

Sensordatenverarbeitung

DIREKTE BILDMERKMALE (8C)

(2.-6.12.24)

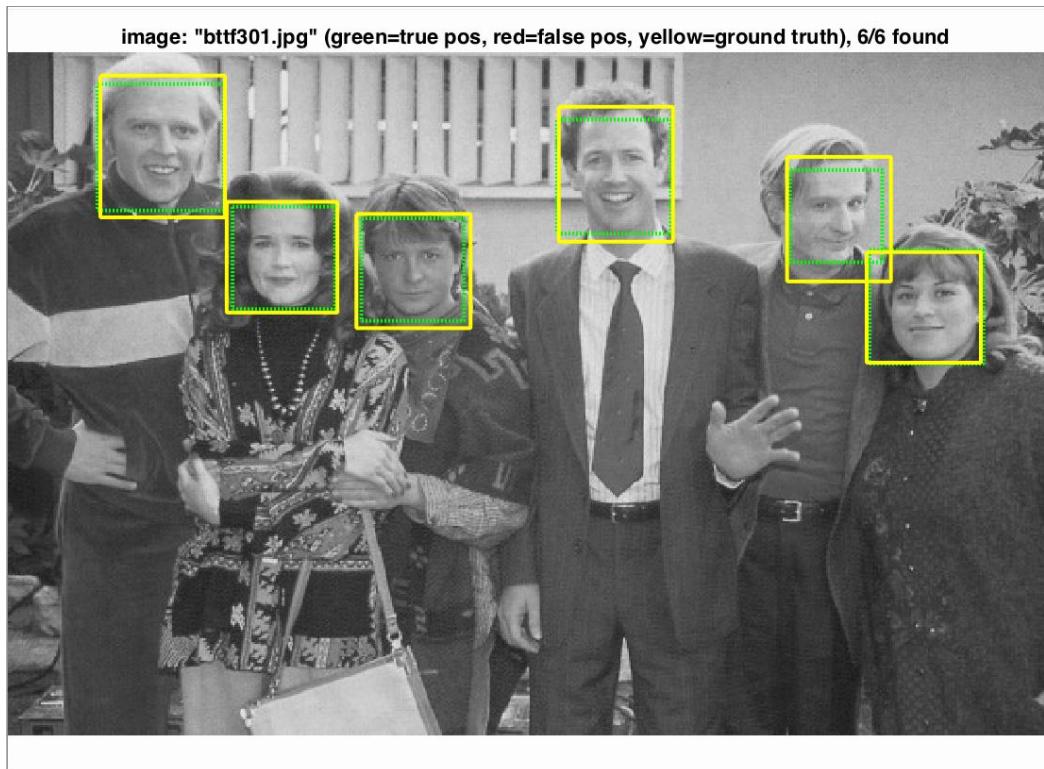
- Kontur ist ein mächtiges Merkmal für visuelle Erkennung
 - stark in menschlicher Wahrnehmung verankert
→ stilisierte Icons erkennen
- Wir wollen: Form der Konturen in einer Bildregion als Merkmal repräsentieren
 - ohne harte Entscheidung wie bei binärem Kantenbild oder Segmentierung
 - ohne geometrisches Modell der Form (z.B. Kreis, Linie; vgl. VL 10 Hough-Transformation)
 - geeignet für Klassifikation mit Machine Learning (vgl. VL 11)
- Histogram of Oriented Gradients (HoG)
 - Gradient = Helligkeitsübergang senkrecht zu Kante
- Navneet Dalal, Bill Triggs: Histograms of Oriented Gradients for Human Detection, CVPR 2005



- Erkenne Menschen in grob aufrechter Geh-/Stehposition
- Für z.B. Mensch-Technik-Interaktion, Überwachungskameras, Sport
- Beispiele unten zeigen eine viel größere Varianz an Situationen als bei kontrollierten industriellen Anwendungen



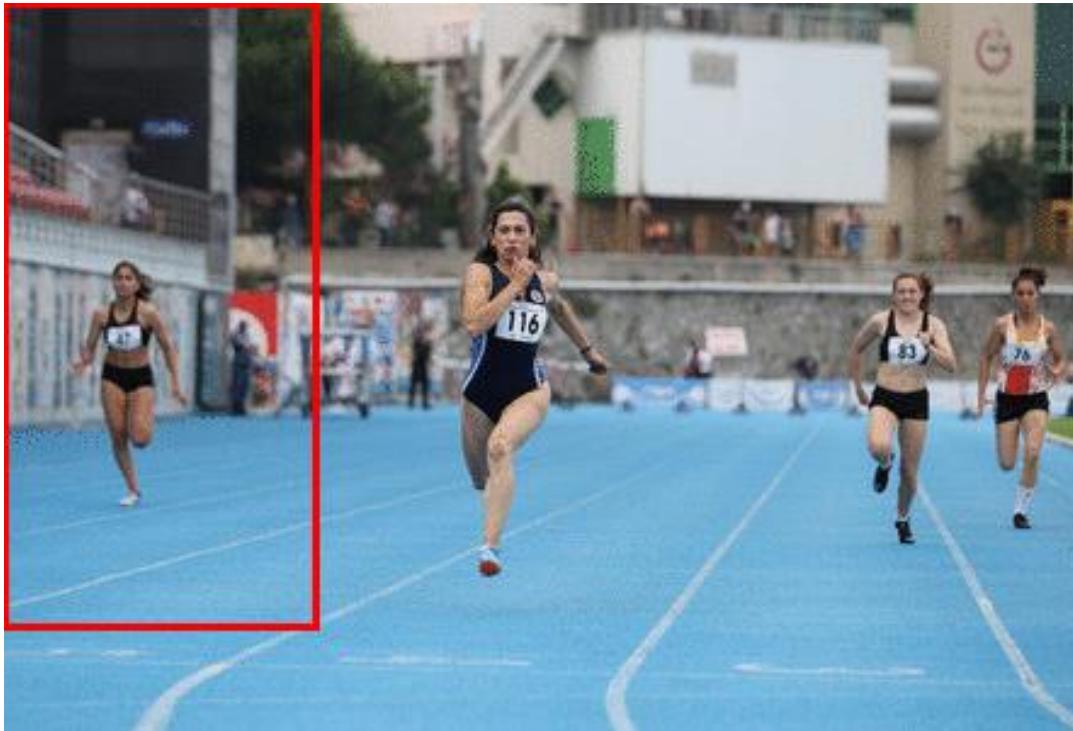
- Erkenne der Kamera zugewandte Gesichter
- Für z.B. intelligenten Autofokus, Mensch-Technik-Interaktion
- Quelle: <http://sklin93.github.io/hog.html>



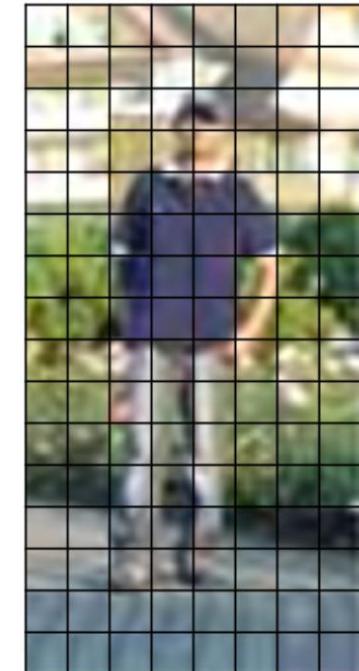
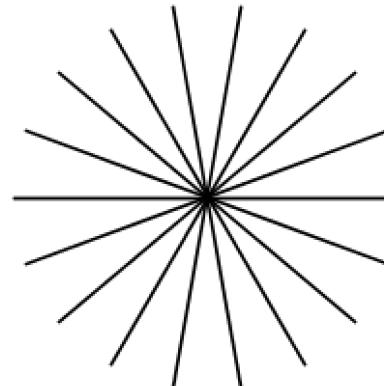
- Erkenne Verkehrszeichen für z.B. autonomes Fahren
- Kontur und zusätzlich Farbe
- Quelle: Creusen, Ivo M., et al. "Color exploitation in hog-based traffic sign detection." *2010 IEEE International Conference on Image Processing*. IEEE, 2010.



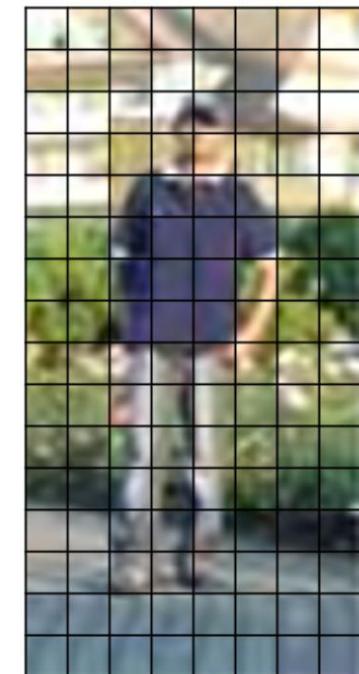
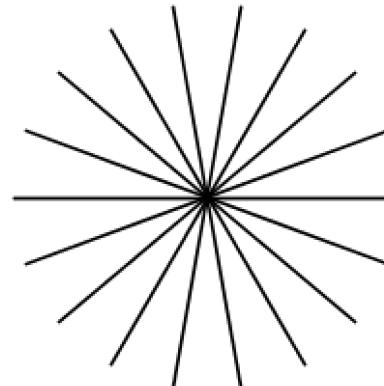
- Laufe durch Bild mit Rechteck variabler Größe aber festem Größenverhältnisses (z.B. 1:2) als Ausschnitt
- Skaliere Ausschnitt auf Einheitsgröße (z.B. 64*128)
- Berechne Merkmalsvektor auf Ausschnitt (hier HoG)
- Klassifizierte den Merkmalsvektor (vgl. VL Klassifizierungsalgorithmen)

