Computer Vision

Unit 4

Object Recognition

Object Detection. Face - Detection - Pedestrian detection; Face recognition: Eigen faces - active appearance and 3D shape models; Instance recognition: geometric alignment; category ~~alignment~~ recognition - Bag of words part-based models; Context and scence understanding: Learning & large image collection; Applications for object recognition

Face Detection – Viola Jones Algorithm

→ a ML technique used for object detection, primarily for faces.

Given a grayscale image, the algorithm analyzes many windows of different sizes & positions to detect the target object.

→ It has 4 main components:

(i) Haar - like Features

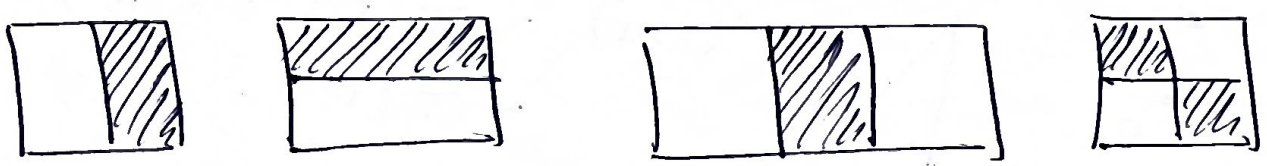
(ii) Computing Integral Image

(iii) AdaBoost for Feature Selection

(iv) Cascade Classifiers

A: **Haar - Like Features**

→ The algorithm uses the following Haar-like features:



→ The regularities of the human face is matched with these Haar Features

→ The feature value is computed as the difference between the sum of the pixel values in the white areas & the sum of the pixel values in the black areas.

B. **Integral Image**

→ The integral image is a data structure for efficiently computing the sum of the pixel values in a rectangular image window.

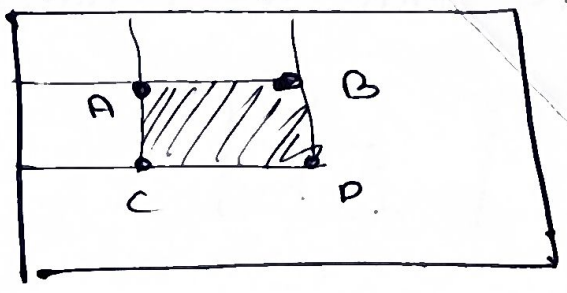
→ The Haar-like features can be quickly computed with the integral image representation

→ For an image I, the integral image value $I'(x,y)$ is

$$I'(x,y) = \sum_{\substack{x' \le x \\ y' \le y}} I(x,y)$$

i.e sum of all pixels above and to the left of (x,y) inclusive

eg.



Inside the rectangle:

$$I'(x,y) = I(D) + I(A) - I(B) - I(C)$$

C. AdaBoost Classifier

→ The number of features from the Haar features is very large. Only a few of them are useful for face detection.

→ AdaBoost finds the best features. Each Haar-like feature is a weak classifier.

→ The final classifier is a linear combination of weak classifiers

→ Larger weights are associated w/ better classifiers using AdaBoost

D. Cascade Classifiers

→ The cascade system divides the process of identifying a face into multiple stages.

→ The first stage has a classifier that makes up the best features, say eyes / nose.

→ An image subregion enters the cascade is evaluated by the first stage. If that stage marks the subregion as +ve, it is sent to the next stage of the cascade.

→ Cascades quickly discard non-face images.

* Pedestrian Detection - Background Subtraction + Contour Extraction

Background - A background model is created by capturing a background image without any pedestrians present

- An adaptive model can be built using GMMs, as parameter are updated to adapt to changes in illumination or scene dynamics.

Subtraction - Analyze the differences between consecutive frames in a video to detect moving objects which are likely to be pedestrians

→ It identifies the pixels that have changed significantly, indicating a moving object.

Contour Extraction

→ Once moving objects are identified, contour extraction methods are used like:

(i) LoG

(ii) Gradient-based methods - Sobel

Pedestrian Detection

→ Bounding boxes for pedestrians can be refined using active contour models or segmentation techniques like GrabCut or watershed algorithms.

