Gaze Tracking

Dennis Müller & Sergej Bomke

Hochschule RheinMain

February 24, 2017

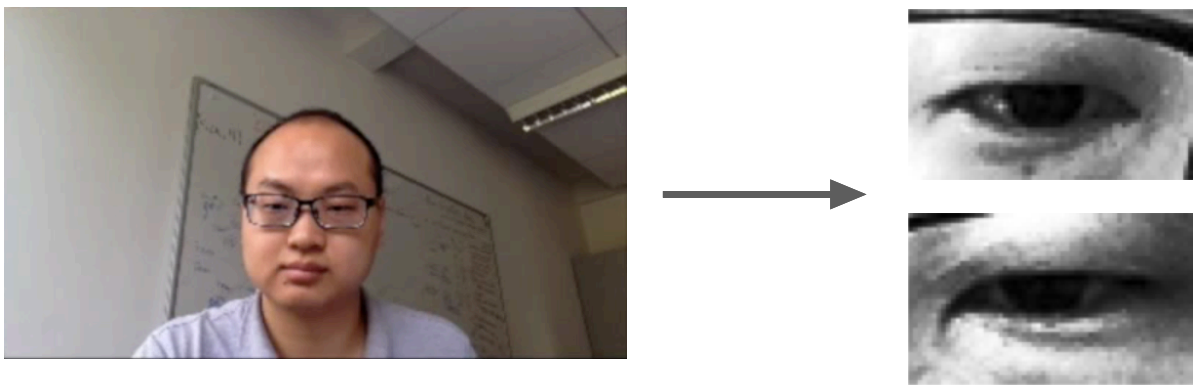
Gliederung

1. Datenquellen
2. CNN
3. SVR
4. Evaluation

Datenquellen

MPIIGaze

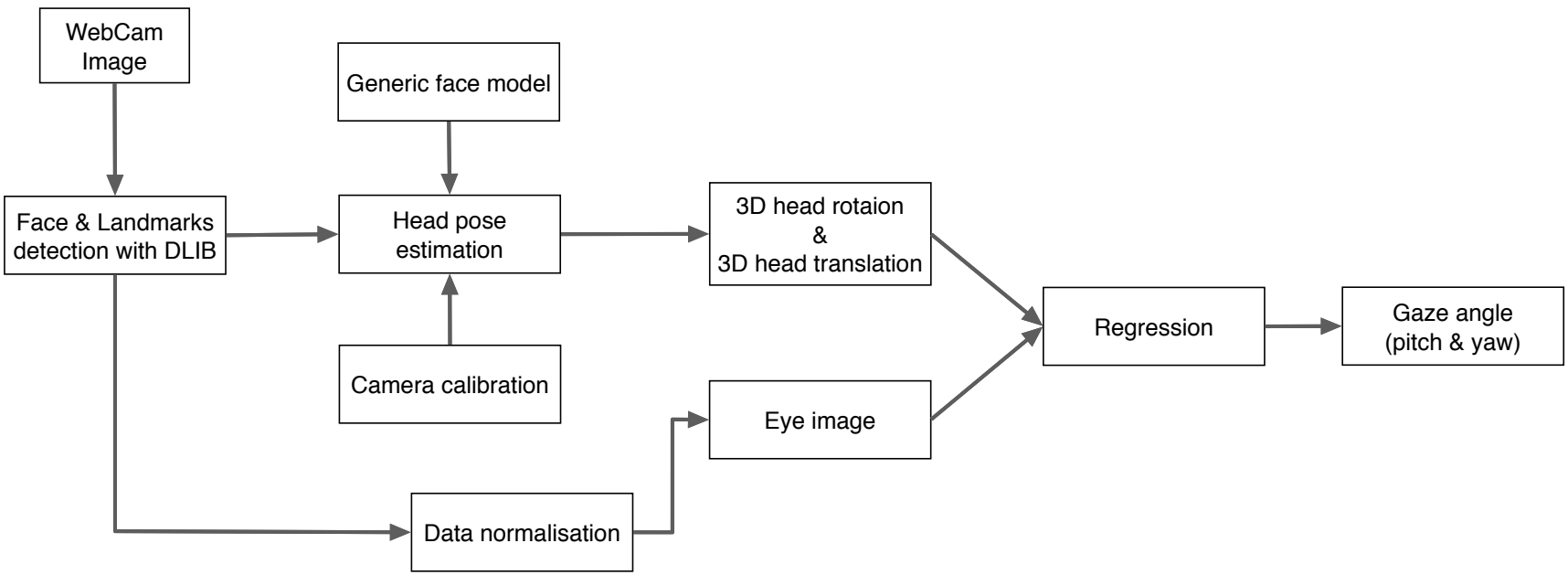
- 213.659 Bilder von 15 Personen
- Breites Spektrum an Aufnahmeorten, Zeit und Beleuchtung
- Bilder sind normalisierte für rechtes und linkes Auge
- 3D Kopfhaltung und 3D Blickrichtung



