PUNE INSTITUTE OF COMPUTER TECHNOLOGY

Department of Computer Engineering

(2023-2024)

LP-6

Batch:- R2

Feature Extraction using Zernike Moments

Group members:

41251-Aditya Magdum

Guided by: Prof. Nikita Kapadnis

Table of Contents: -

1. Problem Statement:

To extract features to capture the shape information of objects in an image using Zernike Moments(ZMs).

2. Motivation:

The motivation for this mini project lies in the need for effective feature extraction techniques in image processing and computer vision tasks. As image datasets grow in size and complexity across diverse domains, there is a rising demand for methods that can efficiently extract meaningful information from images for analysis and decision-making. Zernike Moments offer a promising solution due to their ability to succinctly capture shape information in a manner that is invariant to rotation, scale, and translation. By exploring the application of Zernike Moments for feature extraction, this project aims to address key challenges such as feature representation, invariance, dimensionality reduction, and interpretability, thereby contributing to advancements in image processing and computer vision with potential applications in fields such as medical imaging, object detection, and content-based image retrieval.

3. Problem Scope:

Understanding the scope of utilizing Zernike Moments for feature extraction is pivotal for enhancing image processing tasks. This study aims to explore the applicability of Zernike Moments across different image types and application domains, assessing their effectiveness compared to alternative feature extraction methods. Additionally, it investigates the sensitivity of Zernike Moments to parameters such as order and resolution, while also evaluating their computational efficiency. By comprehensively analyzing these aspects, we can gain insights into the suitability of Zernike Moments for various image processing applications and guide the development of more accurate and efficient algorithms.

4. Objectives:

The objective of this mini project is to demonstrate the use of Zernike Moments for feature extraction in image processing tasks. Specifically, we aim to extract Zernike Moments from a set of images and use them as features for pattern recognition and classification.

5. Outcomes:

The expected outcomes of this report are:

- Zernike Moments offer enhanced feature representation for shape information in digital images, improving object recognition and pattern analysis accuracy.
- Their versatility is demonstrated across diverse application domains, including medical imaging, object recognition, and industrial inspection.
- Sensitivity analysis of Zernike Moments parameters enables optimization for specific scenarios, enhancing performance in different applications.
- Evaluation of computational efficiency highlights their potential for scalable and real-time image processing tasks, facilitating practical deployment in various domains.

6. Hardware and Software Requirements

a. Hardware Requirements

- CPU
- Windows 11, 64bit

- GPU (Nvidia) (Optional)
- 8GB RAM

b. Software Requirements
- Python 3.0
- Jupyter Notebook IDE
- Libraries and Packages
- Images

7. Theory:

Zernike Moments (ZMs) are a set of orthogonal moments used for shape analysis and pattern recognition in digital image processing. They are named after Frits Zernike, who introduced them in the context of optical aberration theory. Zernike Moments are particularly useful for capturing shape information from binary or grayscale images.

Computing Zernike Moments:

Given an image, Zernike Moments are computed by projecting the image onto the set of Zernike polynomials. This process involves integrating the product of the image intensity and the Zernike polynomials over the image domain. The resulting set of moments represents the shape information of the object in the image.

Applications:

Zernike Moments have various applications in image processing and computer vision, including:

1. Object recognition and classification.

2. Shape analysis and matching.

3. Object tracking and localization.

4. Biomedical image analysis, such as cell and tissue analysis.

5. Optical character recognition (OCR).

6. Face recognition and expression analysis.

Advantages:

- Zernike Moments are invariant to rotation, translation, and scaling, making them robust to changes in object orientation and size.

- They provide a compact representation of shape features, reducing the dimensionality of the feature space.

- Zernike Moments are computationally efficient and can be computed using fast algorithms such as the recursive method or the radial recursion method.

Limitations:

- Zernike Moments may not capture fine details or texture information present in the image.

- They are sensitive to noise and may require preprocessing steps such as smoothing or thresholding.

- Zernike Moments may not be suitable for highly complex shapes with irregular contours.

8. Procedure

For the implementation of this mini project, we used Python programming language along with the OpenCV library for image processing and the scikit-image library for Zernike Moments computation. The steps involved in the implementation are as follows:

- Load the dataset of images containing objects of interest.

- Preprocess the images (if necessary) to ensure uniformity in size and orientation.
- Compute Zernike Moments for each image using the skimage.measure.zernike_moments function.
- Use the computed Zernike Moments as features for classification tasks.
- Train a machine learning model (e.g., SVM, KNN) using the Zernike Moments-based features.
- Evaluate the performance of the trained model using cross-validation or train-test split.

9. Code

```
def describe_shapes(image):
    # initialize the list of shape features
    shapeFeatures = []

    # convert the image to grayscale, blur it, and threshold it
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    blurred = cv2.GaussianBlur(gray, (13, 13), 0)
    thresh = cv2.threshold(blurred, 50, 255, cv2.THRESH_BINARY)[1]

    # perform a series of dilations and erosions to close holes
    # in the shapes
    thresh = cv2.dilate(thresh, None, iterations=4)
    thresh = cv2.erode(thresh, None, iterations=2)

    # detect contours in the edge map
    cnts = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL,
        cv2.CHAIN_APPROX_SIMPLE)
    cnts = imutils.grab_contours(cnts)

    # loop over the contours
    for c in cnts:
        # create an empty mask for the contour and draw it
        mask = np.zeros(image.shape[:2], dtype="uint8")
        cv2.drawContours(mask, [c], -1, 255, -1)

        # extract the bounding box ROI from the mask
        (x, y, w, h) = cv2.boundingRect(c)
        roi = mask[y:y + h, x:x + w]
```

```
            # compute Zernike Moments for the ROI and update the list
            # of shape features
            features = mahotas.features.zernike_moments(roi, cv2.minEnclosingCircle(c)[1], degree=8)
            shapeFeatures.append(features)

        # return a tuple of the contours and shapes
        return (cnts, shapeFeatures)
```

10. Conclusion

In conclusion, Zernike Moments provide a robust method for feature extraction in image processing applications. The mini project successfully demonstrated the use of Zernike Moments for extracting shape features from images and utilizing them for classification tasks. Further research and experimentation can explore the application of Zernike Moments in various domains such as medical imaging, object detection, and facial recognition.