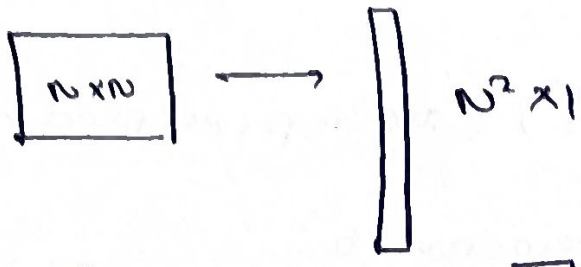
* Face Recognition - Eigen faces

→ The Eigenfaces algorithm uses the PCA to detect faces. It uses eigen values and eigen vectors to reduce dimensionality and project data onto a smaller feature space.

Algorithm - Training

1. Consider a set of m images of dimension N×N.

2. Convert these images into vectors of size N^2 such that



$$V = [x_1, x_2 \dots x_m]$$

3. Calculate the average of these face vectors and subtract it from each vector:

$$\psi = \frac{1}{m} \sum_{i=1}^{m} x_i$$

$$a_i = x_i - \psi$$

4. Take all the face vectors to get a matrix of size $N^2 \times M$

$$A = [a_1 \; a_2 \dots a_m]$$

5. Compute the covariance matrix

$$Cov = A^T A$$

6. Compute the eigen values & eigen vectors

$$A^T A v_i = \lambda_i v_i$$

$$A^T A A v_i = \lambda A v_i$$

$$C' u_i = \lambda_i u_i$$

$$C' = A A^T$$

$$u_i = A v_i$$

7. Select k eigen vectors of C' corresponding to the largest values.

8. The normalized faces $(x_i - \psi)$ are represented as a linear combination of the best k eigenvectors

$$x_i - \psi = \sum_{j=1}^{k} w_j u_j$$

u_j ⟶ eigen faces

make a coeff. vector

Testing

→ Preprocess to make the dimensions the same

→ Subtract from the average face.

$$\phi = y - \psi$$

→ Project the normalized vector into the eigenspace

$$\phi = \sum_{i=1}^{k} w_i u_i$$

generate a coeff. vector $\Omega = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$

→ Compute the distance between the training & testing coeff. vectors

→ The image w/ the minimum distance is the matching one

* Face Detection - Active Appearance Model

→ The AAM is a statistical model to detect faces using shape & appearance information.

↓

or textures

Shape Model Representation

→ A shape model S is represented as a vector of landmark coordinates

$S = [(x_1, y_1), (x_2, y_2) \ldots (x_n, y_n)]$ where n is the number of landmarks

→ PCA is used to create a compact representation of the features using eigenvectors.

→ The best eigenvectors that capture the most significant shape variations, forms the shape basis matrix B_s.

Appearance Model Representation

→ An appearance model A captures pixel intensity variations around each landmark.

→ Each landmark has an appearance vector $A = [a_1, a_2 \ldots a_n]$

→ PCA is applied on the appearance vectors to generate an appearance basis matrix B_a.

Model Formulation

→ The model is formulated as:

$$M = \bar{S} + B_s \cdot \omega_s + B_a \cdot \omega_a$$

$\bar{S}$ = mean shape

ω_s = shape coeff

ω_a = appearance coeff.

→ This model is trained w/ a dataset of annotated images

Model Fitting

→ AAM parameters are adjusted to align the model w/ a new i/p image

→ Algorithms like ASM actively update model parameters by minimizing an energy function.

* Instance Recognition – Geometric Alignment

Alignment – find parameters of model that maps one set of parameters to another.

→ In order to find the best matching alignment, transformations have to be done. These include: (i) Translation (iv) Affine
(ii) Rotation (v) Perspective
(iii) Aspect

Affine Transformations

→ They are combinations of linear transformations and translations

→ The major properties are:

(i) Lines map to lines

(ii) Parallel lines remain parallel

(iii) Ratios are preserved

(iv) closed under composition

→ They are represented as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Instance Recognition Algorithm

The basic steps are: (i) Match keypoints to object model

(ii) Solve for affine transformation parameters

(iii) Score the inliers are choose those that score above a threshold

Key point Matching

1. Identify distinct keypoints.
2. Define a region around each key point
3. Extract and normalize the region cont

4. Compute a local descriptor from the normalized region

5. Match the localized descriptors

Matching Objects

1. Match multiple keypoints from input image & database image.

2. Find images that have matching position / orientation / scales for a minimum threshold.

* Category Recognition - Bag of Visual Words

→ Bag of Visual Words is a technique to compute similarities between images and is used for image classification.

→ This approach is based on the NLP BoW approach

→ In the BoW approach, the entire document is scanned, and a count of each word appearing in the document is maintained.