Mit WebCam aufgenommene Bilder

- MPIIGaze kompatible

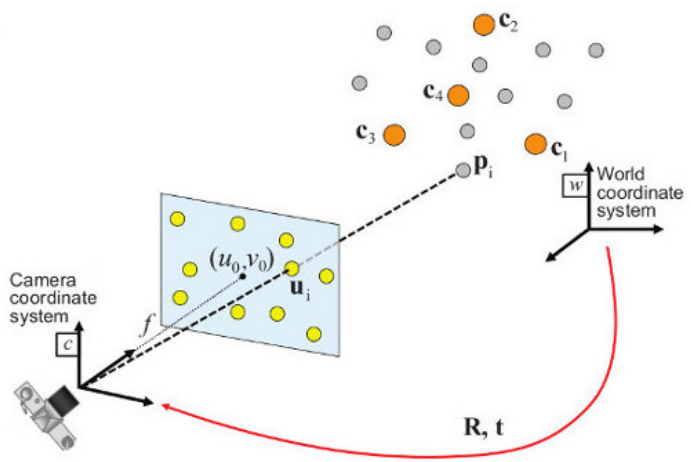
Vorverarbeitung von Bilder



Head Pose Estimation

Perspective-n-Point (PnP) Problem:

- $s\ m' = A[R|t]M'$
- 3D Punkte p_i in Weltkoordinatensystem
- 2D-Projektionen u_i im Bild
- Ziel: Pose (R, t) der Kamera zu ermitteln



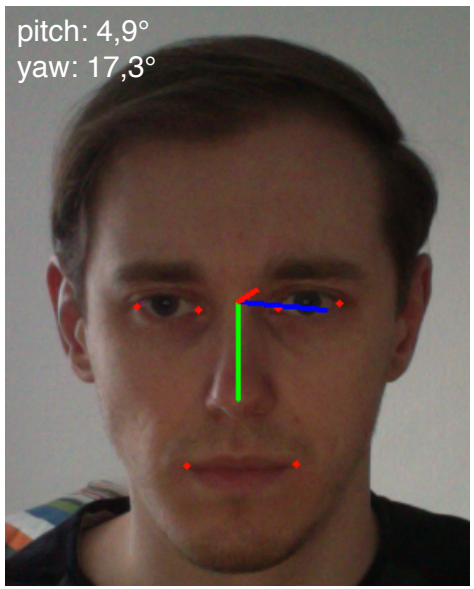
Quelle: OpenCV Tutorials, http://docs.opencv.org (abgerufen am 22.02.2017)

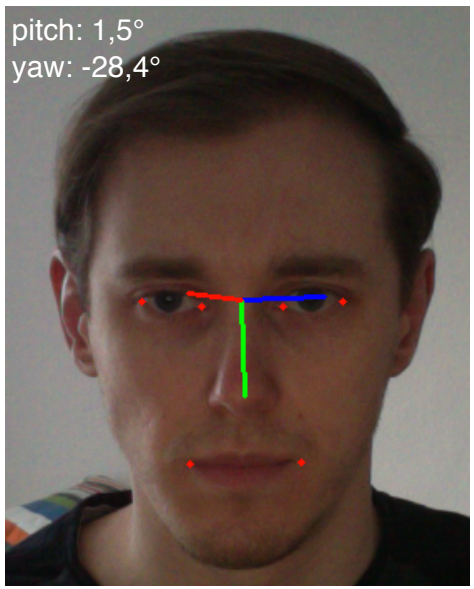
Head Pose Estimation

- OpenCV bietet verschiedene Ansätze zur Lösung des PnP-Problems
- 3D Punkte: generisches Gesichtsmodell aus MPIIGaze
  - Positionen von sechs Landmarks
  - Aufgenommen mit einer externen Stereo-Kamera
  - Mittelwert über alle Teilnehmer
- 2D-Koordinaten: Landmarks
  - Ermittelt mit Dlib Framework
- Parameter der Kamera
  - Kamerakalibrierung mit dem Schachbrett