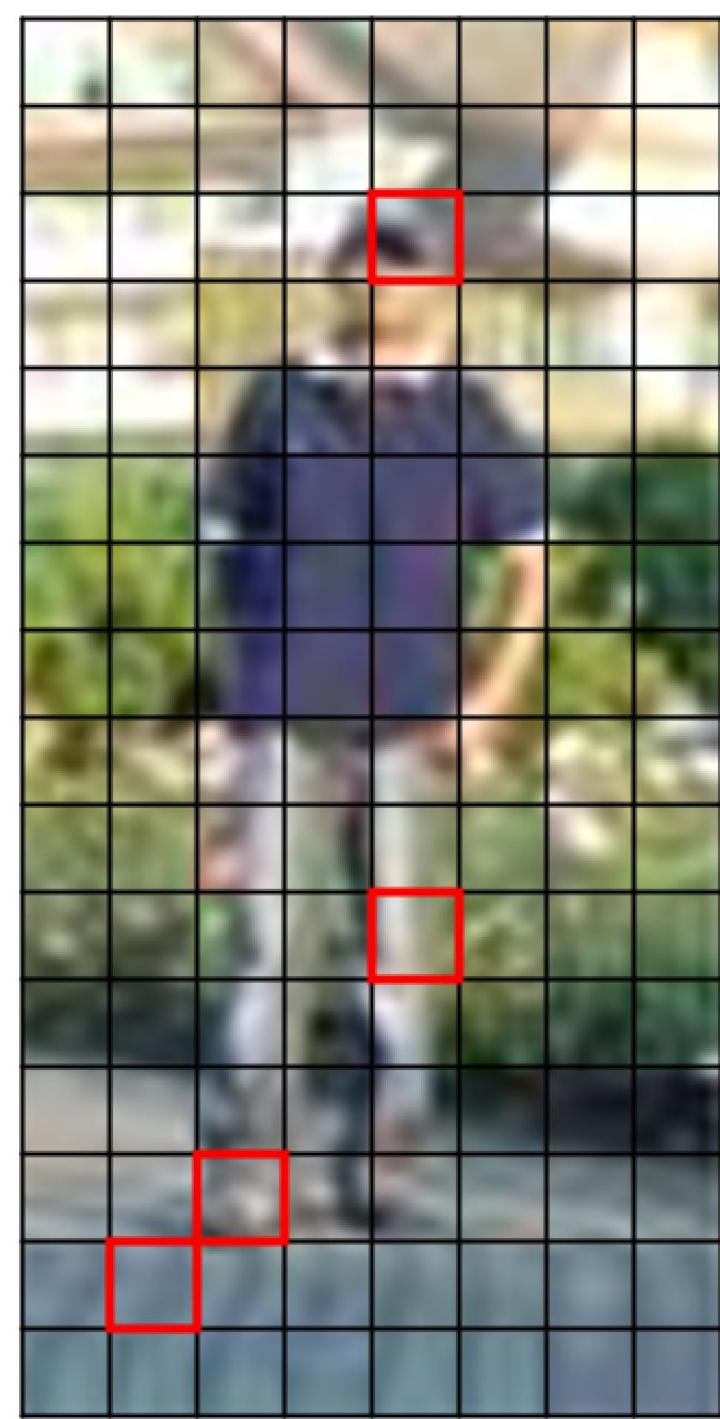


- Konturinformation eines Bildausschnitts
 - (Möglichst) unabhängig von der Beleuchtung
 - (Möglichst) unabhängig von Farben der Objekte
 - Ohne frühe harte Entscheidungen
- Idee des Histogram of Oriented Gradients
 - Wo im Bild verlaufen Kanten in *welcher Richtung?*



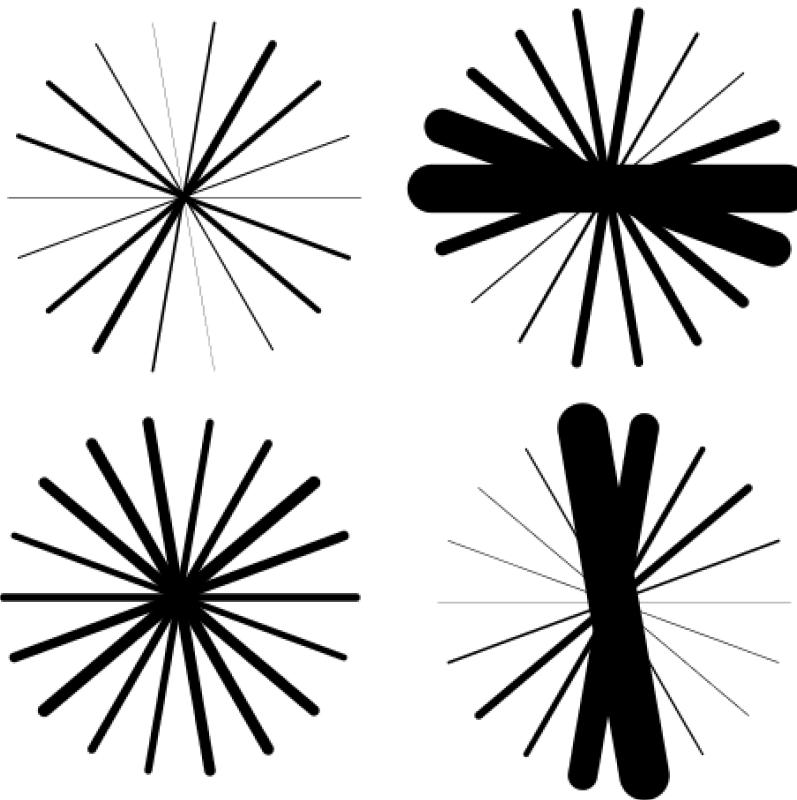
- Konturinformation eines Bildausschnitts
 - (Möglichst) unabhängig von der Beleuchtung
 - (Möglichst) unabhängig von Farben der Objekte
 - Ohne frühe harte Entscheidungen
- Idee des Histogram of Oriented Gradients
 - Wo im Bild verlaufen Kanten in *welcher Richtung*?
 - „Wo“ nur grob → 8*8 Pixel Raster
 - „In welcher Richtung“ nur grob → 20° Raster
 - Ein HoG-Eintrag sagt: „So viel Kante findet sich in diesem Rasterkästchen in dieser Rasterrichtung“
 - Dabei „So viel Kante“ noch im Detail zu klären





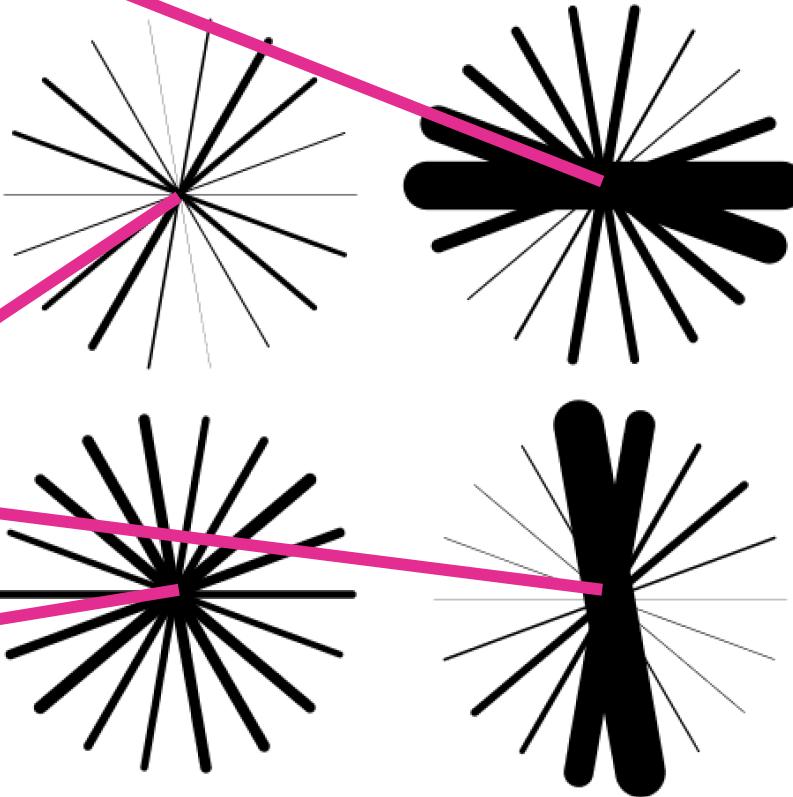
Intuition für HoG

Frage an das Auditorium: Welches HoG gehört zu welchem der rot markierten Rasterkästchen?

