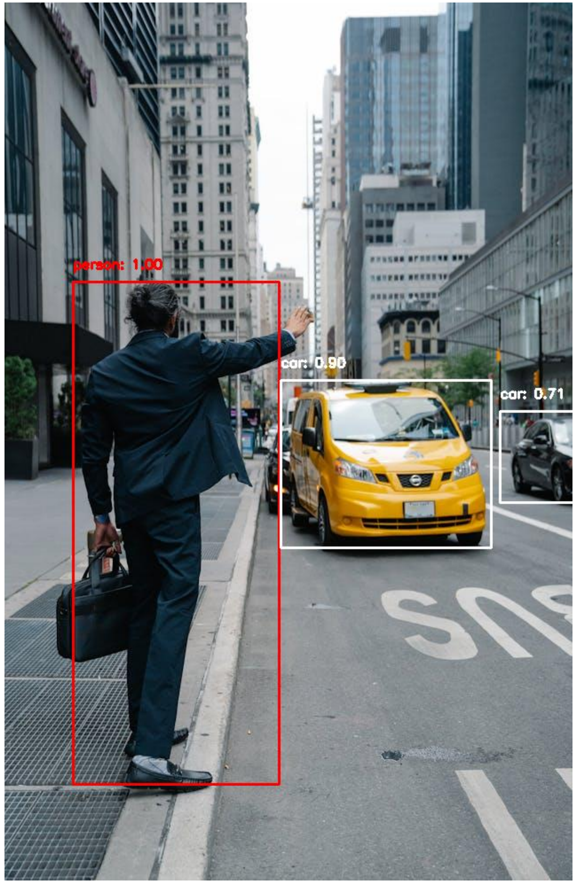
Introductory seminar on Computer Vision

Sara Concas, Cagliari Digital Lab 2024 - Day 4

sara.concas90c@unica.it

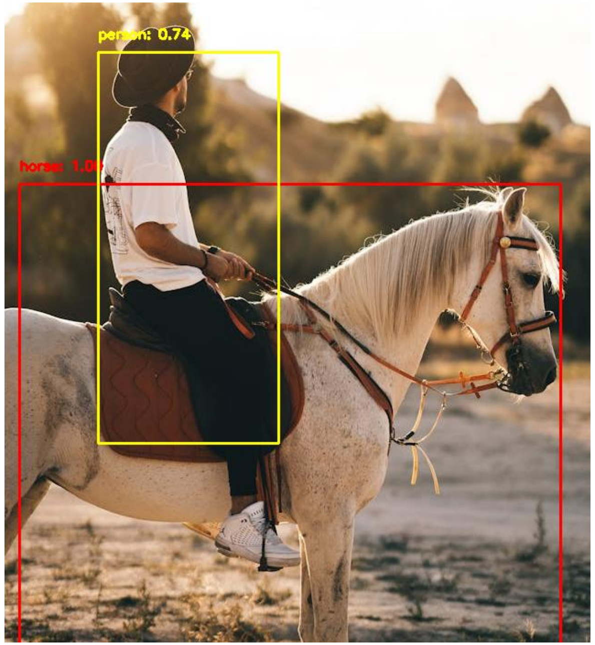
Object Detection

GOAL: identify and locate objects of interest within an image or video. Object detection not only classifies objects but also provides bounding boxes that indicate their position.



Object Detection

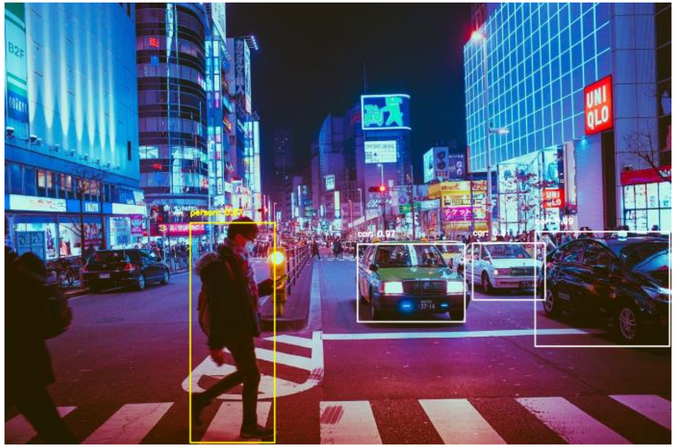
Input: image



Output: set of bounding boxes with class labels. Each bounding box is a rectangle that surrounds the detected object.

Object Detection - Use Cases

Retail: e.g., detecting products on shelves.
https://www.labelvisor.com/enhancing-retail-analytics-with-yolov8-object-detection/



Autonomous driving: detecting pedestrians, vehicles, road signs, etc.

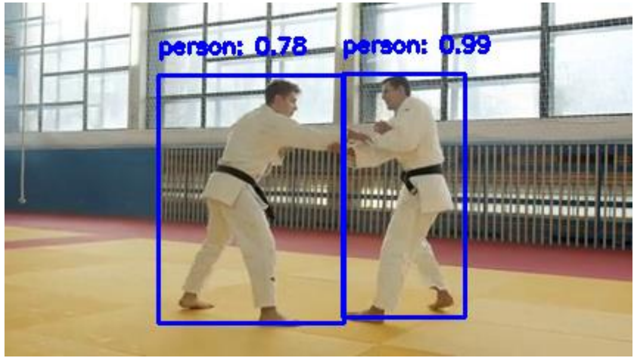
Surveillance: e.g. monitoring of a street view.
https://medium.com/nanonets/how-to-automate-surveillance-easily-with-deep-learning-4eb4fa0cd68d

Object Detection VS Segmentation

Object Detection

Task: identify and locate objects of interest within an image or video.

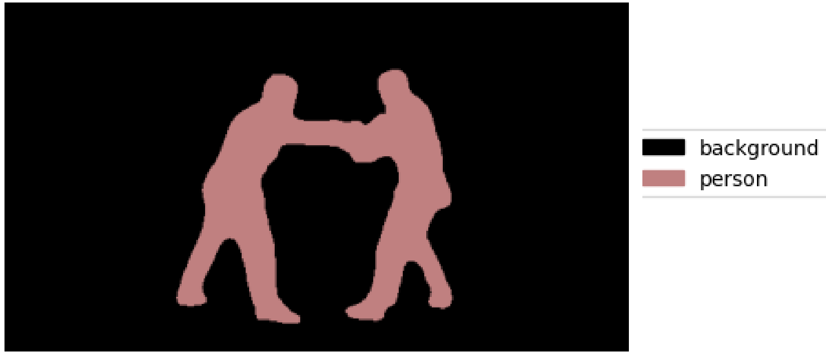
Output: set of bounding boxes with class labels. Each bounding box is a rectangle that surrounds the detected object.



Segmentation

Task: classify each pixel in an image.

Output: pixel-wise mask that indicates the class label for each pixel.