→ Then, a histogram of frequencies is created and used to describe the text document.

→ In Bag of Visual Words, the input is an image instead of a text document.

Step 1: Feature Extraction

→ Breakdown image into a set of independent features.

→ The features consist of keypoints and descriptors

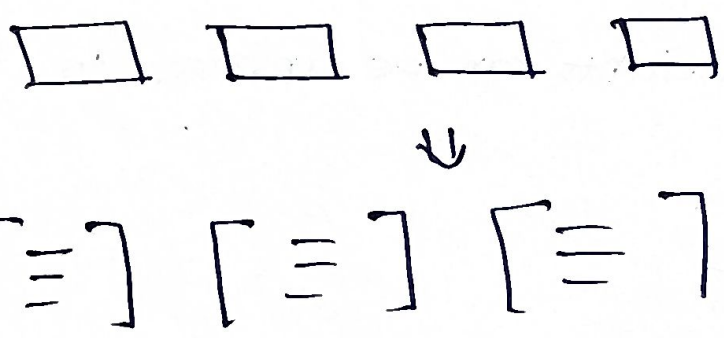
Keypoints - refer to RoI, spatial locations

- They are invariant to rotation, shrinkage, translation, distortion.

Descriptors - are the values corresponding to those keypoints.

Step2: Creation of a Codebook

→ Use feature extractors like SIFT, BRISK etc. They return the array containing the descriptors. This is done for every image in the training dataset

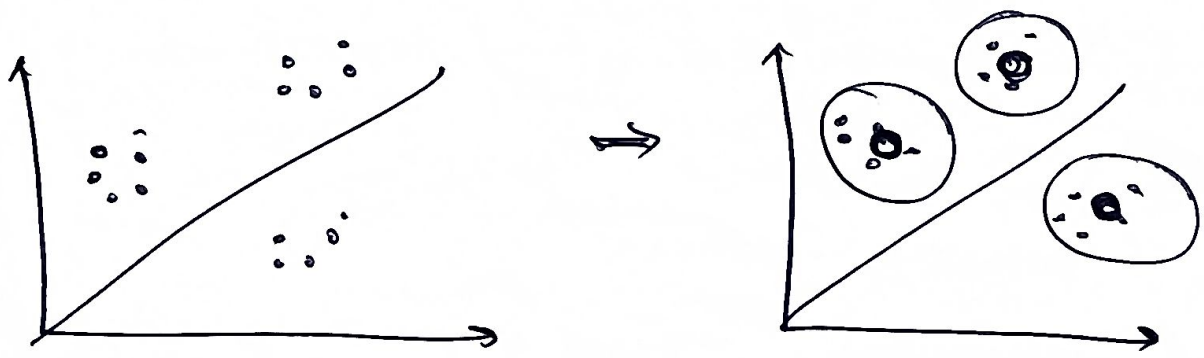


→ Stack these arrays vertically.

Step3: Clustering

→ Use a clustering algorithm like K-Means to form K clusters.

→ Each cluster center (centroid) acts as a visual word. The K centroids form the codebook.



Step 4 : Creation of Histograms

→ Iterate through the images and look for words present in the dictionary.

→ Once a word is detected that is in both the dictionary & image, increase the count of that word.

→ Then create histograms for each image

→ All the images are converted to histogram - Train a classifier (RF, LinearSVC) for classifying the images.

→ For improved performance, weights can be applied to features that occur in many images

* Part-based Models

→ The problem with the bag of words approach is that all words have an equal probability.

→ The location of features is important

→ Part-based models, model the relative locations between parts

→ The correspondence problem

model with P parts

Image with N possible locations

⇒ N^P combinations → use a sparse representation using maximal cliques

Learning Procedure

→ Find regions, their location and appearance

→ Initialize model parameters

→ Use the EM model and iterate till convergence

E → compute assignments for which regions belong to which parts

M → update model parameters

* Context and Scene Understanding

Context Understanding

Spatial Context

→ identify relative positions and spatial relationships between objects

→ Techniques - SPP, R-CNN

→ eg. recognizing that a mouse is typically found near a keyboard using spatial relationships improves object detection accuracy.

Semantic Context → co-occurrence of objects.

→ Techniques - semantic segmentation, co-occurrence matrices

Temporal Context → considers the continuity & changes in objects over time in video sequences.

Techniques - RNN, LSTM