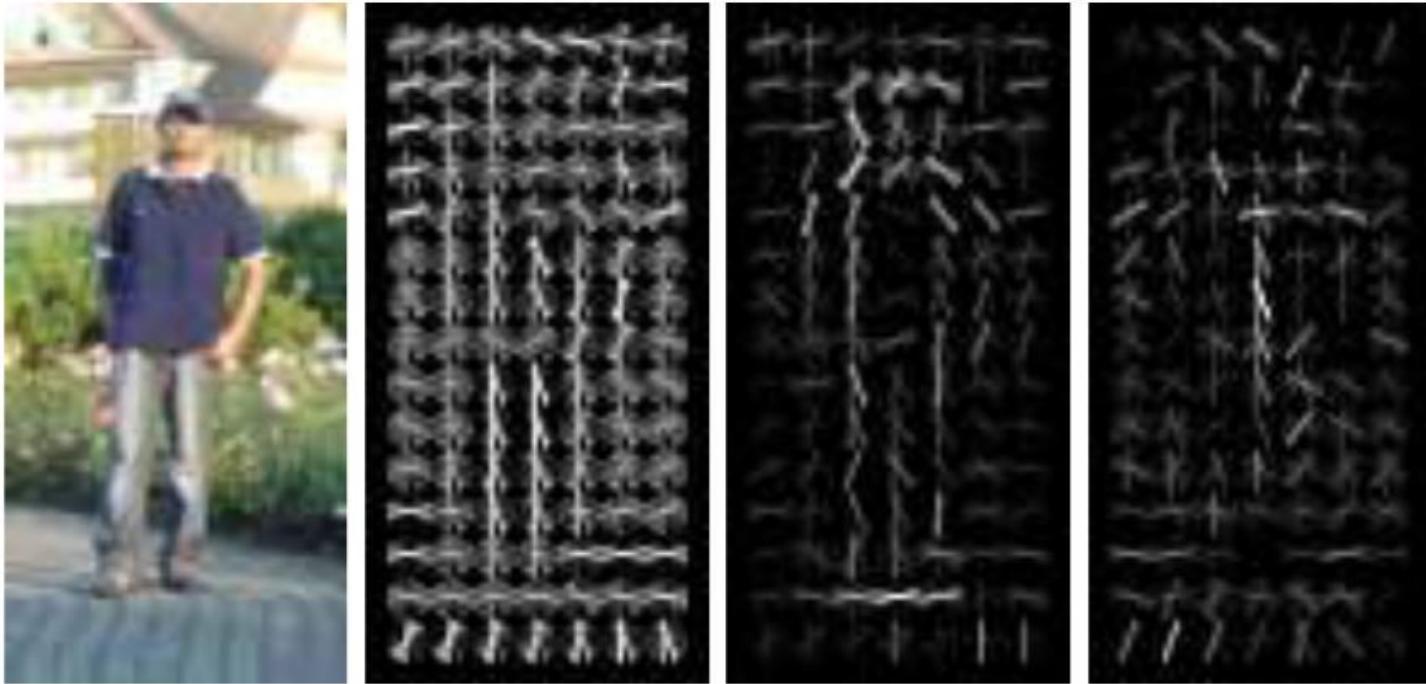


Intuition für HoG

Frage an das Auditorium: Welches HoG gehört zu welchem der rot markierten Rasterkästchen?

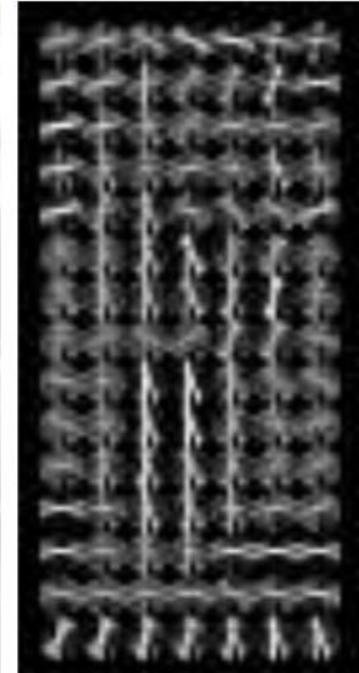


- Von links nach rechts
- Eingabebild
- HoG-Deskriptor
- HoG-Deskriptor-Einträge die später positiv zur Klassifikation beitragen
- HoG-Deskriptor-Einträge die später negativ zur Klassifikation beitragen



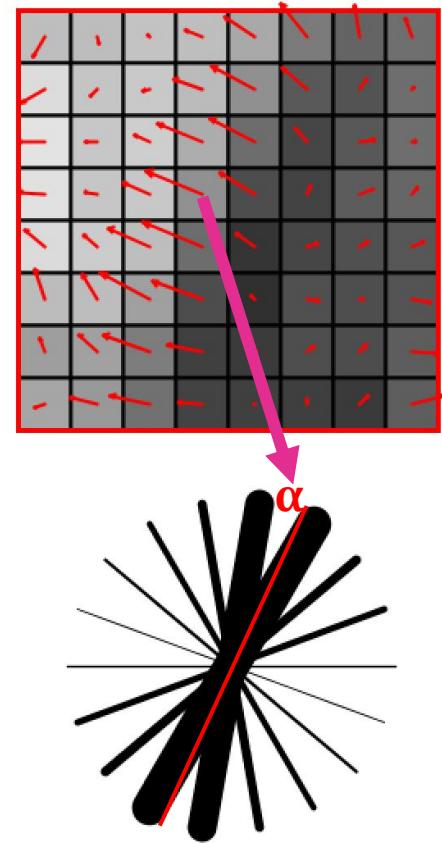
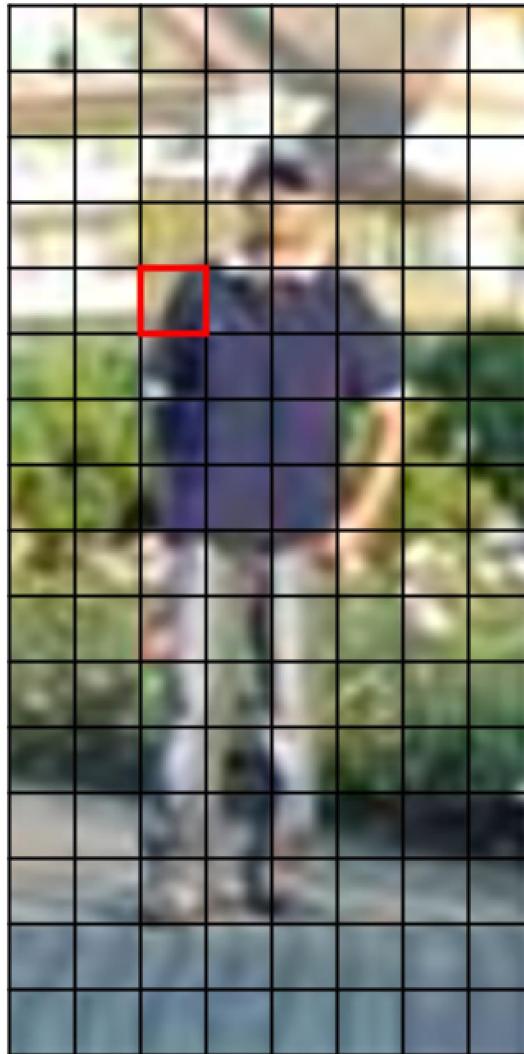
- Überblick HoG Merkmalsberechnung

- Berechne Gradientenstärke und –richtung (z.B. Sobel)
- Berechne in 8*8 Blöcken Histogramme der Gradientenrichtung (gewichtet mit Gradientenstärke) in 9 20°-Schritten
- Normalisiere die Histogramme in überlappenden 16*16 Blöcken (Beleuchtungsinvarianz)
- Kombiniere alle Histogramme zu einem Merkmalsvektor

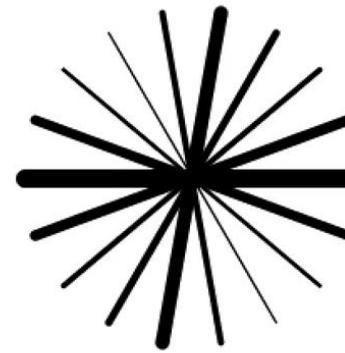
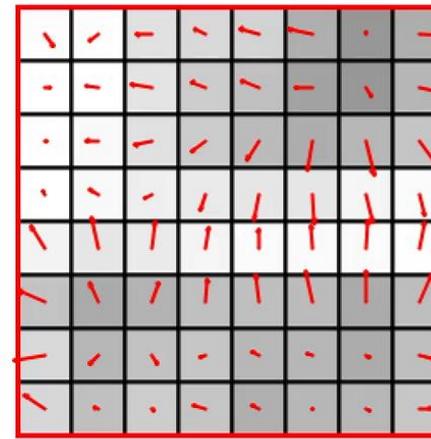
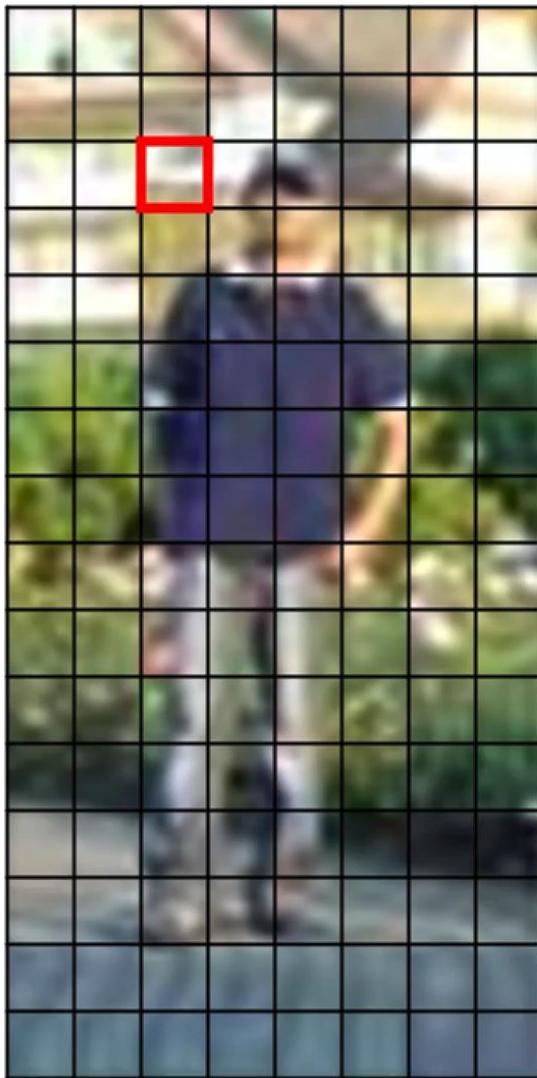


Histogram of Oriented Gradients

- Demo Nextclpud programme / hog, in Processing
- „So viel Kante findet sich in diesem Rasterkästchen in dieser Rasterrichtung“
- Richtungen α nur 0° bis 180°
 - Z.B. 40° und 220° gleich
 - Man weiß nicht, welche Seite einer Kante heller sein muss
- Was bedeutet „So viel“?

