, untere drei Felder z.B. kaum ansteuerbar

Fazit & Ausblick

Positives

- Image Processing
- Erzeugung der Kalibrierungsdaten

Negatives

- CNN: Schwer gute Konfiguration zu finden
- Viele mögliche und schwer aufzufindende Fehlerquellen
 - Problem bei uns: CNN hat aufgrund falscher Sortierung der Daten nicht lernen können

- Eliminierung des Klassenproblems und Erweiterung auf Winkel
- Automatisierung des Virtualisierungsprozesses der erzeugten Kalibrierungsdaten
- Spiegelung eines Auges und der Ground Truth für mehr Daten