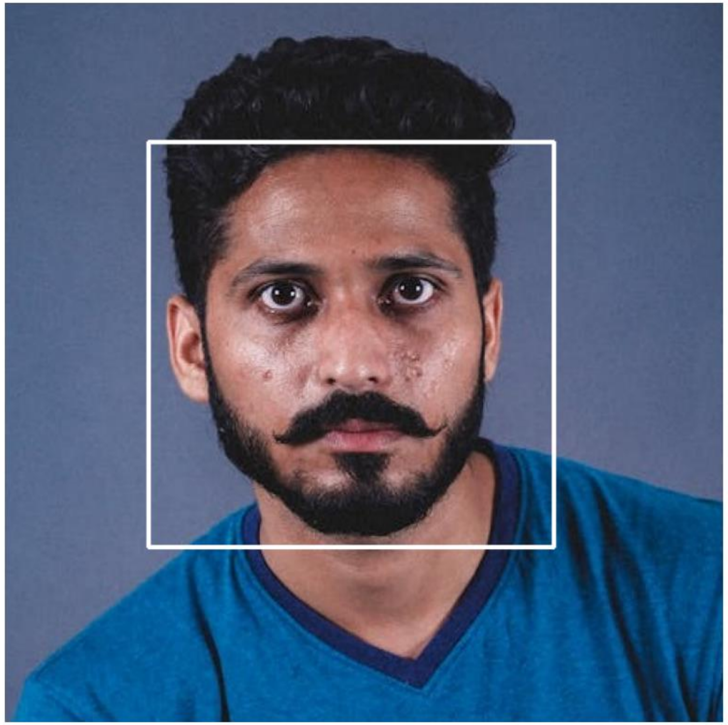


Viola-Jones Object detection

Popular and effective method for object detection, particularly known for its application in real-time face detection.
 Developed by Paul Viola and Michael Jones in 2001, it is one of the first frameworks that enabled rapid object detection.

It consists of 4 stages:

- Selecting Haar-like features
- Creating an integral image
- Running AdaBoost training
- Creating classifier cascades

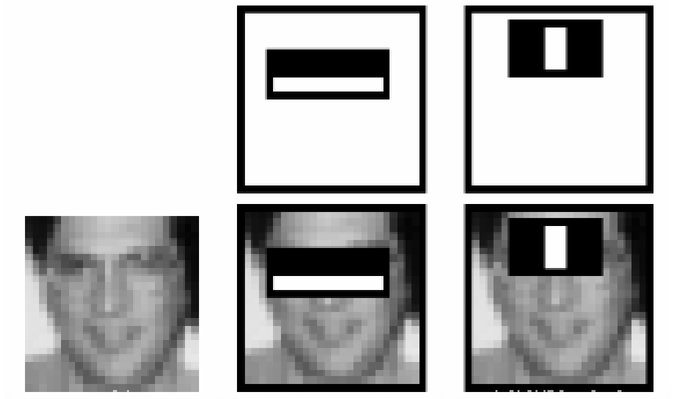


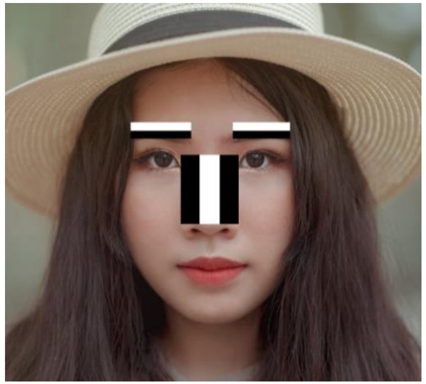
Viola-Jones Object detection

Selecting Haar-like features

Haar features are digital image features. These features are simple rectangular patterns that compare the intensity of adjacent rectangular regions in an image.

For example, one common feature for detecting faces is a pair of adjacent rectangles, where one is over the eyes and the other is over the cheeks. The difference in pixel intensities between these regions can indicate the presence of a feature (such as the edge of the nose or eyes).



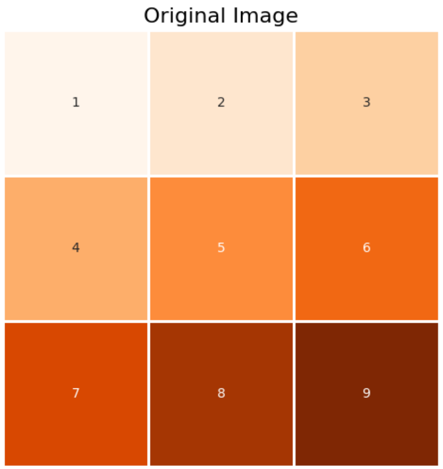


https://docs.opencv.org/4.x/d2/d99/tutorial_js_face_detection.html

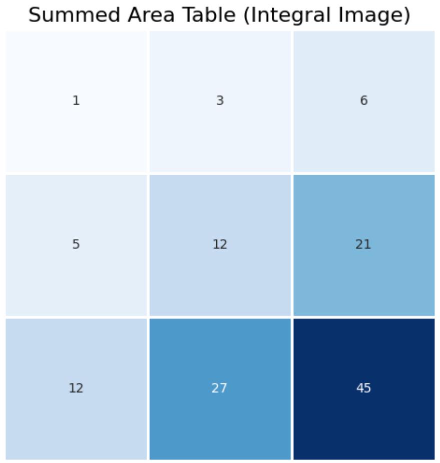
Viola-Jones Object detection

Creating an integral image

The integral image is a data structure where each pixel contains the sum of all pixel values to the left and above it. This allows for fast computation of the sum of any rectangular region in constant time.



1	2	3
4	5	6
7	8	9



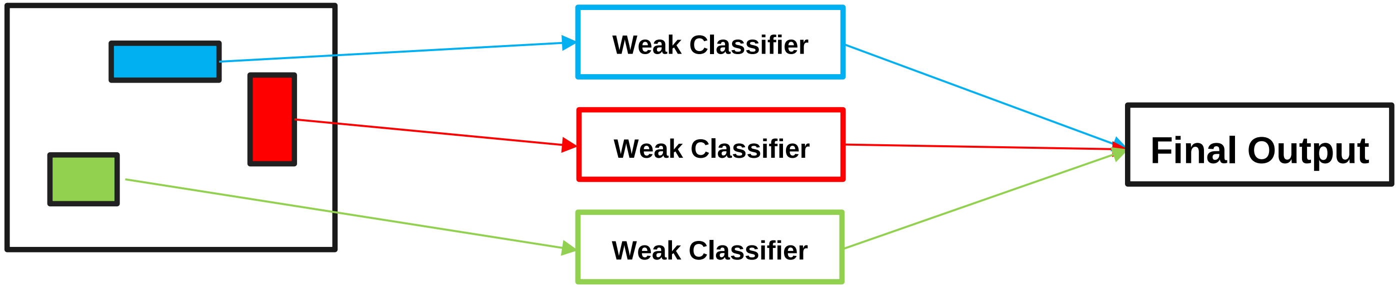
1	3	6
5	12	21
12	27	45

Viola-Jones Object detection

Running AdaBoost (Adaptive Boosting) training

The algorithm uses a machine learning technique called AdaBoost (Adaptive Boosting) to select the most effective features for detecting objects. It combines weak classifiers to form a strong classifier. Each weak classifier is a single Haar-like feature that can distinguish between object and non-object regions.

During training, AdaBoost selects a subset of the most critical Haar-like features and assigns them weights based on their accuracy. The result is a strong classifier that combines these weighted features to make a decision.