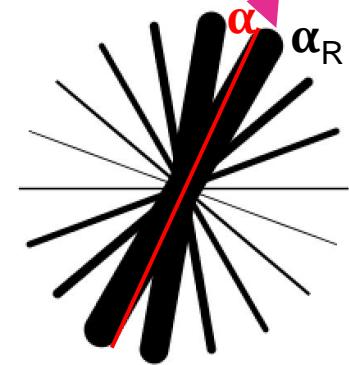
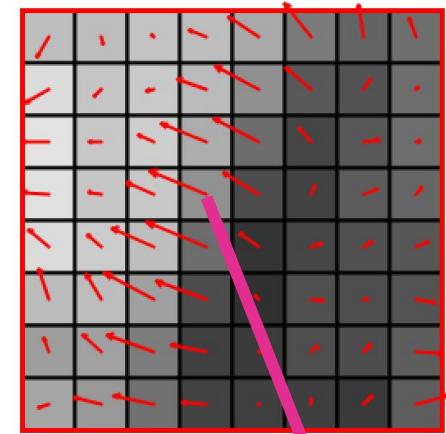
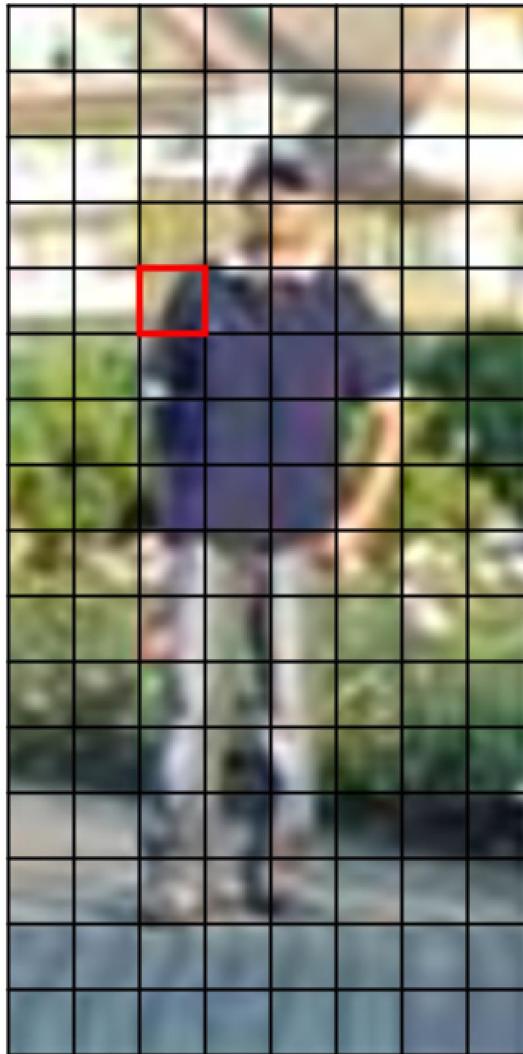


Histogram of Oriented Gradients



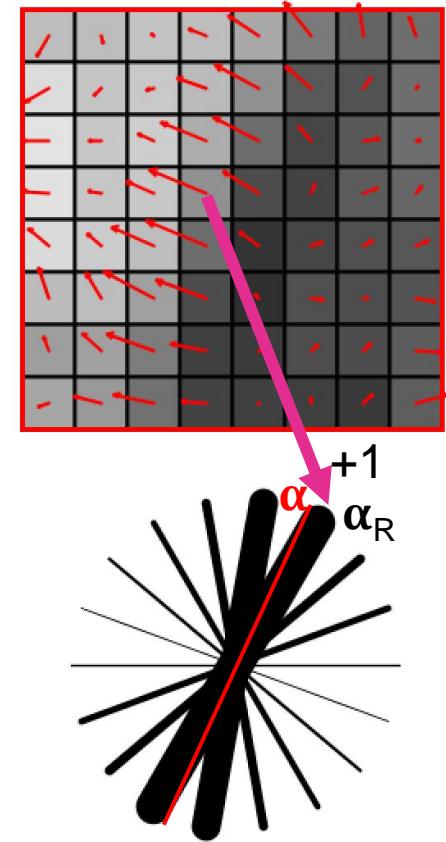
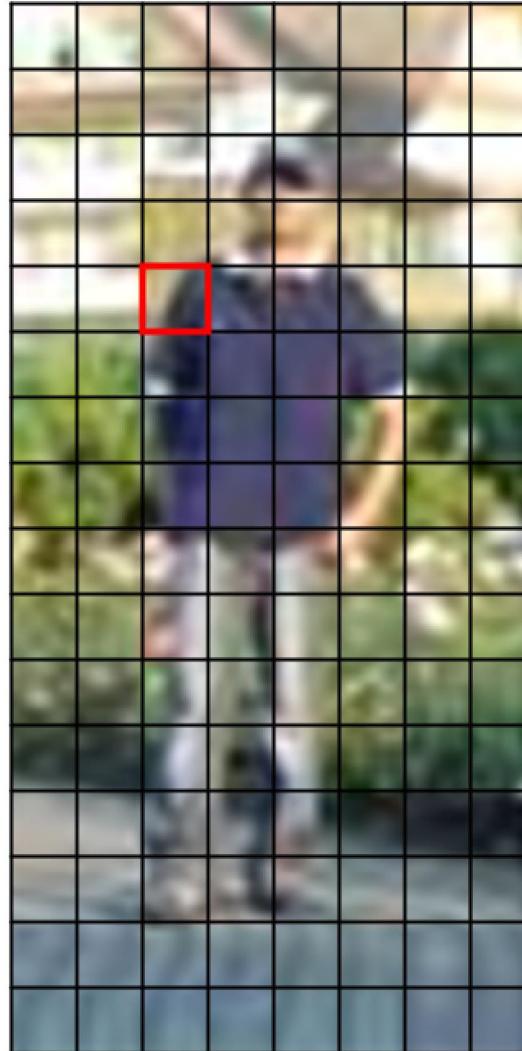
Was bedeutet „So viel Kante“?

- „So viel Kante findet sich in diesem Rasterkästchen in dieser Rasterrichtung“
- Was bedeutet „So viel“?

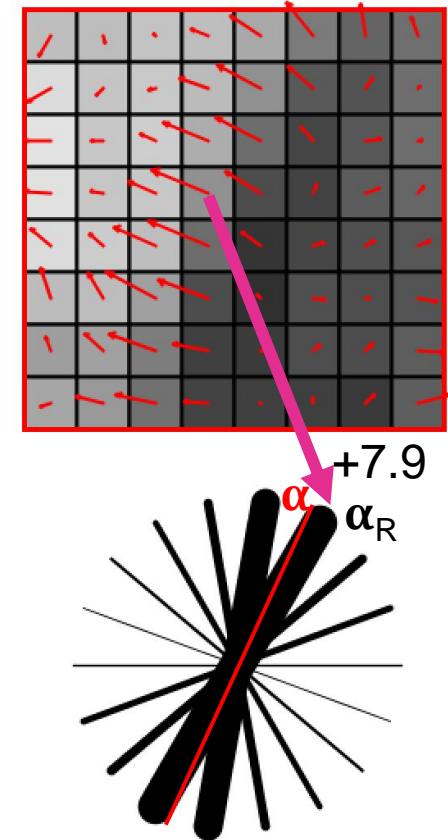
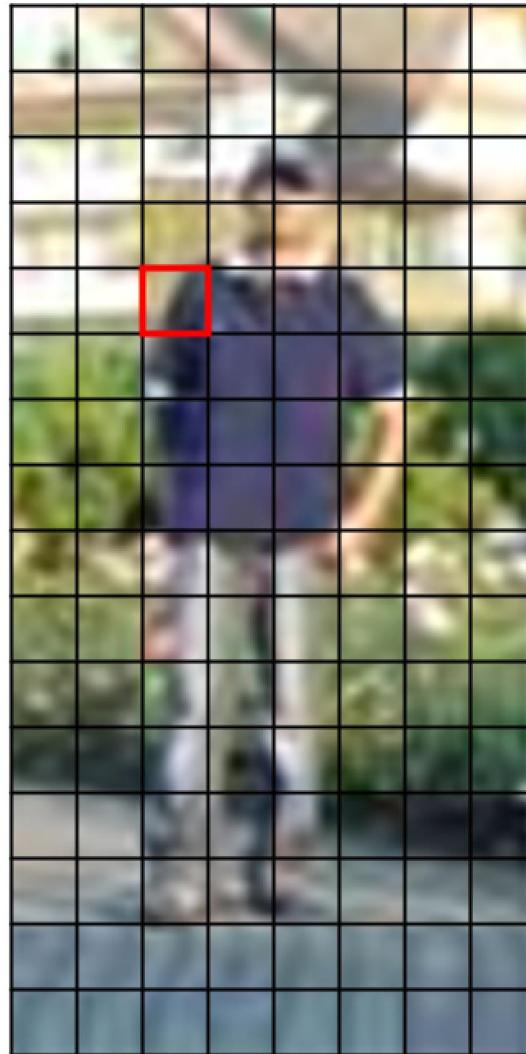