Head Pose Estimation

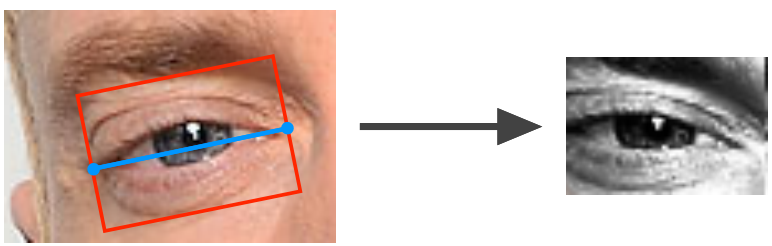
- Ermittelter Wert ist instabil
  - Filterung der Landmarks
  - Diverse Gesichtsmodelle
- Lösung
  - Feste Kopforientierung





Normalisierung

- Skalierung und Drehung
  - cv2.getPerspectiveTransform: Berechnet die Matrix einer 3x3 perspektivischen Transformation
  - cv2.warpPerspective: Wendet eine perspektivische Transformation auf ein Bild an
- Beschneiden auf 60x36 Bilder
- Histogram equalization



Architektur CNN

Entnommen aus Vorlage-Paper

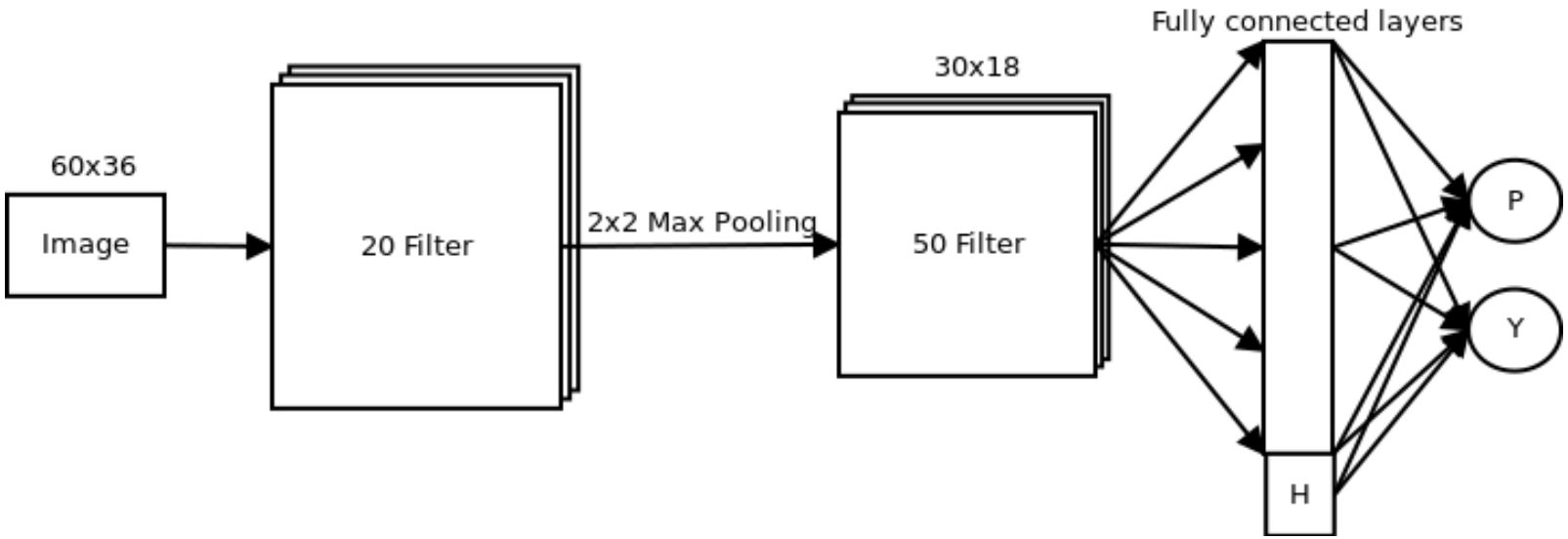


Figure 1: Kernarchitektur des CNNs

Lernen

Lernen auf rechtem Auge von MPIIGaze-Datensatz
Lernrate = 0.0001
Für Demo: 177847 Trainingsbilder, 280 Testbilder
Für Evaluation: 177458, 669 Testbilder
Loss-Funktion: L2-Loss