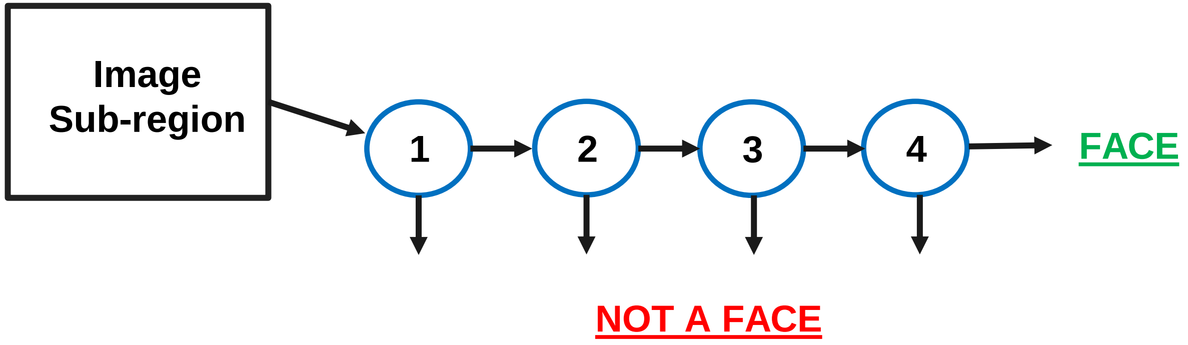


https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html

Viola-Jones Object detection

Creating classifier cascades

A cascade of classifiers is used to improve detection speed.
The cascade consists of several stages, where each stage is a classifier trained to detect objects. If a region passes the first stage, it moves on to the next stage, where a more complex classifier evaluates it.

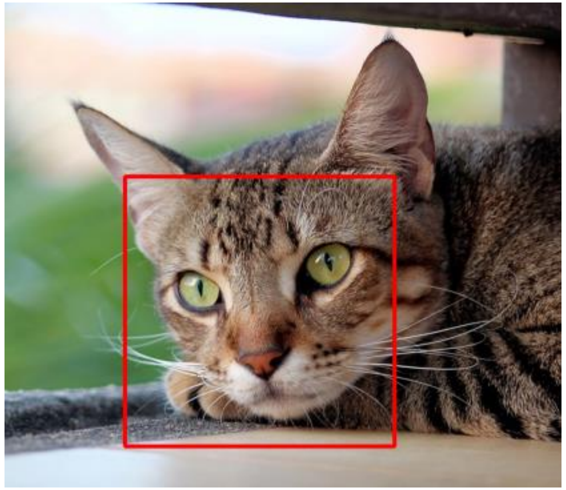


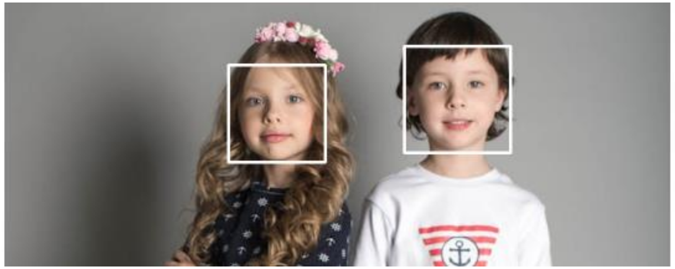
If the region passes the first stage, it moves on to the next, where a more complex classifier evaluates it.
The classifiers become more accurate but also more computationally expensive

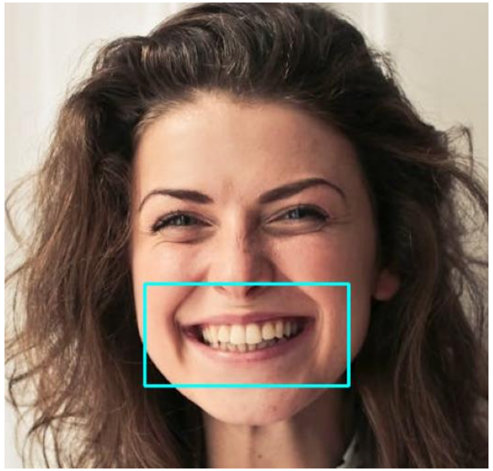
Viola-Jones Object detection - Examples

It can be used to detect different objects, especially when they have well-defined and somewhat rigid structures, with particularly consistent visual patterns across different images.

Some examples





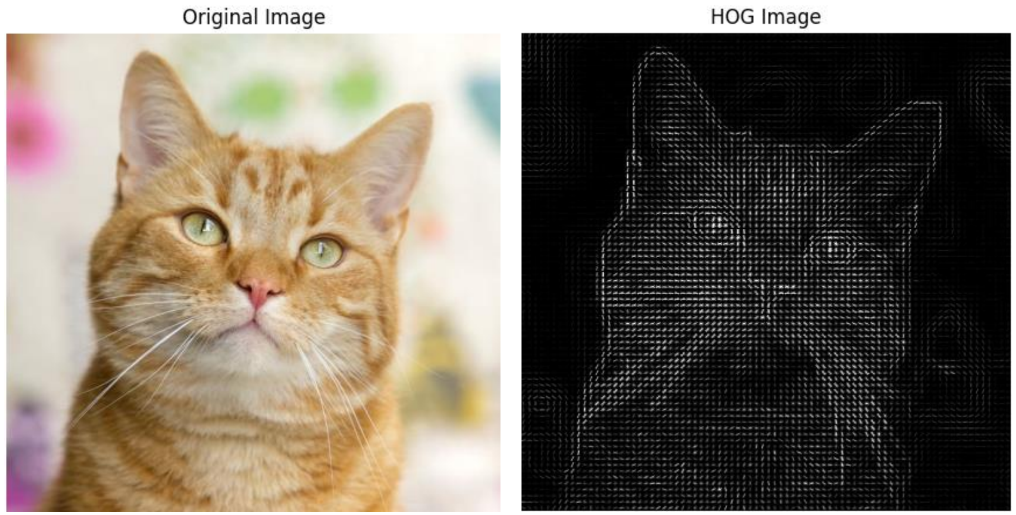


https://bit.ly/3z4Kq0x

Histogram of Oriented Gradients (HOG)

Feature descriptor that captures the appearance and shape of an object by encoding the distribution of gradient directions (oriented gradients) in localized portions of an image.

It is particularly effective at capturing the edges and contours of objects, which makes it robust for detecting objects with distinct shapes and structures, such as people or vehicles.



Histogram of Oriented Gradients (HOG)

Gradients and Edge Information

The gradient is computed at each pixel of the image. It represents the change in intensity (brightness) between neighboring pixels, which highlights edges and contours in the image.

The gradient has 2 components:

- **magnitude** (strength of the edge)
- **direction** (orientation of the edge).

For a given pixel $I(x,y)$ in an image, the gradient at that pixel is calculated by measuring the change in intensity in the horizontal (x) and vertical (y) directions.

Horizontal Gradient

$$G_x(x,y) = I(x+1,y) - I(x-1,y)$$

Vertical Gradient

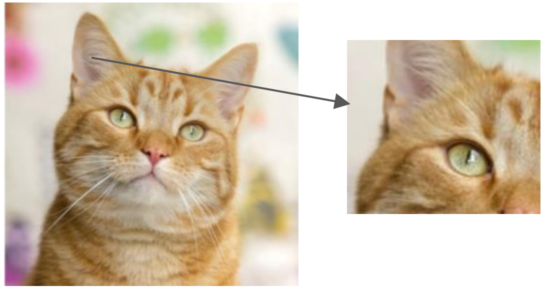
$$G_y(x,y) = I(x,y+1) - I(x,y-1)$$

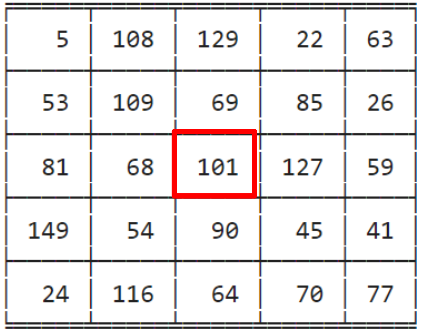
Magnitude $$\boldsymbol{M(x,y)} = \sqrt{G_x(x,y)^2 + G_y(x,y)^2}$$