„So viel Kante“ – Idee 1

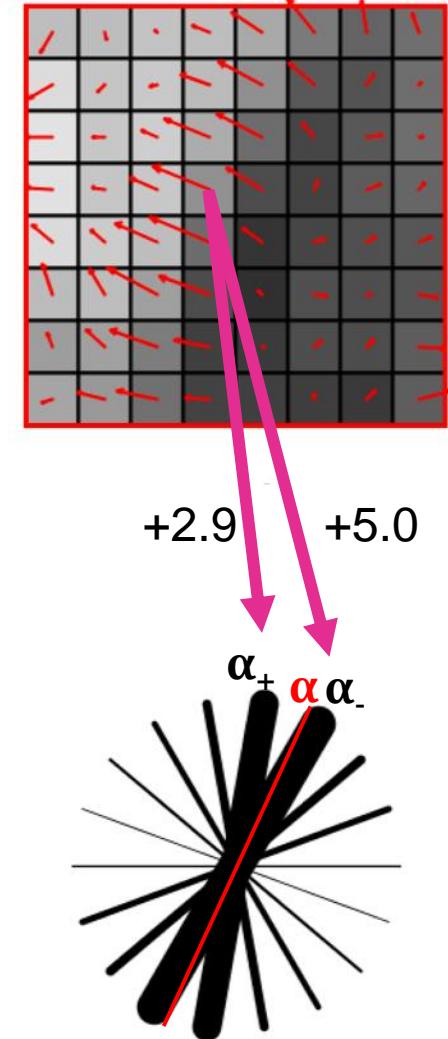
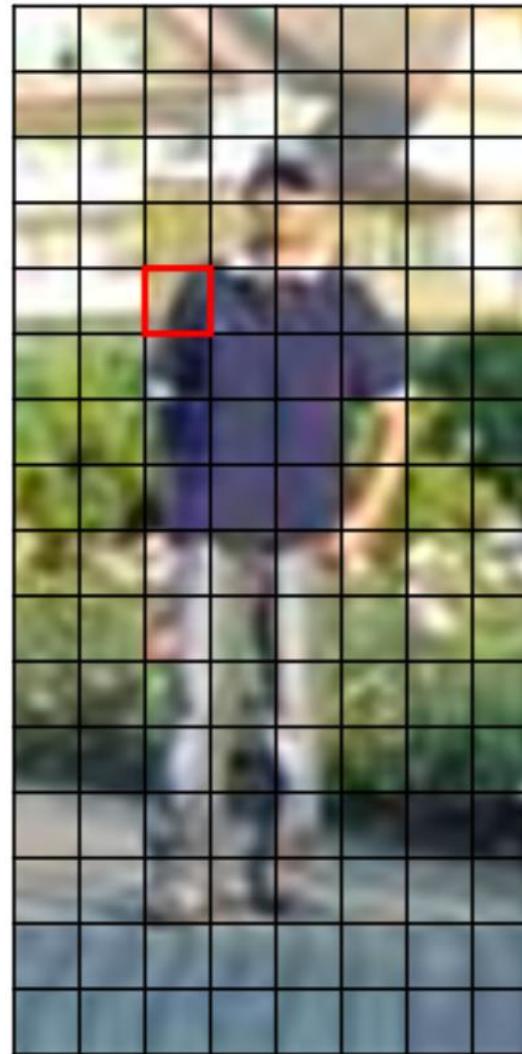
- „So viel Kante“ findet sich in diesem Rasterkästchen in dieser Rasterrichtung“
- Was bedeutet „So viel“?
- Anzahl Gradientenvektoren im Rasterkästchen, deren Senkrechte (α) am nächsten an der Rasterrichtung liegt ($\alpha_R +/- 10^\circ$)
 - Jeweils ‚+1‘ auf Eintrag α_R
 - Problem: Große Gradienten sollten mehr zählen
 - Richtung kleiner Gradienten ungenau
 - Schwellwert wäre „frühe harte Entscheidung“



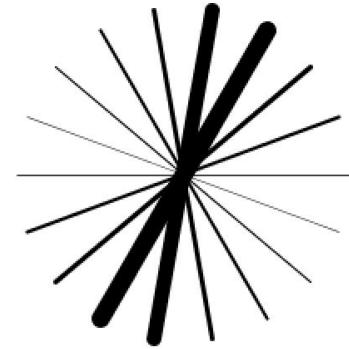
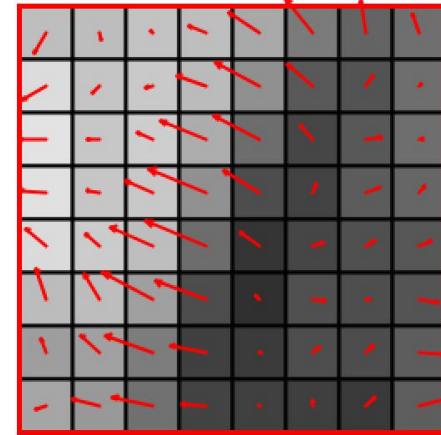
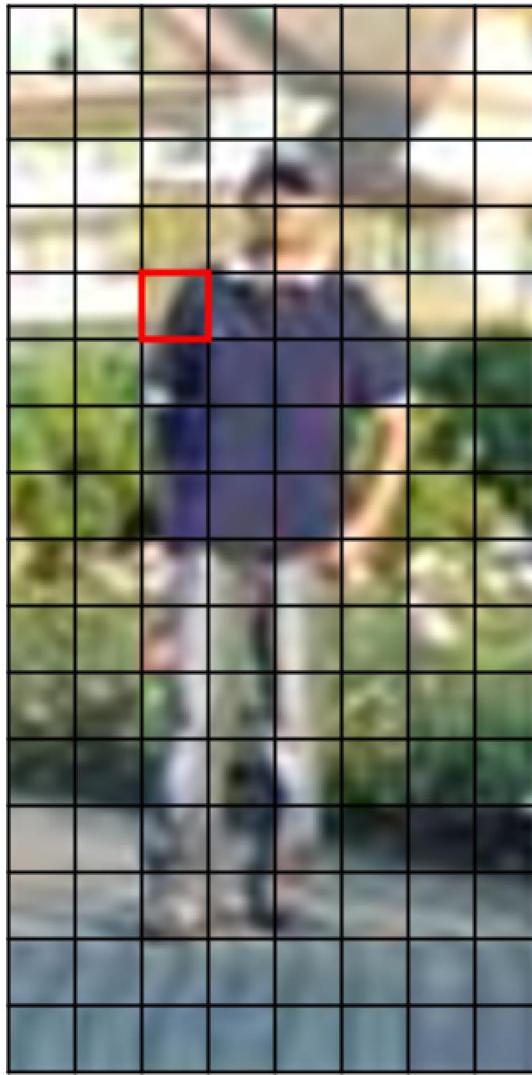
- Summe der Längen der Gradientenvektoren (\mathbf{g}), deren Senkrechte (α) am nächsten an der Rasterrichtung liegt ($\alpha_R \pm 10^\circ$)
 - Jeweils $+|g|$ auf Eintrag α_R
 - Im Beispiel: $|g|=7.9$
 - Entscheidung für eine Rasterrichtung ist auch hart und früh



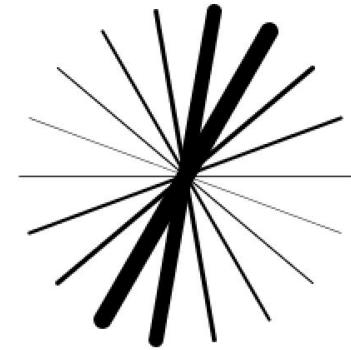
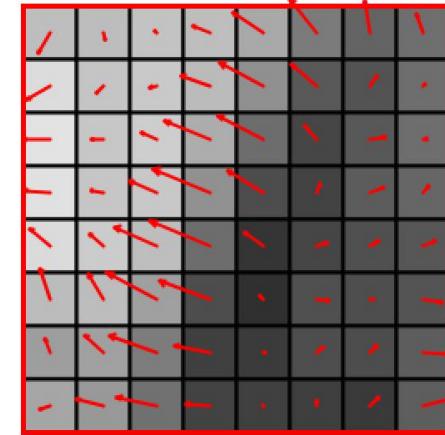
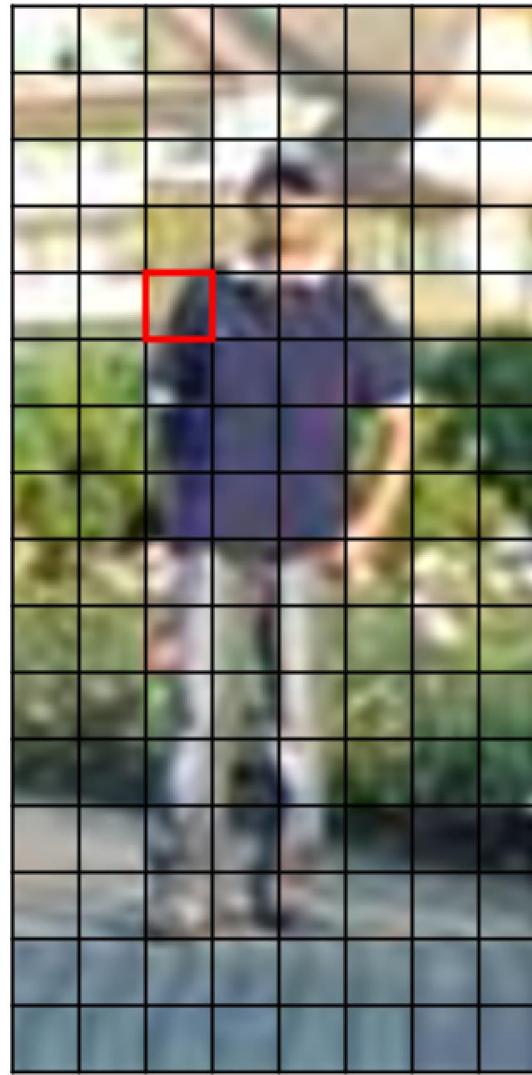
- Summe der Längen der Gradientenvektoren (\mathbf{g}) im Rasterkästchen, verteilt auf die beiden der Senkrechten (α) nächsten Rasterrichtungen ($\alpha_- < \alpha_+$)
 - $\frac{\alpha_+ - \alpha}{\alpha_+ - \alpha_-} |g|$ auf α_-
 - $\frac{\alpha - \alpha_-}{\alpha_+ - \alpha_-} |g|$ auf α_+
 - Im Beispiel 5.0 und 2.9