Hürdenübersicht

- Learning Rate ermitteln
- Inkonsistenzen in Vorlage: Loss- und Accuracy-Funktion
- Loss-Funktionen
- Probleme in MPIIGaze
- Sterbende Neuronen bei Versuch, das Problem runter zu skalieren
- Generalisierungsprobleme

Schwierigkeiten

In Vorlage wurde eine Learning Rate von 0.1 verwendet → Hat in Tensorflow nicht funktioniert

Vorlage benutzt Caffe mit modifizierten "'Euclidian Loss"' und "'Accuracy"'; beides der Winkel zwischen den rekonstruierten Vektoren (Prediction vs. Ground Truth)
→ Kann NaN sein, und Paper spricht von L2-Loss

Verschiedene Loss-Funktionen im Verlaufe des Projektes ausgetestet bei Lernproblemen, aber bei L2-Loss verblieben

MPIIGaze

Beinhaltet unterschiedliche Ground Truth-Daten für linkes und rechtes Auge → Datensatz könnte schwierig ausfallen

Schwere Outlier in Daten der 4.ten von 14 Personen:

Figure 2: "'Normalisiertes"' Bild

Reduktion auf Bilder

Versuch, Netzwerk nur mit Bildern zu füttern...

- Extraktion von Daten aus MPIIGaze mit Kopfwinkel nahe 0 $\rightarrow$ Nur noch $\tilde{1}50$ Bilder
- Training mit ganzen Daten $\rightarrow$ Funktioniert nicht, Neuronen in Convolution sterben

Generalisierung

Netzwerk fitted recht gut auf Trainingsdaten, aber nicht auf Testdaten.

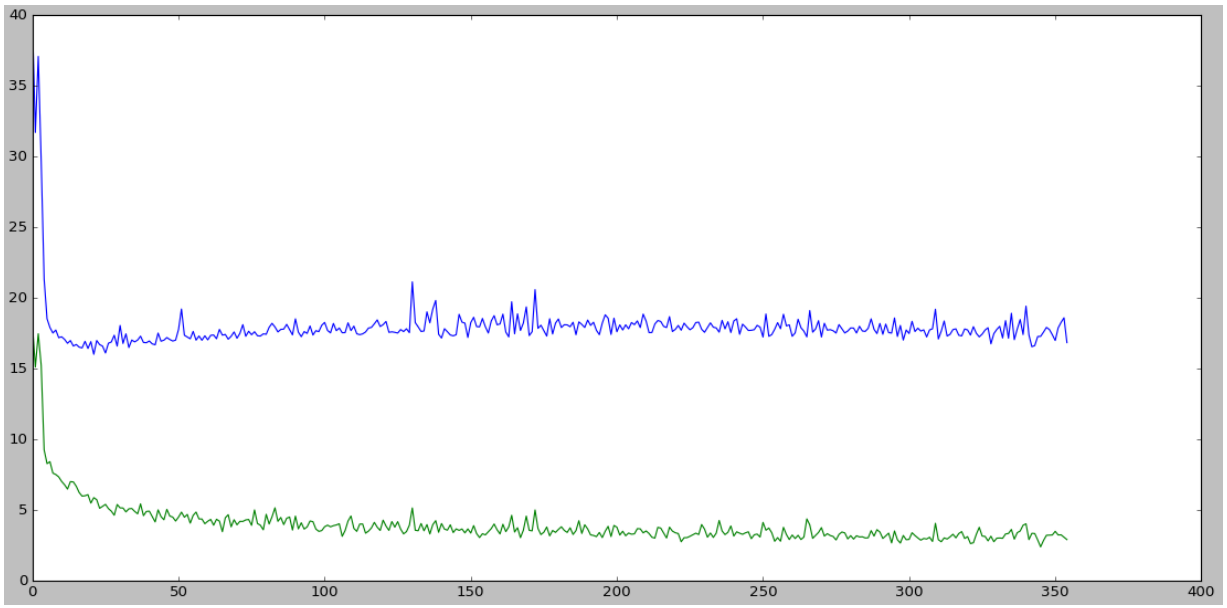


Figure 3: Trainingspase auf MPIIGaze: Batch vs. Testdaten, Winkelabweichung zwischen Vektoren

Test=16.83364486694336°, Training=5.811657905578613°

Generalisierung

Diverse Versuche, bei Generalisierung zu helfen:

- Dropout auf beiden Fully-Connected-Layers: Bremst nur Fitting auf Trainingsdaten aus
- L1-Regulation auf Convolution-Filter: Kein Effekt
- L2-Regulation auf Gewichte der FCLs: Moderate Angleicung von Test- und Trainingsdaten

Keine weiteren Versuche aufgrund von Zeitlimits, aber Anzahl der versteckten Neuronen ist auch noch ein möglicher Parameter.

