Feature Engineering techniques for text, images, audio, and video.

Features extraction techniques for images

- Histogram of gradients (HOG)
- Maximally stable extremal regions (MSER)
- Scale-invariant feature transform (SIFT)
- Speeded Up Robust Features (SURF)
- Features from accelerated segment test (FAST)
- Local binary pattern (LBP)
- Local phase Quantization (LPQ)
- Edge detection techniques (Canny etc.)

1. Histogram of Oriented Gradients (HOG)

HOG is a feature descriptor that captures gradient orientation and magnitude distributions in localized image regions. It is widely used in object detection, especially pedestrian detection.

How it works:

1. **Gradient Computation:** Compute horizontal (G_x) and vertical (G_y) gradients using Sobel operators.
2. **Gradient Orientation and Magnitude:** Calculate the gradient magnitude and direction:

$$\text{Magnitude} = \sqrt{G_x^2 + G_y^2}$$

$$\text{Orientation} = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

3. **Spatial Binning:** Divide the image into small cells (e.g., 8×8).
4. **Histogram Creation:** Form histograms of gradient orientations within each cell.
5. **Block Normalization:** Normalize histograms over overlapping blocks to improve invariance to illumination.
6. **Feature Vector Formation:** Concatenate histograms into a single feature vector.

Example Application:

- Used in pedestrian detection in self-driving cars.

2. Maximally Stable Extremal Regions (MSER)

MSER is a region-based feature detector that identifies stable regions under varying illumination and transformations.

How it works:

1. Convert the image to grayscale.
2. Identify connected components at different intensity thresholds.
3. Select regions that remain stable across multiple thresholds.
4. Store these as MSER regions.

Example Application:

- Used in text detection for Optical Character Recognition (OCR).

3. Scale-Invariant Feature Transform (SIFT)

SIFT detects keypoints and describes them using a robust descriptor, making it scale and rotation-invariant.

How it works:

1. **Scale-space Representation:** Apply Gaussian filters at multiple scales.
2. **Keypoint Detection:** Find extrema in the Difference of Gaussian (DoG) space.
3. **Keypoint Localization:** Use Hessian matrix to refine keypoints.
4. **Orientation Assignment:** Assign dominant gradient orientations to make descriptors rotation-invariant.
5. **Feature Descriptor Computation:** Extract histograms of gradients in local patches around keypoints.

Example Application:

- Used in image stitching for panorama creation.

4. Speeded-Up Robust Features (SURF)

SURF is a faster alternative to SIFT and uses integral images for rapid computation.

How it works:

1. **Scale-space Representation:** Uses Hessian matrix for fast keypoint detection.
2. **Orientation Assignment:** Uses Haar wavelet responses.
3. **Descriptor Generation:** Uses wavelet responses to create a 64 or 128-dimensional feature vector.

Example Application:

- Used in object recognition and tracking in real-time applications.

5. Features from Accelerated Segment Test (FAST)

FAST is a high-speed corner detection algorithm.

How it works:

1. Consider a circular region of 16 pixels around a candidate pixel.
2. If a contiguous set of pixels (e.g., 12 out of 16) are brighter or darker than the central pixel, mark it as a corner.
3. Apply non-maximum suppression to remove weak corners.

Example Application:

- Used in real-time applications like mobile augmented reality.

6. **Local Binary Patterns (LBP)**

LBP is a texture descriptor that encodes pixel neighborhood relationships.

How it works:

1. Divide the image into small regions.
2. For each pixel, compare its intensity with its 8-neighboring pixels.
3. Assign a binary value (1 if the neighbor is greater, 0 otherwise).
4. Convert the binary pattern into a decimal value.
5. Construct a histogram of LBP values.

Example Application:

- Used in face recognition systems.

7. Local Phase Quantization (LPQ)

LPQ is a blur-invariant texture descriptor.

How it works:

1. Apply a Short-Time Fourier Transform (STFT) to small image regions.
2. Quantize the phase information into a binary code.
3. Create a histogram of phase-coded values.

Example Application:

- Used in blur-robust biometric recognition.

8. Canny Edge Detector

Canny is an edge detection technique that finds strong edges in an image.