- Problem: Werte hängen von Helligkeit der Bildregion ab
 - alle Pixel $\cdot \lambda \rightarrow$ HoG $\cdot \lambda$
 - nicht unabhängig von Beleuchtung und Textilfarben

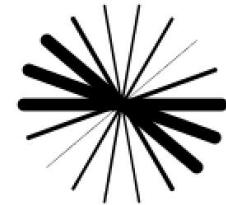
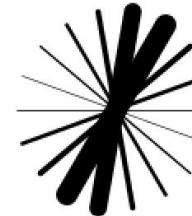
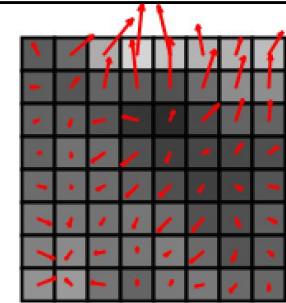
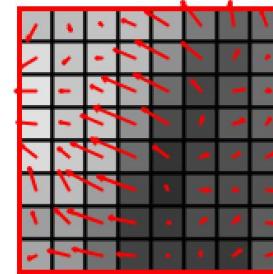
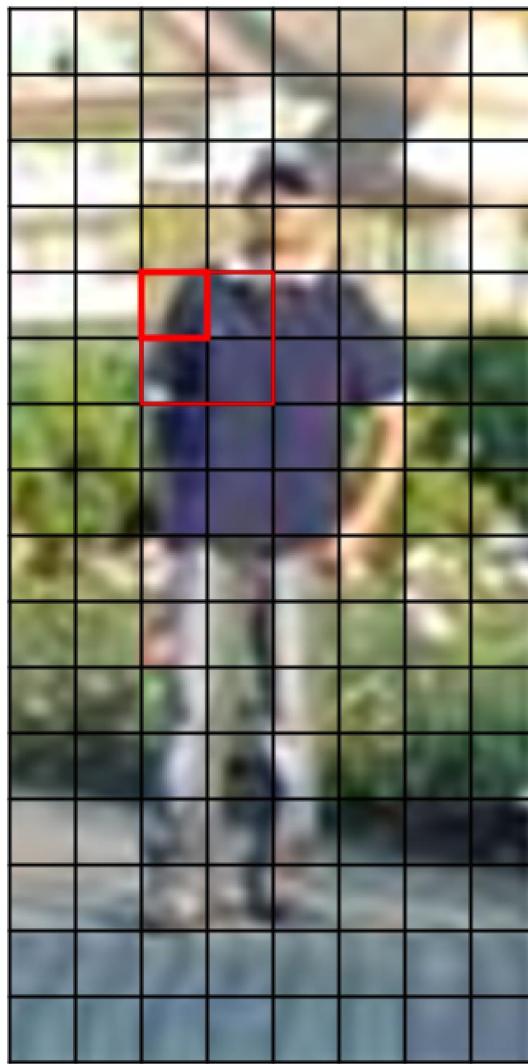


- Idee 3 + HoG als 9-D Vektor auf Länge 1 normieren
 - Alle Pixel· $\lambda \rightarrow$ HoG gleich
 - beschreibt Struktur lokaler Konturen relativ zu einem Bildkontext zur Normierung
 - → Kontext sollte größer sein als Struktur

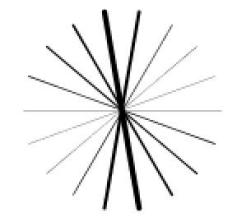
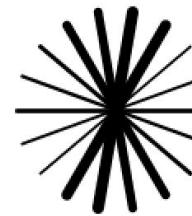
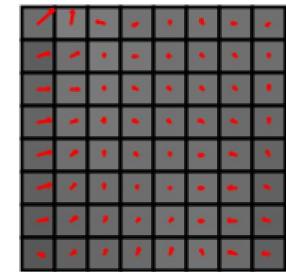
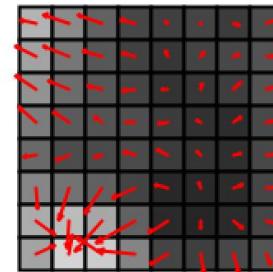


„So viel Kante“ – Idee 5

- Idee 3 + HoG von 2×2 Rasterkästchen als 36-D Vektor auf Länge 1 normieren

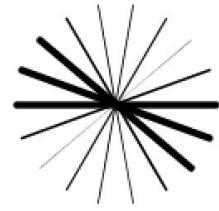
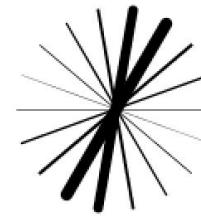
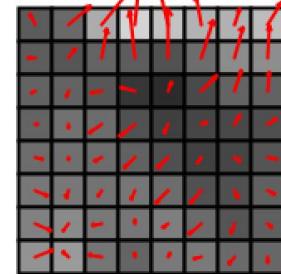
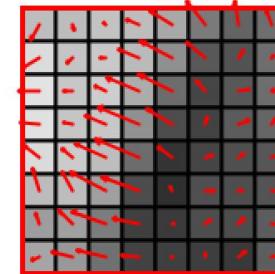
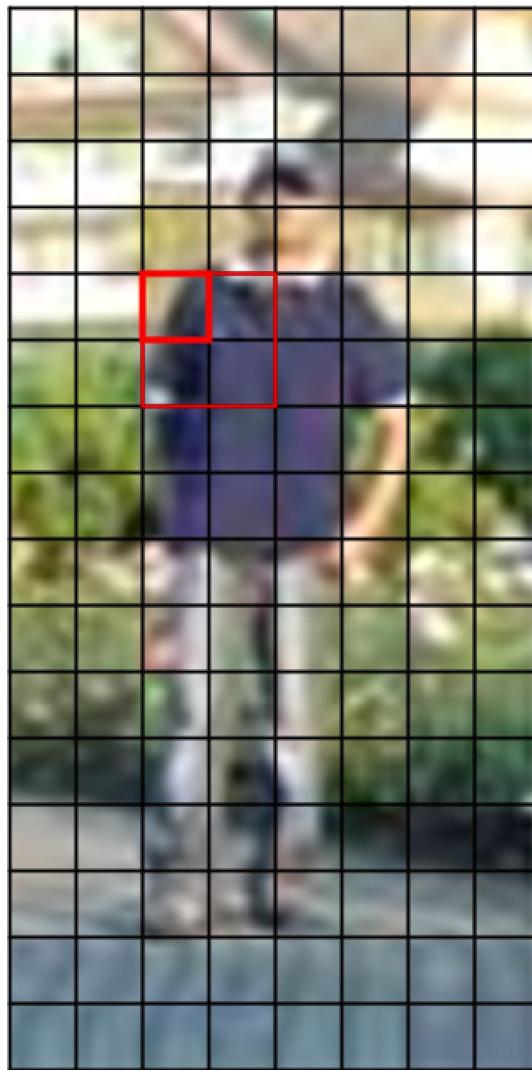


vor Normierung

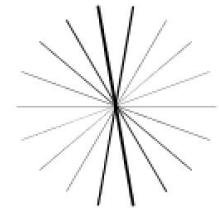
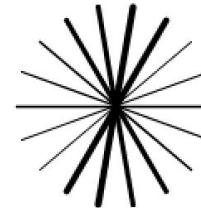
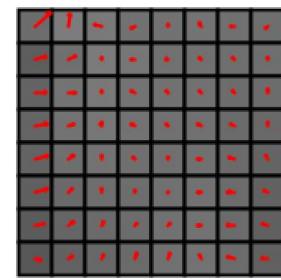
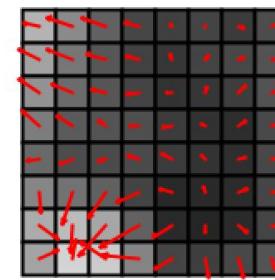