Direction $$\boldsymbol{\theta(x,y)} = atan2(G_y(x,y)\ ,G_x(x,y))$$

Histogram of Oriented Gradients (HOG)

Gradients and Edge Information





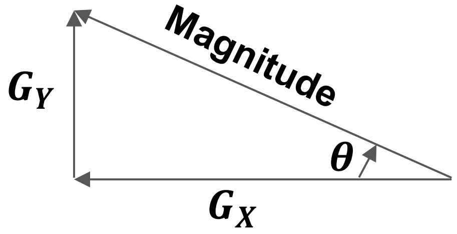
5	108	129	22	63
53	109	69	85	26
81	68	101	127	59
149	54	90	45	41
24	116	64	70	77

Horizontal gradient G_x = 127-68 = 59

Vertical gradient G_y = 69-90 = -21

This is repeated for each pixel in the image. After this process, we will have 2 new matrices

Magnitude and Orientation are computed using Pythagoras theorem

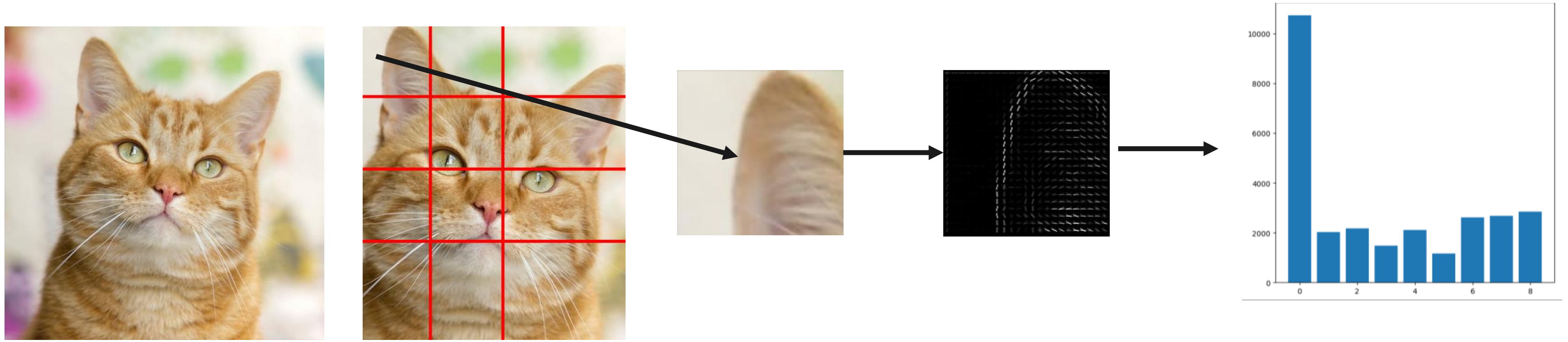


Finally we will have the Magnitude and Orientation for each pixel

Histogram of Oriented Gradients (HOG)

Cell Division

The image is divided into small, connected regions called cells.
Within each cell, the gradient directions of all the pixels are accumulated into a histogram.
The magnitude of the gradient at each pixel contributes to the corresponding orientation bin in the histogram.
This process captures the local shape information.



Histogram of Oriented Gradients (HOG)

Block Normalization

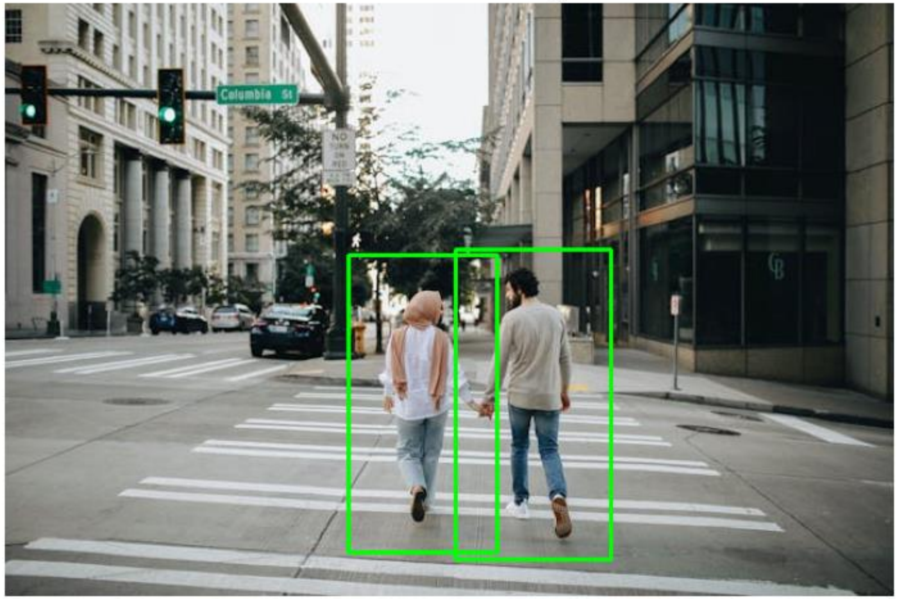
To make the HOG features robust to lighting conditions (i.e., changes in brightness and contrast), normalization is applied.
This is done by grouping cells into larger regions called blocks. The histograms within a block are normalized together to reduce the effects of local variations in illumination and shadows. This normalization ensures that the features are more invariant to changes in lighting and contrast.

Descriptor Construction

The final HOG descriptor is constructed by concatenating the histograms into a single, high-dimensional feature vector.
This is used as input to a machine learning algorithm.

Histogram of Oriented Gradients (HOG)

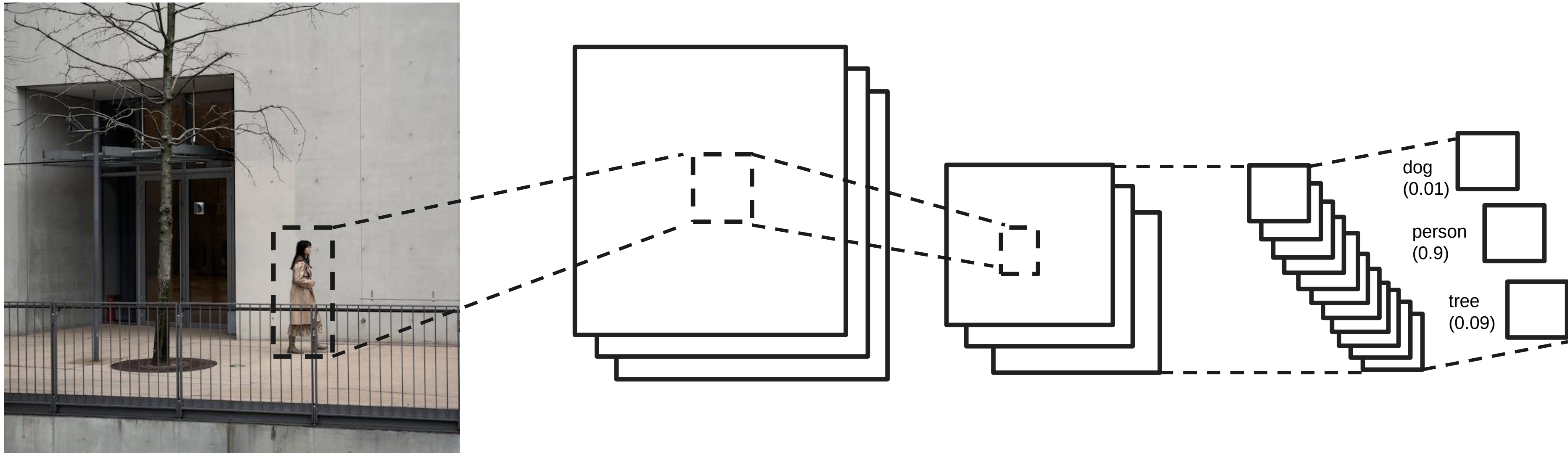
Detected Pedestrians using HOG



https://bit.ly/3T11V8Q

Convolutional Neural Networks (CNNs)

CNNs are particularly powerful for this task because they can learn spatial hierarchies of features through convolutional layers.
The hierarchical nature of CNNs enables them to detect low-level features like edges in the initial layers and more complex features like object parts in deeper layers.