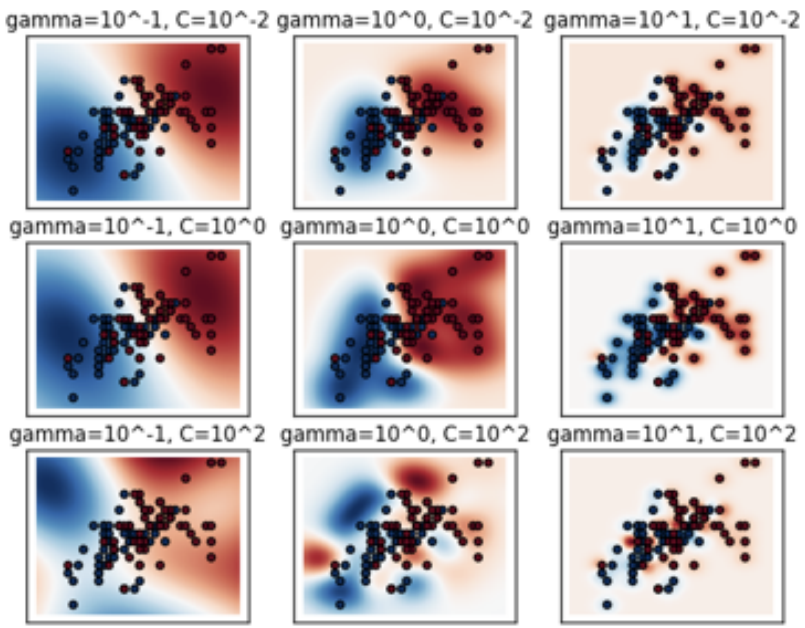
Support Vector Regression

- sklearn.svm.SVR
  - RBF-Kernel
  - Keine native Unterstützung von Multi-Target-Regression
- sklearn.multioutput.MultiOutputRegressor
  - Ein Regressor per Target
  - Trainieren kann parallelisiert werden
  - Nicht möglich verschiedene Parameter zu übergeben

Support Vector Regression

Freie Parameter:

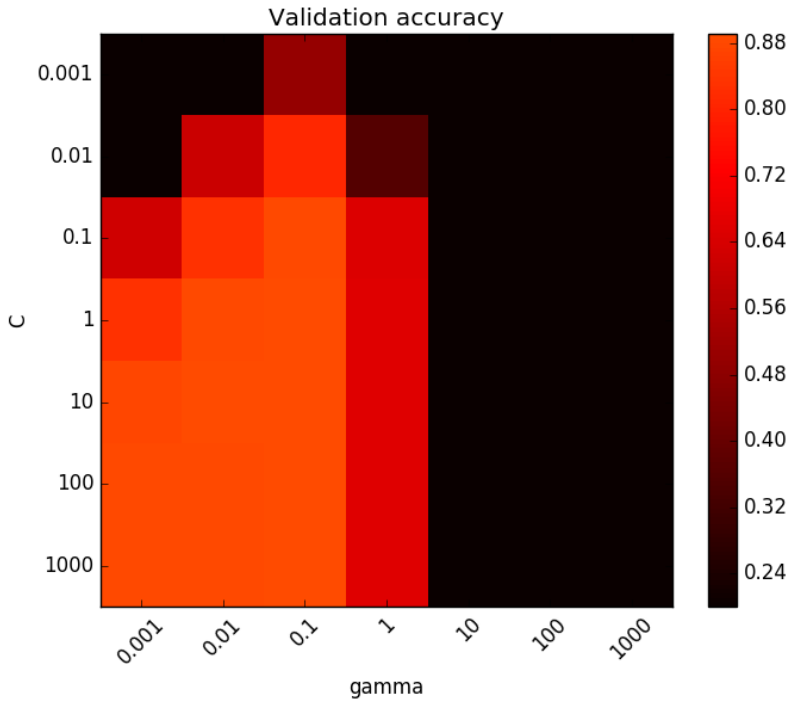
- C
- γ



Quelle: Scikit-Learn, http://scikit-learn.org (abgerufen am 22.02.2017)