„So viel Kante“ – Idee 5

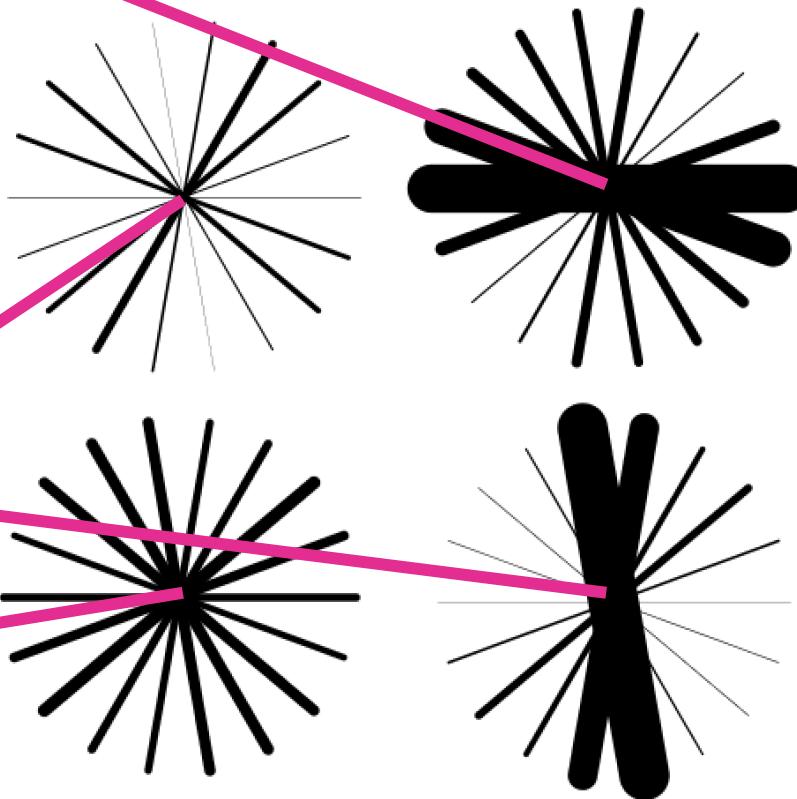
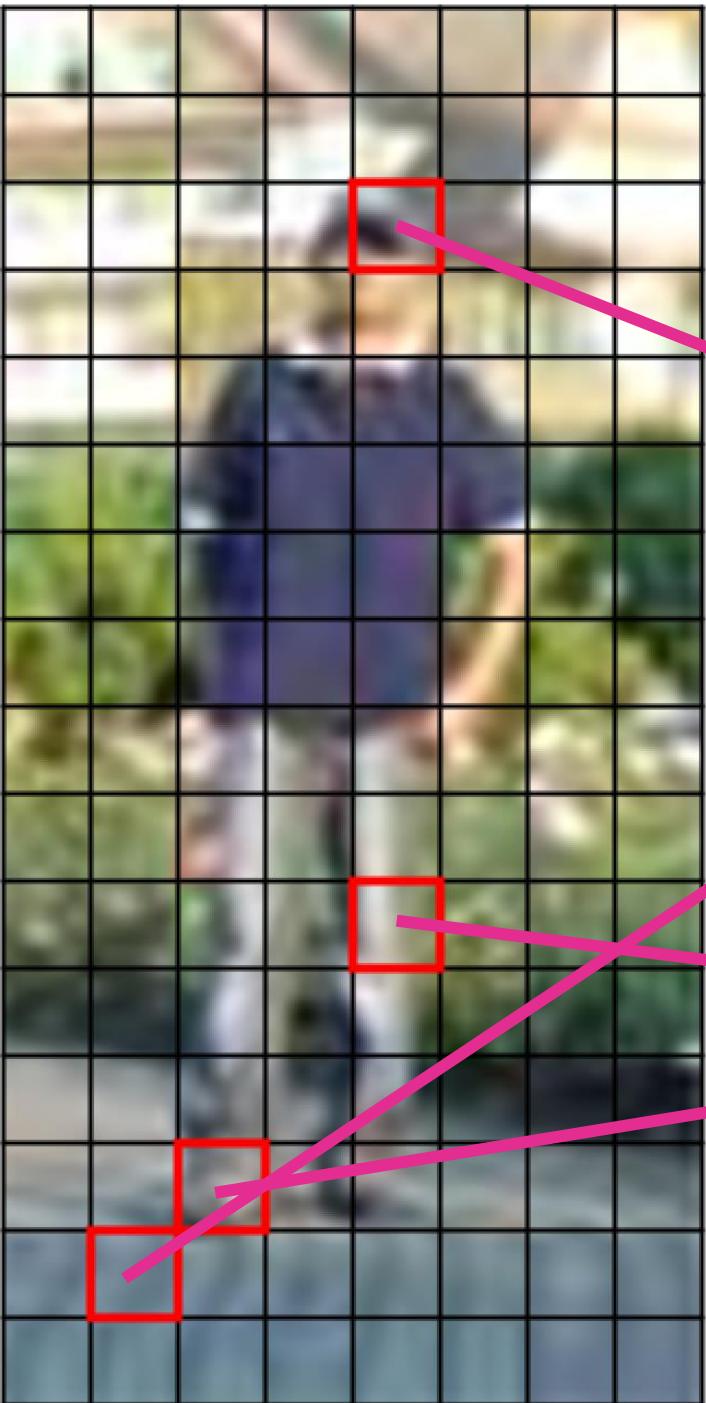
- Idee 3 + HoG von 2^*2 Rasterkästchen als 36-D Vektor auf Länge 1 normieren
 - Alle Pixel $\cdot \lambda \rightarrow$ HoG gleich
 - Empirisch belegt (Dalal et al.): besser



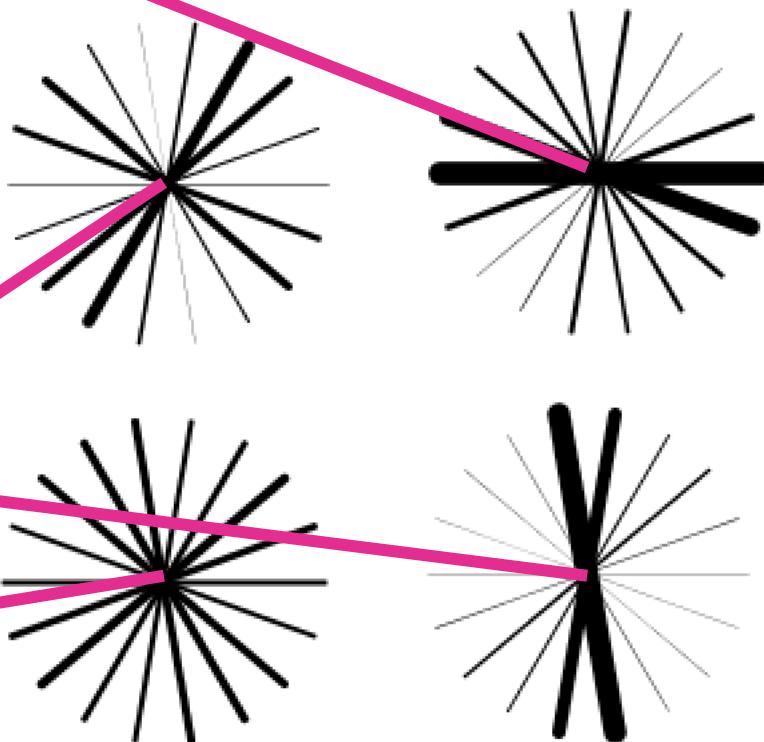
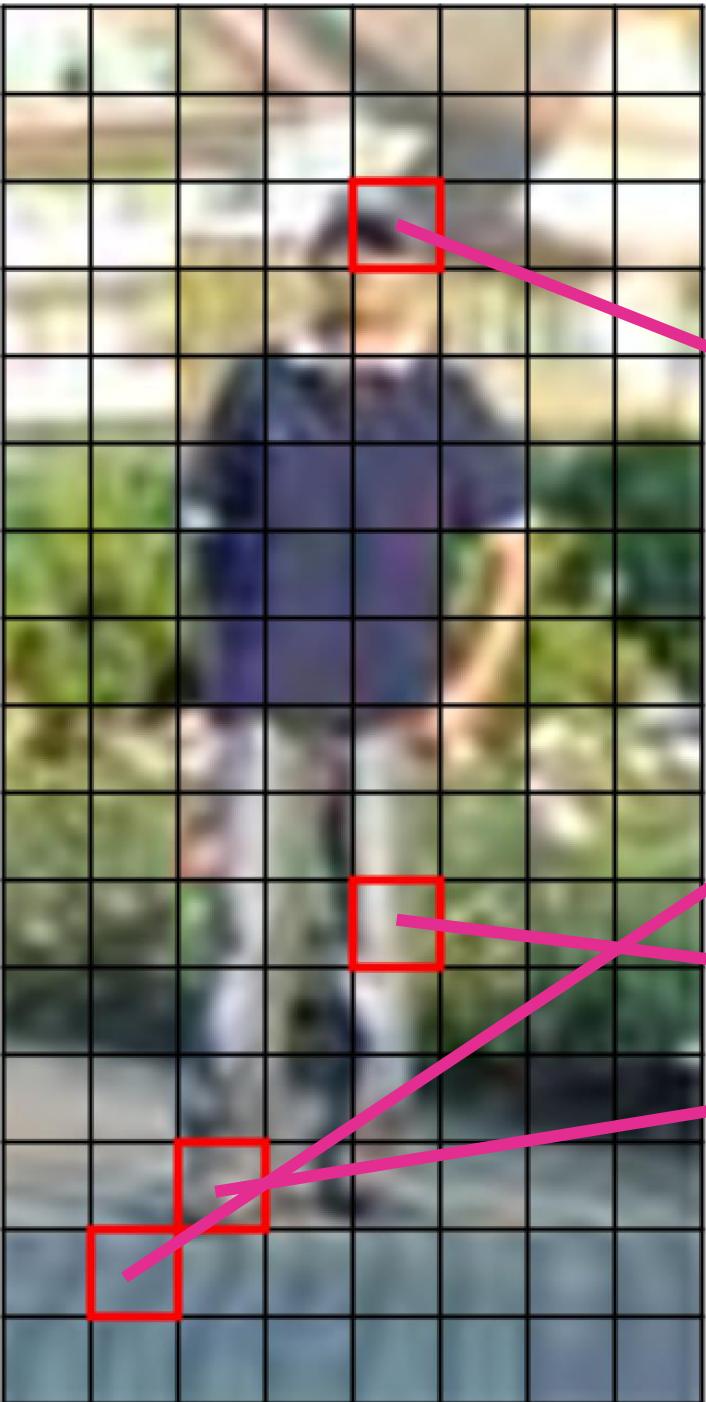
normiert