Convolutional Neural Networks (CNNs) - Example

We will use two pre-trained models from dlib to detect:

- Faces

  link to the pre-trained model:
https://github.com/davisking/dlib-models/blob/master/mmod_human_face_detector.dat.bz2

- Vehicles

  link to the pre-trained model:
https://github.com/davisking/dlib-models/blob/master/mmod_rear_end_vehicle_detector.dat.bz2

https://github.com/davisking/dlib-models/tree/master

Convolutional Neural Networks (CNNs) - Example

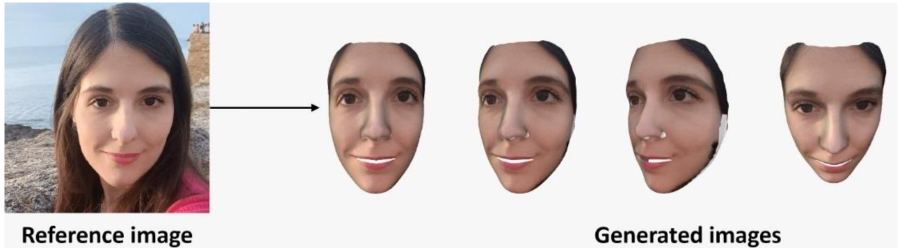
https://bit.ly/471iVlh

https://github.com/davisking/dlib-models/tree/master

Object Alignment

Process of transforming images or parts of images so that they conform to a common coordinate system or reference frame.
It is crucial when comparing images, merging data from different views, or performing tasks like object recognition, feature matching, or 3D reconstruction.

Face Alignment



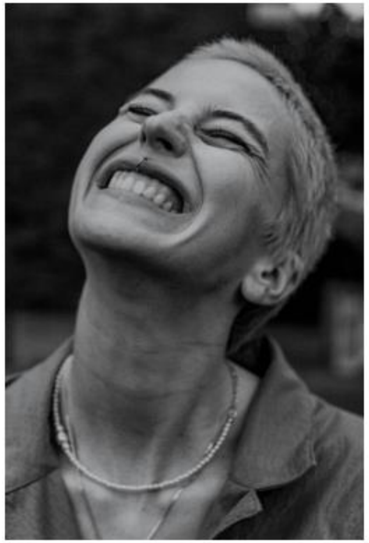
3D Reconstruction

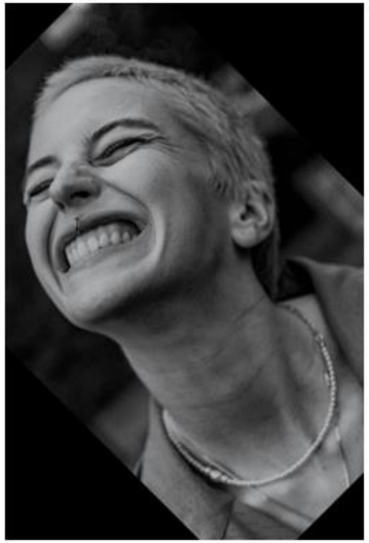
Augmented Reality (AR)

Key Concepts in Object Alignment

Geometric Transformation

Transformations such as rotation, scaling, translation, or reflection to the object in the image. These transformations adjust the object so that its orientation and scale are consistent across different images.

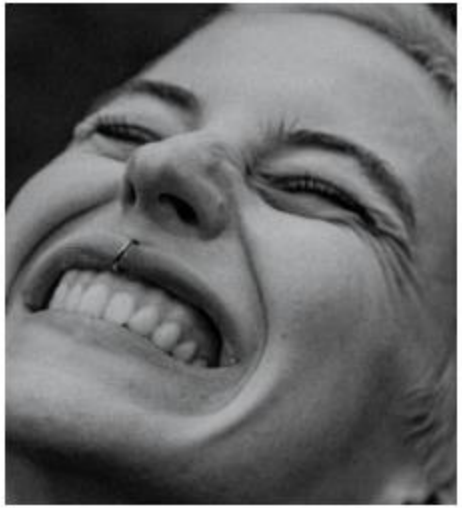
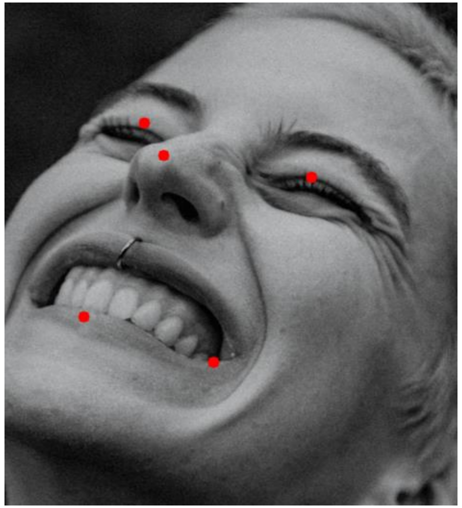
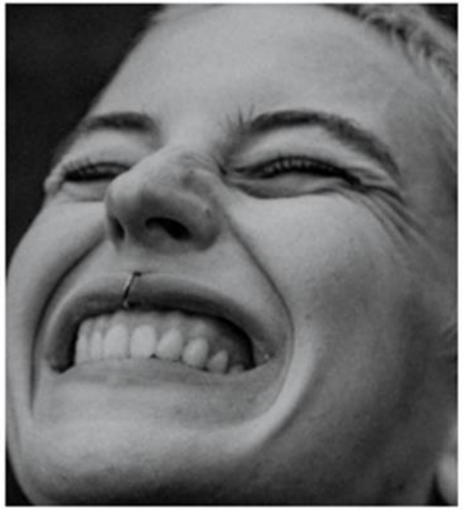




Key Concepts in Object Alignment

Keypoint Detection

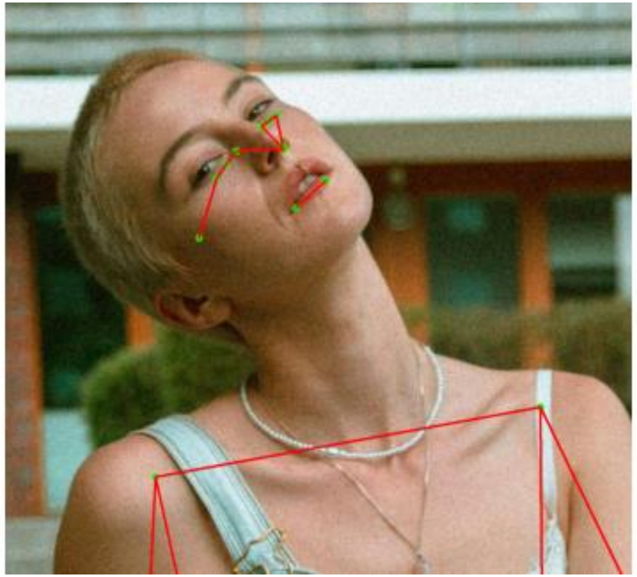
In some alignment methods, specific points or landmarks on the object (e.g., corners of a face, joints of a human body) are identified. These keypoints can then be used to align the object by ensuring that the keypoints match a predefined template.