How it works:

1. **Noise Reduction:** Apply Gaussian smoothing.
2. **Gradient Computation:** Compute Sobel gradients.
3. **Non-Maximum Suppression:** Thin edges by keeping only local maxima.
4. **Hysteresis Thresholding:** Retain edges using high and low thresholds.

Example Application:

- Used in medical image segmentation.

Summary Table

Method	Type	Key Feature	Application
HOG	Feature Descriptor	Gradient orientation histograms	Pedestrian detection
MSER	Feature Detector	Stable connected regions	OCR text detection
SIFT	Feature Detector & Descriptor	Scale and rotation invariance	Image stitching
SURF	Feature Detector & Descriptor	Fast alternative to SIFT	Object recognition
FAST	Feature Detector	Rapid corner detection	Augmented reality
LBP	Texture Descriptor	Binary texture encoding	Face recognition
LPQ	Texture Descriptor	Blur-invariant texture features	Biometric recognition
Canny	Edge Detector	Multi-stage edge detection	Medical imaging

Textual Features

- Number of words
- Frequency
- Parts of speech
- Paragraph
- Sentences

1. **Number of Words**

The number of words in a document helps analyze text length, readability, and complexity.

How it Works:

1. **Tokenize the text:** Split the text into words.
2. **Count the total words**, including or excluding stop words (common words like "the," "is," "and").

Example:

Text:

"Natural Language Processing is a branch of AI."

- **Word Count (including stop words):** 7
- **Word Count (excluding stop words like "is", "a", "of"):** 5

Applications:

- Used in **readability analysis** (e.g., Flesch-Kincaid readability score).
- Helps in **spam detection** (e.g., extremely short messages might be spam).

2. Word Frequency

Word frequency measures how often a word appears in a document or corpus. It is useful for text mining, keyword extraction, and sentiment analysis.

How it Works:

1. Tokenize the text into words.
2. Count the occurrences of each word.
3. Normalize (optional): Convert counts into percentages.

Example:

Text:
"Data science is fun. Data science involves statistics."

- Word Frequencies:
  - "data" → 2
  - "science" → 2
  - "is" → 1
  - "fun" → 1
  - "involves" → 1
  - "statistics" → 1

Applications:

- Used in **search engines** (higher frequency words help in ranking).
- Helps in **sentiment analysis** (e.g., frequency of "happy" vs. "sad").

4. Paragraph Detection

Paragraph detection involves segmenting text into meaningful sections. A paragraph is a collection of related sentences.

How it Works:

1. Split the text based on line breaks (`\n\n`) or indentation.
2. Count the number of paragraphs.

Example:

Text:

```pgsql
AI is transforming industries. It improves efficiency and reduces costs.

Machine learning, a subset of AI, enables computers to learn from data.
```

- Number of Paragraphs: 2

Applications:

- Used in document summarization (each paragraph may discuss a separate topic).
- Helps in document structuring (e.g., AI-based formatting).

5. Sentence Detection

Sentence detection breaks text into individual sentences for grammatical analysis.

How it Works:

1. **Split text using punctuation (`.` `!` `?`) and spacing.**
2. **Identify sentence boundaries carefully (e.g., "Dr. Smith is here." is one sentence, not tw**

Example:

Text:
"NLP is amazing! It helps machines understand language. What do you think?"

- Sentences:
  1. NLP is amazing!
  2. It helps machines understand language.
  3. What do you think?
- Sentence Count: 3

Applications:

- Used in **chatbots and voice assistants** (breaking responses into sentences).
- Helps in **grammar checking** (e.g., Grammarly detects run-on sentences).

Features extraction techniques for textual data

Bag of words

Term frequency-inverse document frequency

Word Embeddings

Feature Extraction Techniques for audio data

Features extraction techniques for audio data

- Mel frequency cepstral coefficients (MFCCs)
- Linear Prediction Coefficient (LPC)
- Linear Prediction Cepstral Coefficients (LPCC)
- Line Spectral Frequencies (LSF)
- Discrete Wavelet Transform (DWT)
- Perceptual Linear Prediction (PLP)