

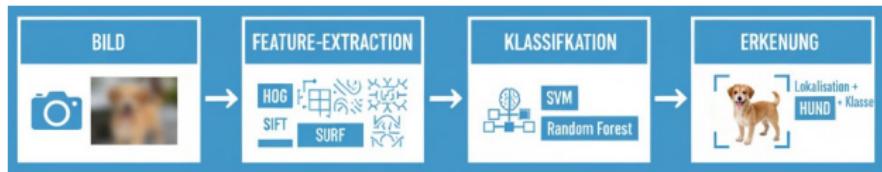
Computer Vision

Objekterkennung - Convolutional Neural Networks and YOLO

Alexandra Posekany

December 2025

Von Merkmalen zur (klassischen) Objekterkennung



Vorteile:

- ▶ Interpretierbar (wir wissen, was die Features sind)
- ▶ Weniger Daten benötigt (keine riesigen Datensätze)
- ▶ Schnell zu trainieren
- ▶ Gut für spezifische, eingeschränkte Aufgaben

Histogram of Oriented Gradients (HOG)

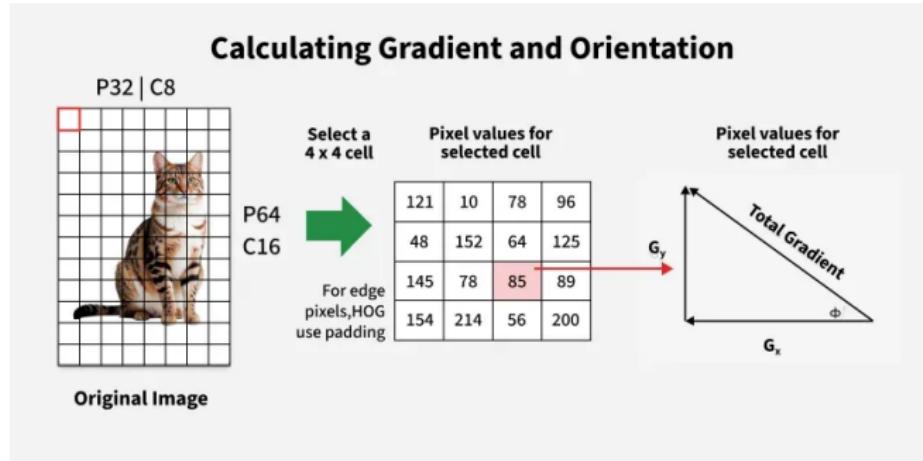
Motivation: Menschliche Wahrnehmung

Idee: Die Form eines Objekts (z.B. Mensch) lässt sich sehr gut durch die Verteilung von Kantenrichtungen beschreiben.

- ▶ Lokale Gradientenrichtungen sind robust gegenüber Beleuchtungsänderungen
- ▶ Die Verteilung (Histogramm) dieser Richtungen beschreibt die Form

HOG Algorithmus visuell

1. Bild in kleine Zellen (z.B. 8×8 Pixel) unterteilen.



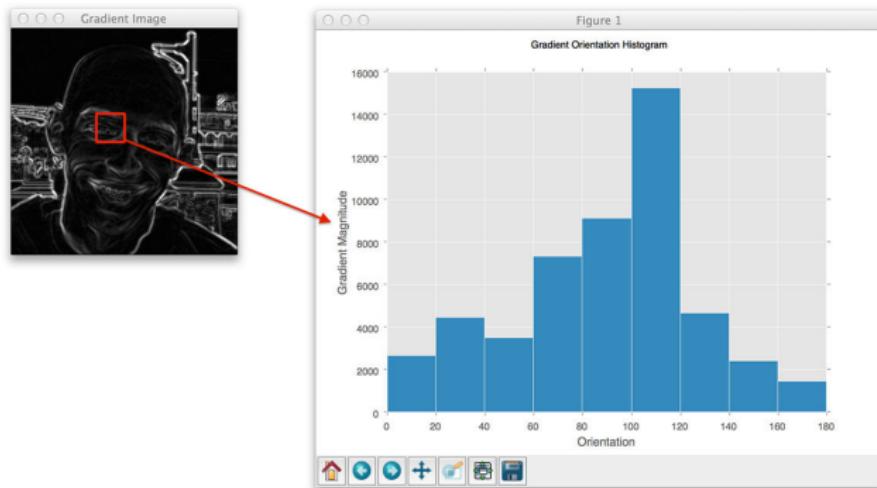
2. In jeder Zelle: Berechne Gradienten (Richtung und Stärke).

$$G_x = I * [-1, 0, 1], \quad G_y = I * \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

$$|G| = \sqrt{G_x^2 + G_y^2}, \quad \theta = \arctan\left(\frac{G_y}{G_x}\right)$$

HOG Algorithmus visuell

3. Erstelle ein Histogramm der Richtungen (z.B. "Viel vertikal, wenig horizontal").



Ergebnis: Eine vereinfachte Darstellung, die den "Umriss" einfängt, aber Farben ignoriert.

Support Vector Machine

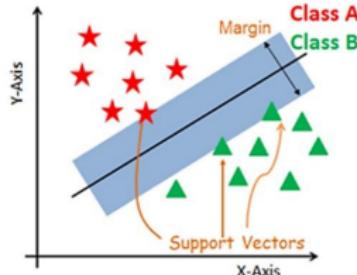
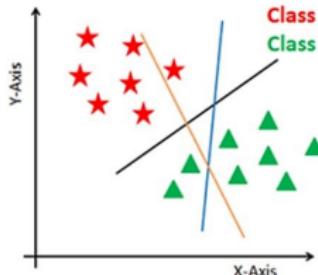
Wir haben den HOG-Vektor (Features). Jetzt brauchen wir eine Entscheidung, welches "Label" (Klasse) wir vergeben.

