KRISHAI TECHNOLOGIES

Haar Cascades For Face Detection

www.krishnaik.in

Haar Cascades are machine-learning based object-detection algorithms introduced by Paul Viola and Michael Jones in 2001. They rely on Haar-like features and use a cascade of classifiers to identify objects within an image.

Haar-Like Features

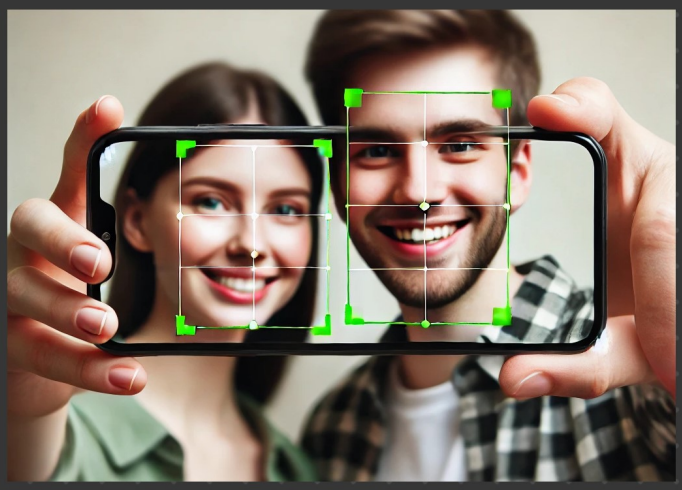
- Haar-like features are rectangular patterns that compare the intensity of adjacent regions in an image. These features are used to detect contrast in pixel intensity, which help identify structures like edges, lines, and textures.
- Think of Haar-like features as templates or filters that detect specific patterns in an image.

Cascading Classifiers

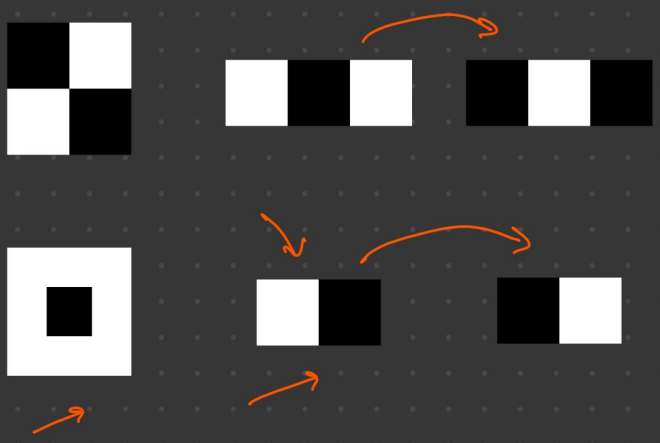
- The algorithm applies multiple stages (a cascade) of classifiers sequentially. If a region fails in one stage, it's immediately discarded. This makes Haar Cascades computationally efficient.
- These are Boosted ML algorithm which are boosted cascade of weak classifiers and the choice of classifier is chosen from : DAB - Discrete AdaBoost, RAB - Real AdaBoost, LB - LogitBoost, GAB - Gentle AdaBoost

Use Cases

- Face detection based selfie camera application.
- Person and Car detections in Smart Security Systems
- Person detection in Robotics for interactions like automatic greeting and tracking person or avoiding collisions.



Haar-like Features



Haar-like Features

- Haar-like features are rectangular patterns that compare the intensity of adjacent regions in an image.
- These features are used to detect contrasts in pixel intensity, which help identify structures like edges, lines, and textures.
- Examples of Haar-Like Features:
  - Edge Feature: Distinguishes a dark region from a bright region, like the edge of a nose.
  - Line Feature: Detects horizontal or vertical lines, like the bridge of the nose or the mouth.
  - Four-Rectangle Feature: Detects diagonal differences in pixel intensity, often used for detecting eyes or windows.
- Haar features are calculated by summing up the pixel intensities in the rectangular regions and comparing their differences.