Key Concepts in Object Alignment

Pose Estimation

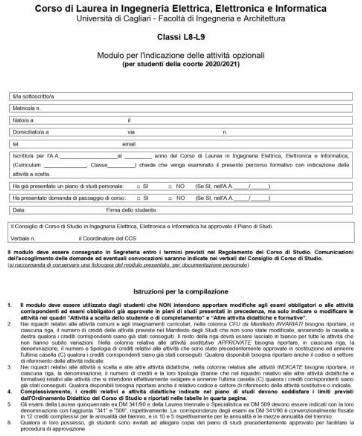
Determining the position and orientation (pose) of an object in the image relative to the camera. Pose estimation can be used to align objects based on their 3D orientation in space.



Key Concepts in Object Alignment

Affine and Homography Transformations

Mathematical transformations used to align objects. They can be used to map points between two planes, allowing for more complex alignment, such as perspective correction.



Corso di Laurea in Ingegneria Elettrica, Elettronica e Informatica
Università di Cagliari - Facoltà di Ingegneria e Architettura

Classi L8-L9

Modulo per l'indicazione delle attività opzionali
(per studenti della coorte 2020/2021)

Il/la sottoscritto/a		
Matricola n.		
Nato/a a	il	
Domiciliato/a a	via	n.
tel.	email	

Iscritto/a per l'A.A.________________al ________ anno del Corso di Laurea in Ingegneria Elettrica, Elettronica e Informatica, (Curriculum ______________, Classe________) chiede che venga esaminato il presente percorso formativo con indicazione delle attività a scelta.

Ha già presentato un piano di studi personale:	□ SI	□ NO	(Se SI, nell'A.A.____/_____)
Ha presentato domanda di passaggio di corso:	□ SI	□ NO	(Se SI, nell'A.A.____/_____)
Data:	Firma dello studente:		

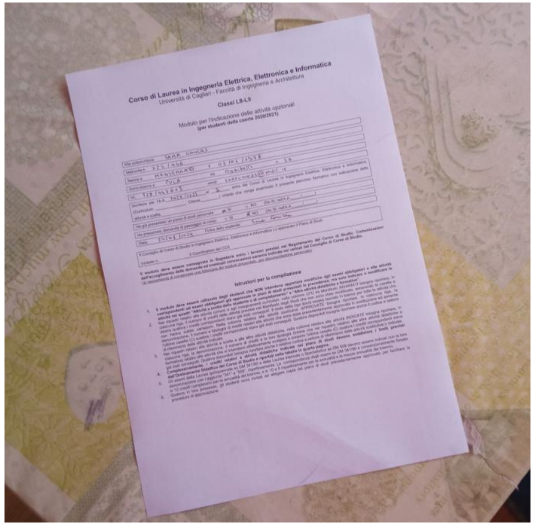
Il Consiglio di Corso di Studio in Ingegneria Elettrica, Elettronica e Informatica ha approvato il Piano di Studi

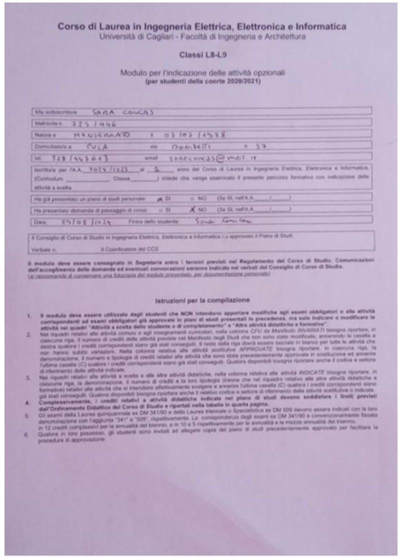
Verbale n. il Coordinatore del CCS

Il modulo deve essere consegnato in Segreteria entro i termini previsti nel Regolamento del Corso di Studio. Comunicazioni dell'accoglimento delle domande ed eventuali convocazioni saranno indicate nei verbali del Consiglio di Corso di Studio.
(*si raccomanda di conservare una fotocopia del modulo presentato, per documentazione personale*)

Istruzioni per la compilazione

1. **Il modulo deve essere utilizzato dagli studenti che NON intendono apportare modifiche agli esami obbligatori o alle attività corrispondenti ad esami obbligatori già approvate in piani di studi presentati in precedenza, ma solo indicare o modificare le attività nei quadri "Attività a scelta dello studente o di completamento" e "Altre attività didattiche e formative".**
2. Nei riquadri relativi alle attività comuni e agli insegnamenti curricolari, nella colonna *CFU da Manifesto INVARIATI* bisogna riportare, in ciascuna riga, il numero di crediti delle attività previste nel Manifesto degli Studi che non sono state modificate, annerendo la casella a destra qualora i crediti corrispondenti siano già stati conseguiti. Il resto della riga dovrà essere lasciato in bianco per tutte le attività che non hanno subito variazioni. Nella colonna relativa alle *attività sostitutive APPROVATE* bisogna riportare, in ciascuna riga, la denominazione, il numero e tipologia di crediti relativi alle attività che sono state precedentemente approvate in sostituzione ed annerire l'ultima casella (C) qualora i crediti corrispondenti siano già stati conseguiti. Qualora disponibili bisogna riportare anche il codice e settore di riferimento delle attività indicate.
3. Nei riquadri relativi alle attività a scelta e alle altre attività didattiche, nella colonna relativa alle *attività INDICATE* bisogna riportare, in ciascuna riga, la denominazione, il numero di crediti e la loro tipologia (tranne che nel riquadro relativo alle altre attività didattiche e formative) relativi alle attività che si intendono effettivamente svolgere e annerire l'ultima casella (C) qualora i crediti corrispondenti siano già stati conseguiti. Qualora disponibili bisogna riportare anche il relativo codice e settore di riferimento delle attività sostitutive o indicate.
4. **Complessivamente, i crediti relativi a attività didattiche indicate nel piano di studi devono soddisfare i limiti previsti dall'Ordinamento Didattico del Corso di Studio e riportati nelle tabelle in quarta pagina.**
5. Gli esami della Laurea quinquennale ex DM 341/90 e della Laurea triennale o Specialistica ex DM 509 devono essere indicati con la loro denominazione con l'aggiunta "341" e "509", rispettivamente. La corrispondenza degli esami ex DM 341/90 è convenzionalmente fissata in 12 crediti complessivi per le annualità del biennio, e in 10 e 5 rispettivamente per le annualità e le mezze annualità del triennio.
6. Qualora in loro possesso, gli studenti sono invitati ad allegare copia del piano di studi precedentemente approvato per facilitare la procedura di approvazione.