Linear SVM (Support Vector Machine):

Findet die Hyperebene (Trennlinie), die die Klassen voneinander im hochdimensionalen Raum am besten trennt (größtmöglicher Sicherheitsabstand / Margin).



SUPPORT VECTOR MACHINE



Das Such-Problem (Sliding Window)

Ein Klassifikator (SVM/CNN) kann nur sagen "Ja/Nein" für einen Bildausschnitt.

Wie finden wir das Objekt im großen Bild?

Sliding Window Ansatz:

- ▶ Schiebe ein Fenster fester Größe über das Bild (von links oben nach rechts unten).
- ▶ Wiederhole das für verschiedene Skalierungen (Bildpyramide), um kleine und große Objekte zu finden.

Problem:

Extrem rechenintensiv!

Ein VGA Bild (640×480) generiert zehntausende Fenster.

99,9% der Fenster sind Hintergrund.

Region Proposals (Der smarte Weg)

Statt "dumm" überall zu suchen, suchen wir nur dort, wo "etwas" ist.

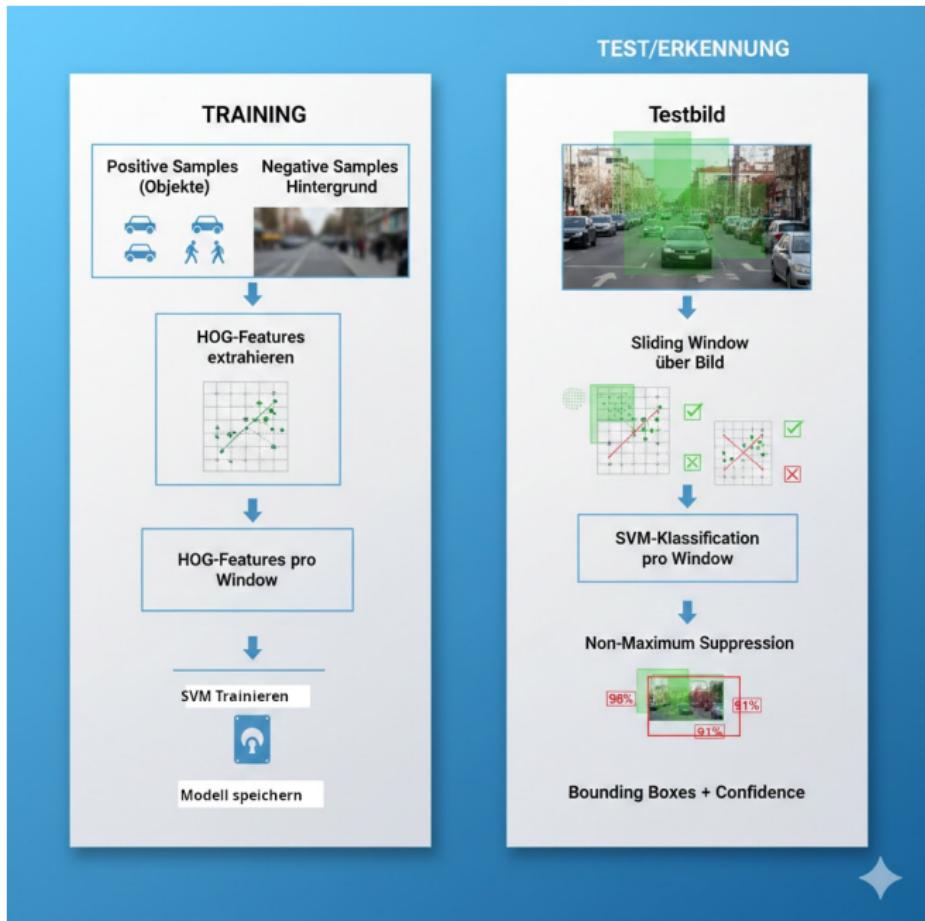
Selective Search:

- ▶ Pixel basierend auf Farbe/Textur zu Blobs zusammenfassen (Superpixel).
- ▶ Generiert ca. 2.000 Vorschläge (Proposals) statt 100.000 Fenster.

Diese Vorschläge ("Blobs") werden dann dem Klassifikator übergeben.

Vorläufer von R-CNN: R-CNN nutzte genau diesen Selective Search Ansatz.

Ablauf



Warum reicht das nicht? (Limitierungen)

- ▶ Geschwindigkeit: Sliding Window ist zu langsam für Echtzeit.
- ▶ Region Proposals (Selective Search) sind schneller, aber immer noch ein Flaschenhals (CPU-basiert).
- ▶ Hand-Crafted Features: HOG ist gut für stehende Menschen, aber schlecht für deformierbare Objekte (z.B. schlafende Katze).
- ▶ Pipeline-Trennung:
 - ▶ Schritt 1 (Proposals) und Schritt 2 (Klassifikation) sind getrennt.
 - ▶ Der Klassifikator kann den Proposal-Algorithmus nicht korrigieren (“Du hast den Kopf abgeschnitten!”).

Lösung: Wir brauchen eine Methode, die gleichzeitig guckt (wo?) und erkennt (was?). → YOLO (You Only Look Once).

Klassische Pipeline Objekterkennung