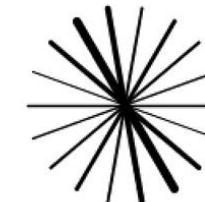
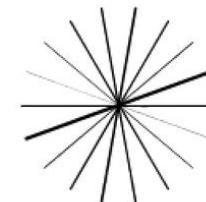
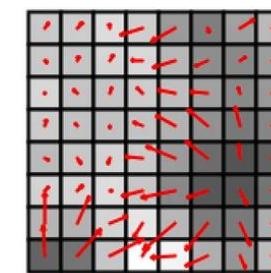
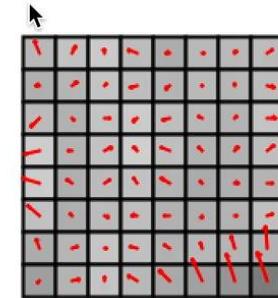
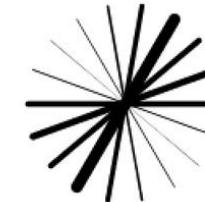
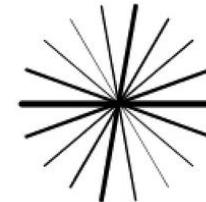
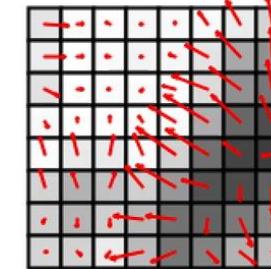
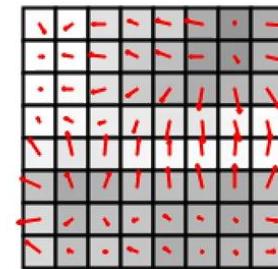
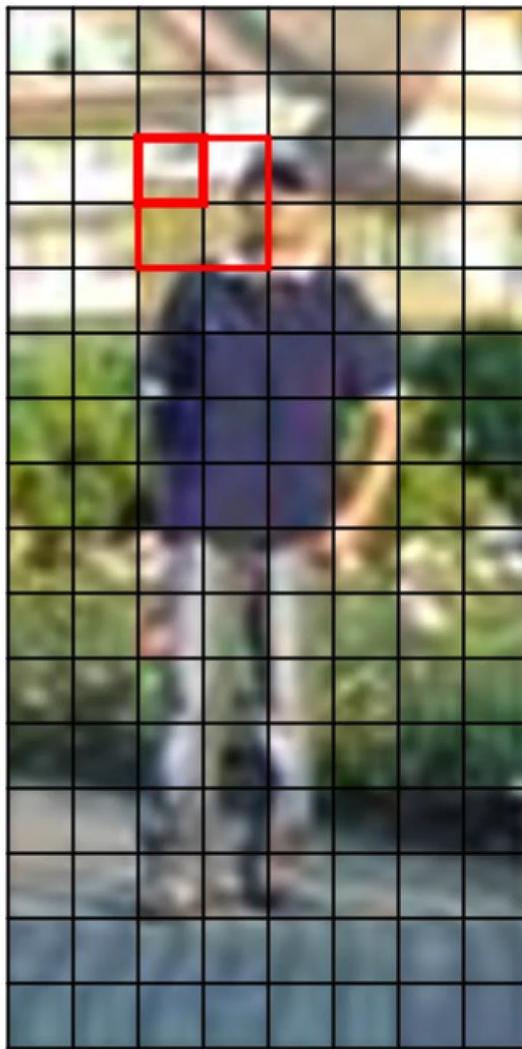
Vor der Normierung



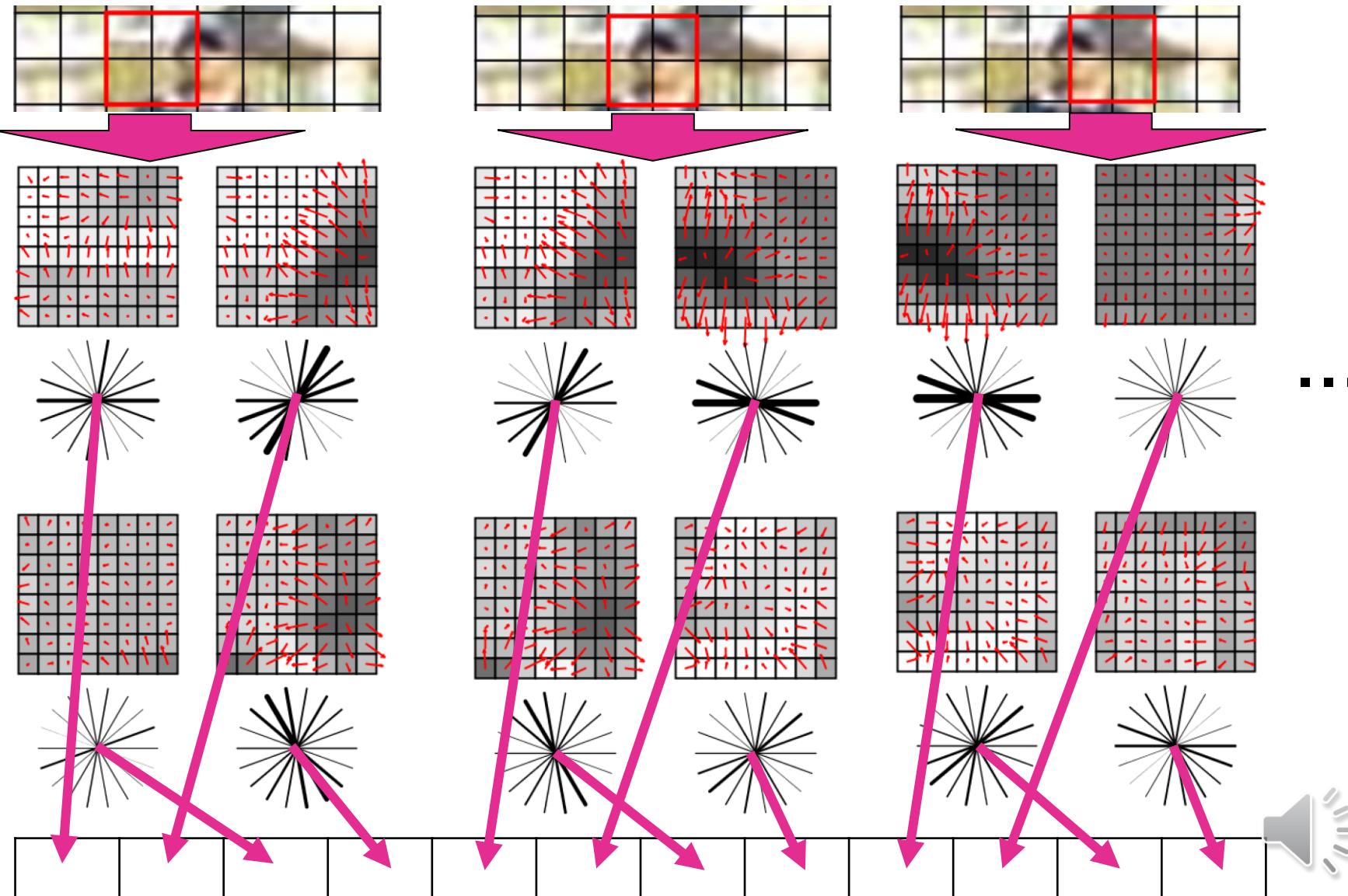
Nach der Normierung



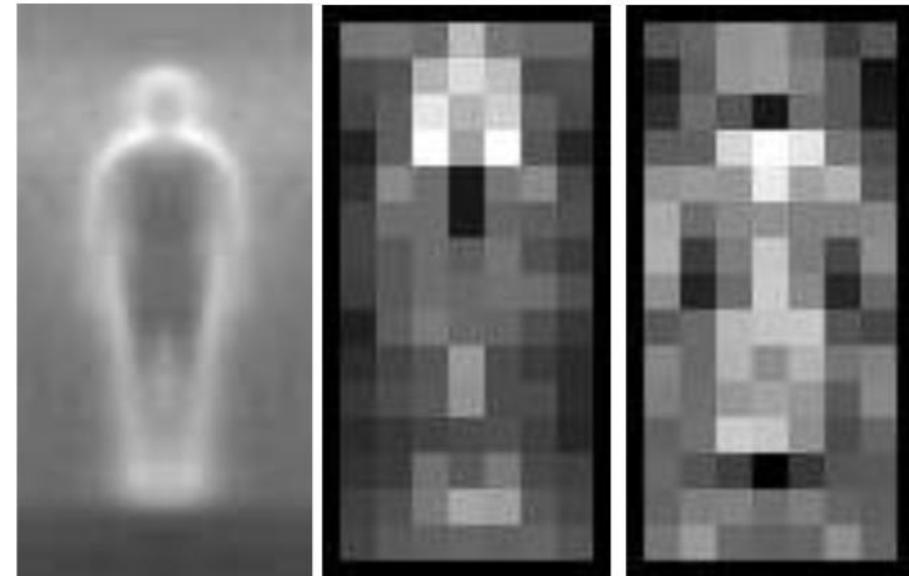
Histogram of Oriented Gradients



- Idee 6: Idee 5 + überlappende 2*2 Blöcke für Normalisierung
 - 2*2 Rasterkästchenblock immer 1 Rasterkästchen weiterziehen
 - → Jedes Rasterkästchen 4x unterschiedlich normalisiert im Merkmalsvektor
 - Empirisch belegt (Dalal et al.): besser



- Merkmalsvektor an Machine-Learning-Klassifizierer
 - vgl. VL Klassifizierungsalgorithmen
 - Trainingsdatensatz mit annotierten Personen
- Links: Mittleres Gradientenbild der Menschen im Trainingsdatensatz
 - wichtig: Alle Menschen mit einheitlichem Bildausschnitt
 - deshalb suche über Bild mit Rechtecken als Menschenhypothesen
- Mitte: Stellen im Raster, wo Einträge ein stark positives Indiz sind
 - vor allen Dingen Kopf
- Rechts: Stellen im Raster, wo Einträge ein stark negatives Indiz sind
 - senkrechte Kanten im Schulter-Bereich
 - waagerechte Kanten im Rumpf / Beine Bereich
- Hintergrundbereich ziemlich egal



- Histogram of Oriented Gradients (HoG)
 - Wo im Bild verlaufen Kanten in *welcher Richtung?*
 - in 8×8 Pixel- und 20° -Raster
 - Summe der Längen der Gradientenvektoren im Rasterkästchen, verteilt auf die beiden der Senkrechten nächsten Rasterrichtungen
 - Normierung in Blöcken von 2×2 Rasterkästchen