1000 → 8 px → True

Haar similarity = Sum of white region - Sum of black region

Image ≥ 0.5 → Bright

0 - 1

1	1	0	0
1	1	0	0

0.5 0.9 0.1 0.3

0.6 0.7 0.4 0

→ Haar = 4 - 0
= 4

→ Image 1 =
(0.5 + 0.9 + 0.6 + 0.7) -
(0.1 + 0.3 + 0.4 + 0)
= 1.9

↓ rotate 180°

0 0.4 0.7 0.6

1	1	0	0
1	1	0	0

0.3 0.1 0.9 0.5

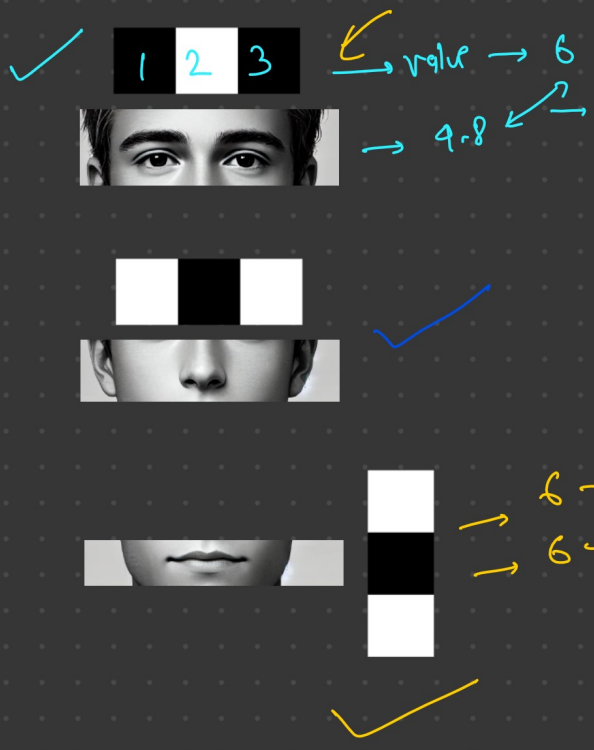
→ Image 2 =
(0.1 + 0.3 + 0.4 + 0) -
(0.5 + 0.9 + 0.6 + 0.7)
= -1.9