Parameterbestimmung mit GridSearch

- 3-fache Kreuzvalidierung
- 49 candidates
  - 147 Fits
- R^2 Score



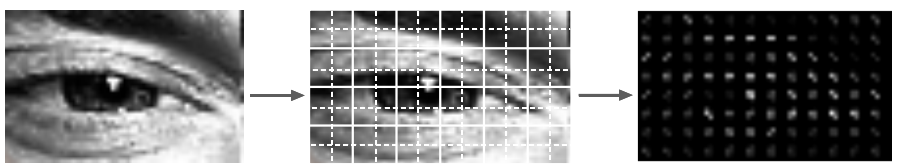
Features

Raw-Bilder:

- 2.160 Features

HOG:

- Signed gradients, 2x2 Cells per Block und 6x4 Blocks
- 2.772 Features



Multiple HOG

- 2x2 Cells per Block, 1x2, 3x1, 3x2, 6x4 Blocks
- 3.600 Features

Features

PCA:

- Reduktion auf 256 Features
- Hohe Speicherverbrauch beim fit()

\+ Head Pose (yaw und pitch)

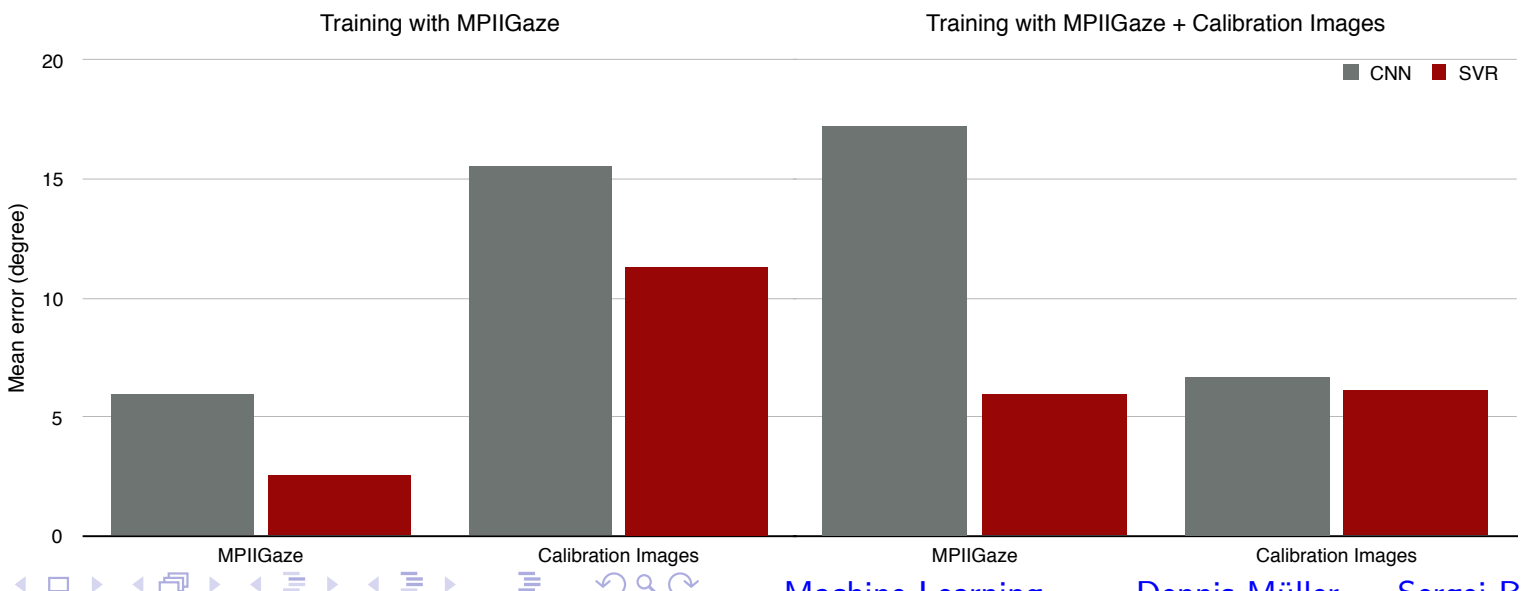
- Problem: Bild-Features dominieren
- Lösung: $HeadPose = HeadPose * \frac{AnzahlFeatures}{2} * \alpha$