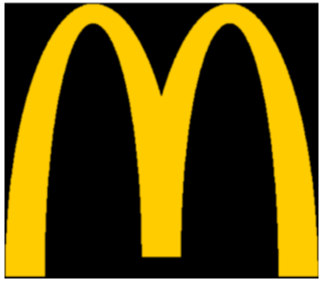


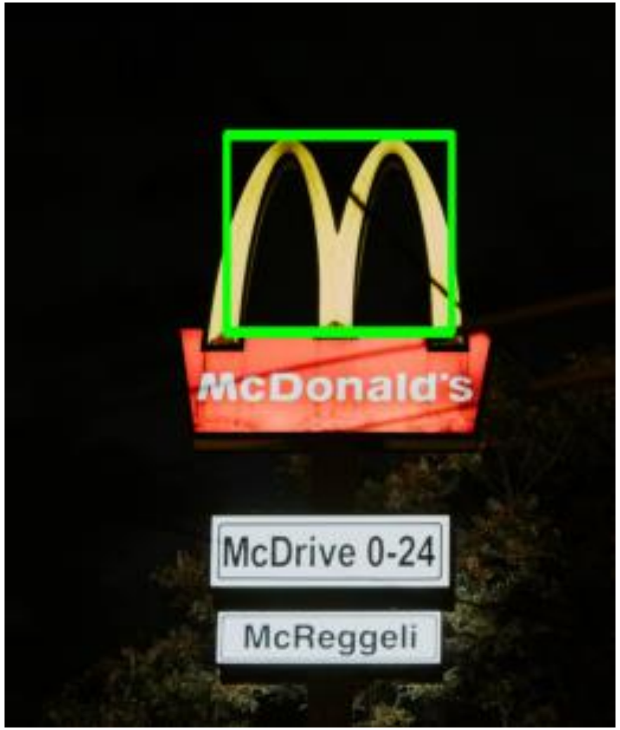


Key Concepts in Object Alignment

Template Matching

A predefined template (such as an image or a set of features) can be used to align objects in new images. The object in the image is transformed until it best matches the template.
This can be useful to detect objects corresponding to the template, inside an image.



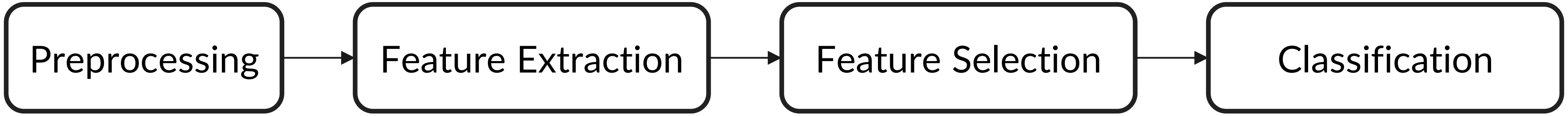


Object Alignment - Example

https://bit.ly/4cFlMRT

Object classification - Handcrafted Features

Manually designed attributes or characteristics of images that are used to distinguish between different classes.
They are specific patterns, textures, shapes, or other visual elements that humans identify as important for recognizing objects in an image.



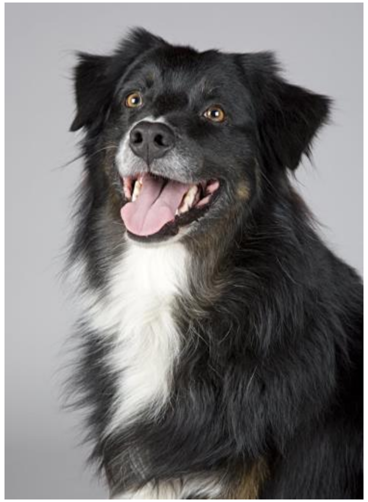
Object classification - Handcrafted Features

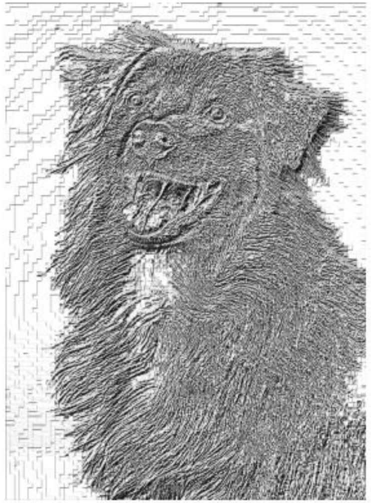
Local Binary Pattern (LBP)

Texture descriptor used in computer vision and image processing. It is a simple yet powerful method for texture classification. The basic idea behind LBP is to summarize the local structure of an image by comparing each pixel with its surrounding neighbors.

Extracts certain uniform patterns corresponding to micro-features in the image.

Looks at points surrounding a central point and tests whether the surrounding points are greater than or less than the central point





Object classification - Handcrafted Features

Local Binary Pattern (LBP)

1. Select a pixel in the image and find its neighboring pixels

30	40	10	0	50
70	50	20	20	65
30	0	30	40	15
20	10	20	70	90
5	50	50	10	20

$\longrightarrow$

50	20	20
0	30	40
10	20	70

2. Take the intensity of the selected pixel as a threshold (30 in this case)

Object classification - Handcrafted Features

Local Binary Pattern (LBP)

3. Go through every neighboring pixel and check whether its intensity is greater than or less than the threshold.
    - Assign 1 to the neighboring pixel, if the intensity of the neighboring pixel is greater than the threshold.
    - Assign 0 to the neighboring pixel, if the intensity of the neighboring pixel is less than the threshold.

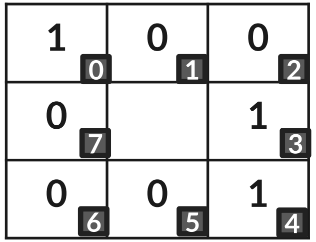
50	20	20
0	30	40
10	20	70

1	0	0
0		1
0	0	1

Object classification - Handcrafted Features

Local Binary Pattern (LBP)

4. Combine the binary values for all neighboring pixels to obtain a binary code for the central pixel, and convert it to a decimal value.



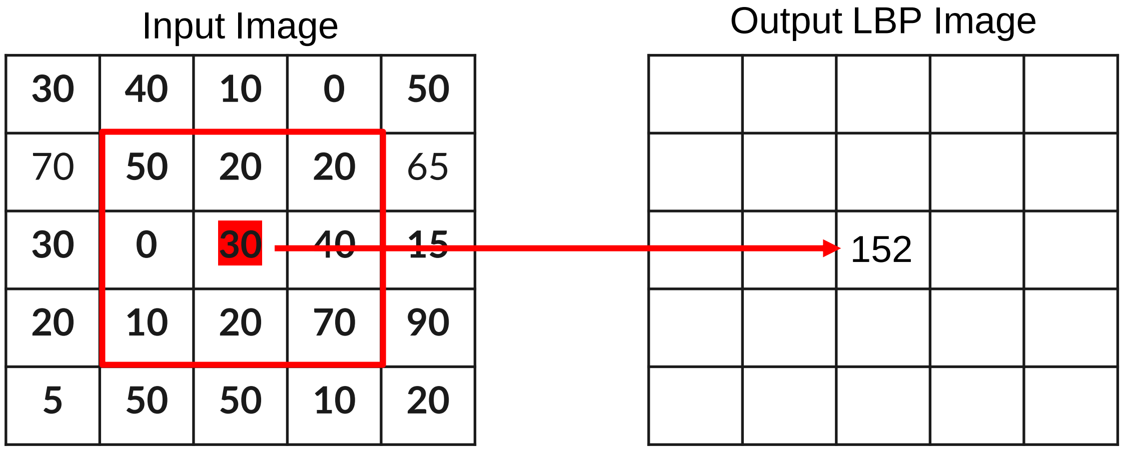
1	0	0	1	1	0	0	0
2^7			2^4	2^3			

$128 \quad + \quad 16 + 8$ = 152

Object classification - Handcrafted Features

Local Binary Pattern (LBP)

5. The calculated LBP value is stored in an output array with the same width and height as the original image.



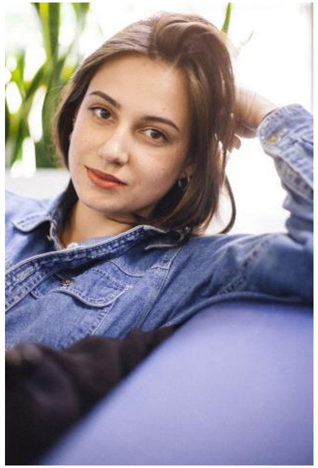
30	40	10	0	50
70	50	20	20	65
30	0	30	40	15
20	10	20	70	90
5	50	50	10	20

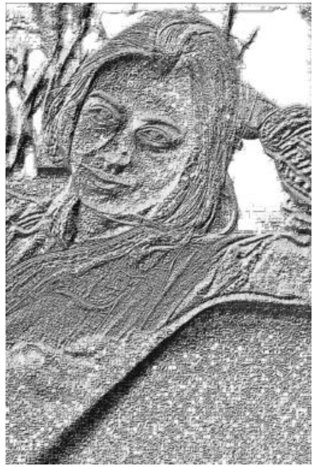
6. The steps 1-5 are repeated for each pixel in the image.