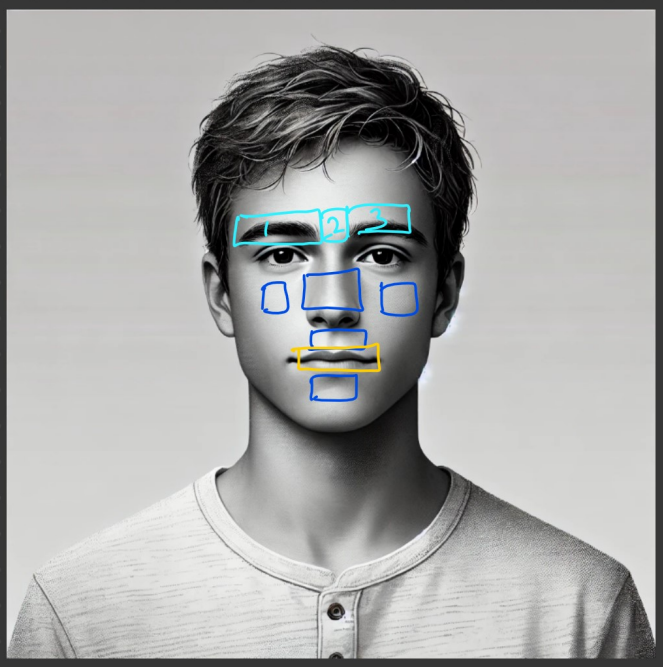
Image 1 ← Haar → Image 2
1.9 ← 4 → -1.9

N = 10

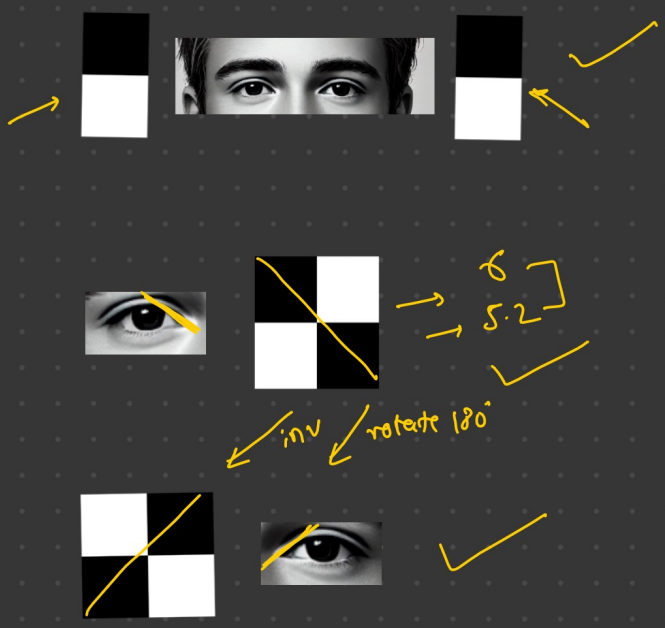
1 → 3 → N → Face detect
→ No face

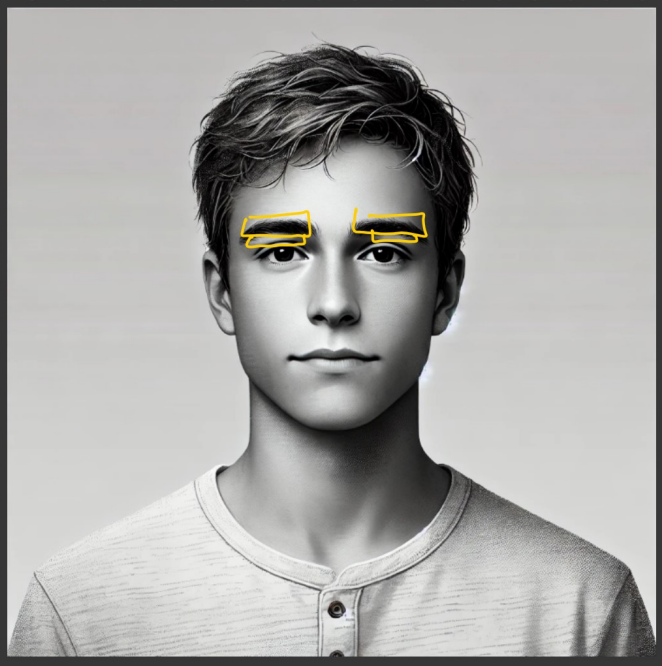
Stage 1 - Validation



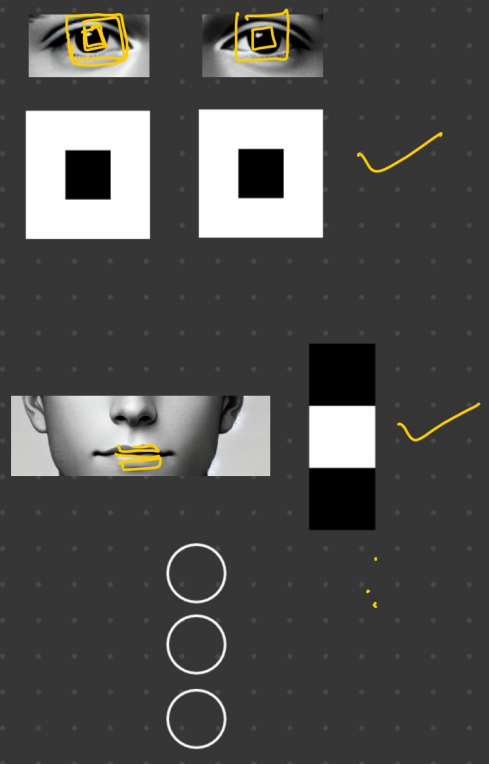


Stage 2 - Validation





Stage 3 - Validation



Stage N - Validation

No face detected

Face Detected

→ x, y, w, h →