Featuresbestimmung mit Crossvalidierung:

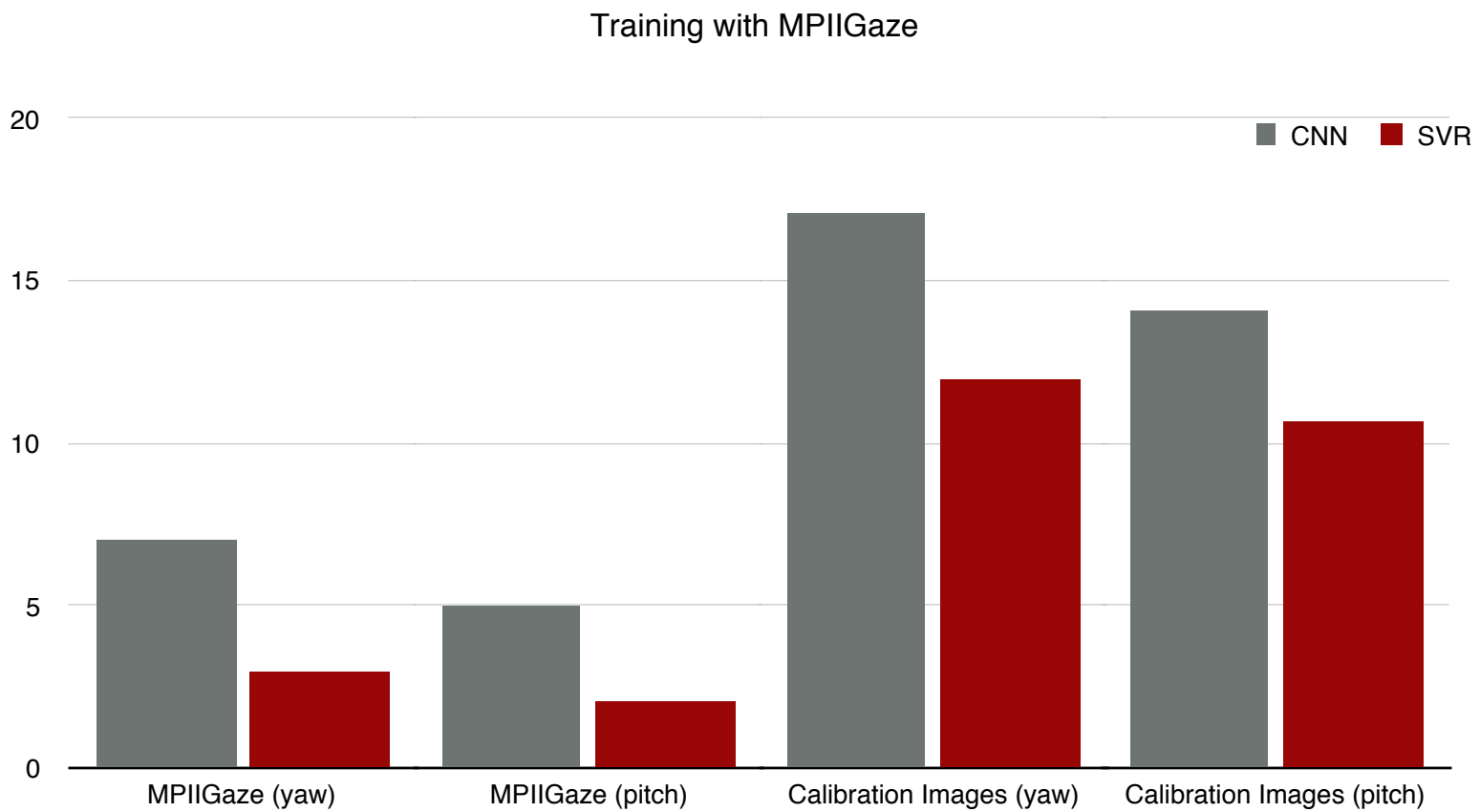
- HOG + PCA
- $\alpha = 0.25$

CNN vs SVR

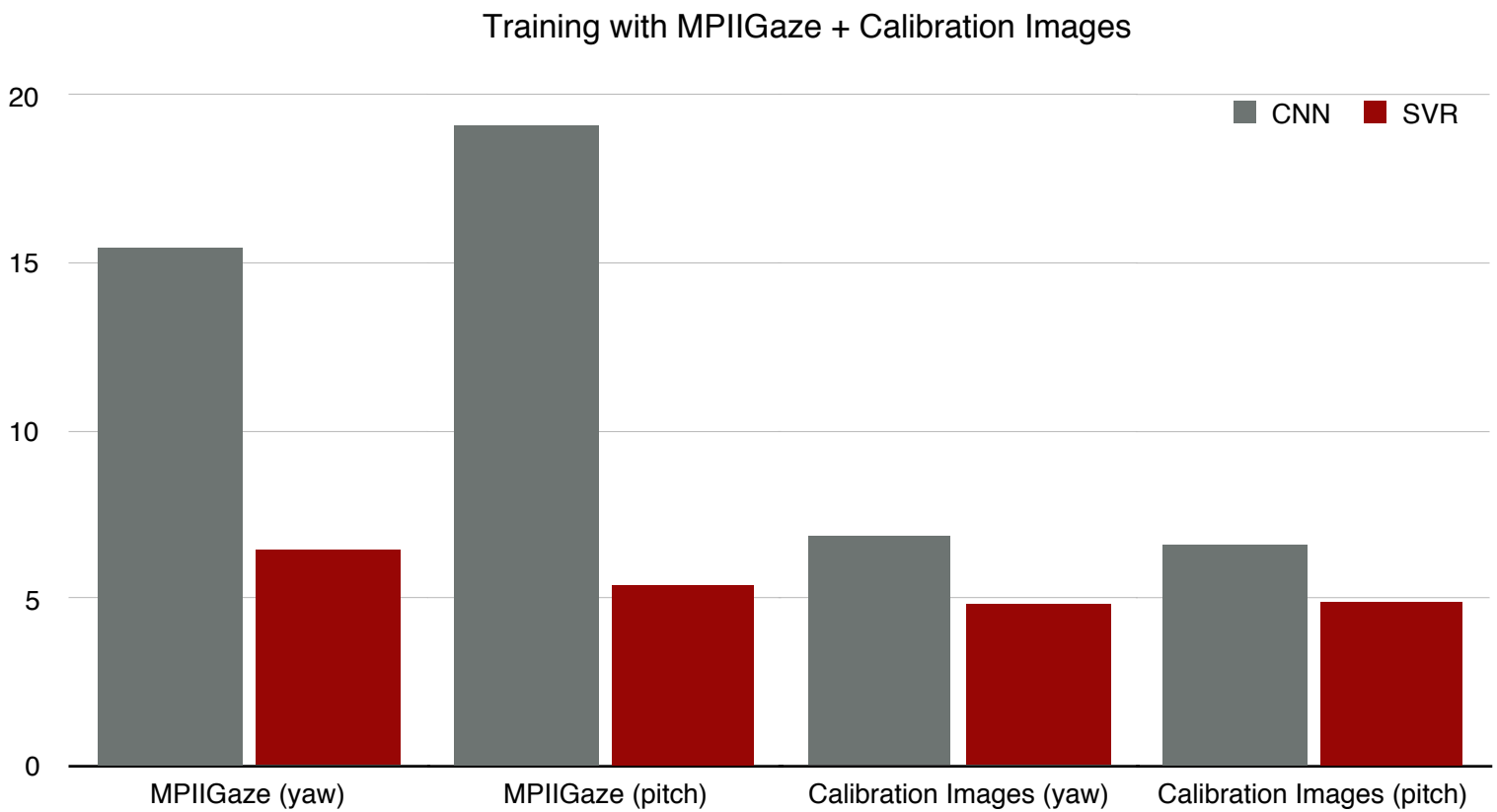
- Trainingsdaten:
  - 177.913 MPIIGaze
  - 5.880 Kalibrierungsbilder + 6.000 MPIIGaze
- Testdaten:
  - 670 MPIIGaze
  - 1176 Kalibrierungsbilder



CNN vs SVR


Training with MPIIGaze
CNN
SVR
20
15
10
5
0
MPIIGaze (yaw)
MPIIGaze (pitch)
Calibration Images (yaw)
Calibration Images (pitch)

CNN vs SVR


Training with MPIIGaze + Calibration Images
CNN
SVR
20
15
10
5
0
MPIIGaze (yaw)
MPIIGaze (pitch)
Calibration Images (yaw)
Calibration Images (pitch)