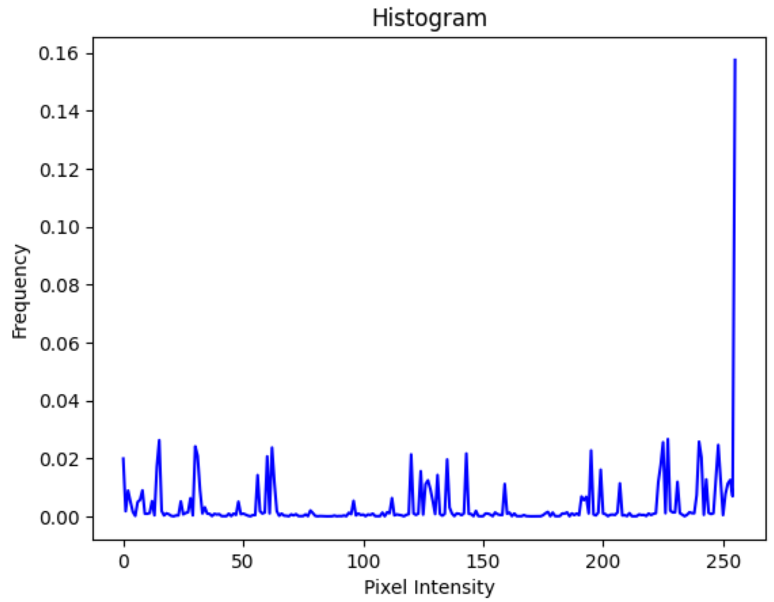
Object classification - Handcrafted Features

Local Binary Pattern (LBP)

7. Finally, we compute the histogram over the LBP image.

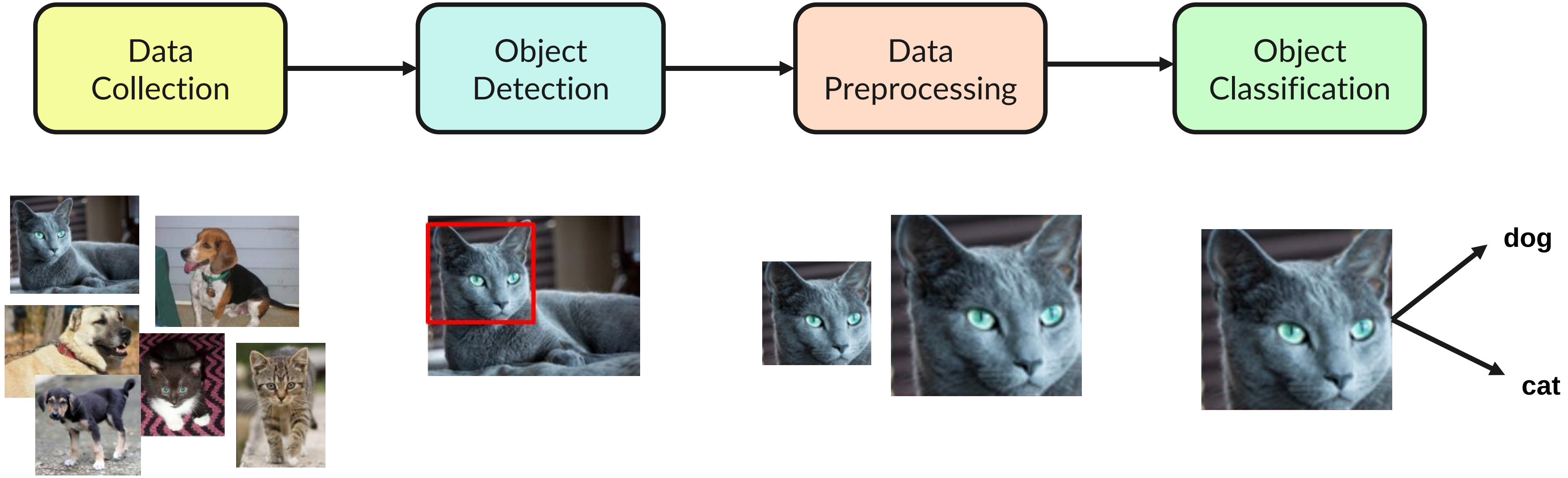






Object Detection and Classification

The complete workflow



Object classification - Handcrafted Features

Local Binary Pattern (LBP) - Example

https://bit.ly/3z5xE1V

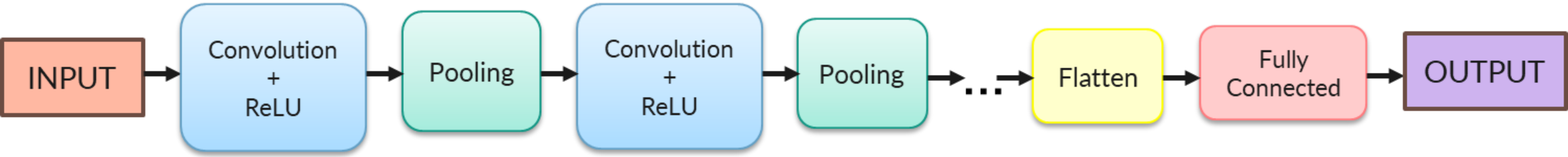
Object classification - Handcrafted Features

Other popular methods

- SIFT (Scale-Invariant Feature Transform):
  https://link.springer.com/article/10.1023/B:VISI.0000029664.99615.94
- LBP (Local Binary Patterns):
  https://ieeexplore.ieee.org/document/1017623
- Color Histograms:
  https://ieeexplore.ieee.org/document/139558
- Gabor Filters:
  http://www.granularsynthesis.com/pdf/gabor.pdf
- Wavelet Transforms:
  https://ieeexplore.ieee.org/document/192463

Object classification - CNNs

CNNs automatically extract hierarchical features that are more complex and suited to the task, outperforming traditional handcrafted methods in many scenarios.



Object classification – CNNs Advantages

- **Automatic Feature Extraction**: CNNs automatically learn relevant features from the data, eliminating the need for manual feature engineering.
- **Spatial Hierarchies**: CNNs are designed to capture spatial hierarchies in images. Lower layers capture low-level features like edges, while deeper layers capture high-level features like objects or parts of objects.
- **Translation Invariance**: Through convolution and pooling, CNNs are naturally invariant to small translations, making them robust to variations in the input image.
- **Scalability**: CNNs can be scaled to work with large datasets and complex images

Object classification – CNN Example

https://bit.ly/4dIEr0I

Object Detection and Classification

Exercise

TASK: Modify the example below (seen in lecture 2) adding adding an appropriate object detection step.

https://bit.ly/4fQkkiF