Usage Manual for MATLAB Script: Object Detection with Feature Matching

Purpose:

This MATLAB script performs object detection using geometric properties of binary images from the `Kimia99_DB` dataset. It focuses on:

1. Training with a set of binary images.
2. Testing the trained model for object classification.
3. Evaluating the recognition accuracy by comparing image features.

---

1. Requirements:

- MATLAB software with Image Processing Toolbox.
- A dataset directory (`Kimia99_DB`) containing binary images of objects.
- Input images should have white foreground objects on a black background.

---

2. Functional Workflow

2.1. Initialization

- **Code Segment:**

```matlab
clc
clear all
close all
```

Clears the workspace, command window, and closes all figures to ensure a clean environment.

- **Directory and Variable Setup:**

```matlab
listing = dir('Kimia99_DB');
num_rec = zeros(9,11);
```

  - Scans the dataset directory (`Kimia99_DB`) for image files.
  - Initializes `num_rec` to track classification results.

3. Training Phase

- **Main Loop:** Iterates through images for training and extracts geometric features.

```matlab
for train_no = 1:11
    for class_indx = 1:9
```

- **Image Processing:**
  - Reads and resizes each training image:

```matlab
filename = listing(tmp_indx).name;
bw = im2bw(imread(sprintf('Kimia99_DB/%s', filename)), 0);
tmp_bw = imresize(bw, [256 256]);
```

  - Stores the resized image for later processing.

- **Feature Extraction:**
  - Extracts region properties (e.g., centroid, axes, orientation):

```matlab
s = regionprops(bw1{class_indx}, {'Centroid', 'MajorAxisLength', 'MinorAxisLength', 'Orientation', 'Area'});
```

  - Identifies the largest region (ellipse) based on area to avoid noise:

```matlab
[max_area, max_indx] = max(size_ellipse);
```

- **Geometric Vertex Calculation:** Computes the coordinates of major and minor axis endpoints for the largest ellipse:

```matlab
vertex(1,1) = s(max_indx).Centroid(1) - (s(max_indx).MinorAxisLength/2) * sind(s(max_indx).Orientation);
vertex(1,5) = s(max_indx).Centroid(1) + (s(max_indx).MajorAxisLength/2) * cosd(s(max_indx).Orientation);
```

4. Testing Phase

- **Main Loop:** Tests images not used in the training phase.

```matlab
for tst_indx = 1:11
```

- **Image Processing:** Processes test images similarly to training images:

```matlab
bw = im2bw(imread(sprintf('Kimia99_DB/%s', filename)), 0);
tmp_bw = imresize(bw, [256 256]);
```

- **Region Properties:** Extracts features for test images:

```matlab
s = regionprops(bw2{tst_no}, {'Centroid', 'MajorAxisLength', 'MinorAxisLength', 'Orientation', 'Area'});
```

- **Transformation Matrix Calculation:** Aligns test image vertices with training image vertices using a transformation matrix:

```matlab
M = coor_im2 * inv(coor_im1);
```

- **Overlap Rate Calculation:** Projects and compares white pixels of test and training images to calculate the overlap rate:

```matlab
ln_pro = length(PL_pro);
cntr = sum(PL(:,1) == PL_pro(:,1) & PL(:,2) == PL_pro(:,2));
overlap_rate(org_class_indx, tst_no) = cntr / ln_pro;
```

5. Recognition and Evaluation

- **Recognition Index:**

  ```matlab
  rec_indx(class_indx, tst_indx) = max_cls_indx;
  ```

  Assigns the test image to the class with the highest overlap rate.

- **Accuracy Tracking:**

  ```matlab
  num_rec(class_indx, train_no) = num_rec(class_indx, train_no) + 1;
  ```

  Tracks the number of correctly classified images for each class.

6. Visualization

- **Ellipse and Vertex Display:** Plots detected ellipse boundaries and vertices:

  ```matlab
  plot(x, y, 'r', 'LineWidth', 5);
  ```

- **Projected Points:** Displays projected white pixels for comparison:

  ```matlab
  plot(PL_pro(:,1), PL_pro(:,2), 'r.');
  ```

7. Output

- **Overlap Rate Matrix:** Stores overlap rates between test and training images for each class.
- **Recognition Matrix:** Provides classification results for all test images.
- **Elapsed Time:**

  ```matlab
  elapsed_time = toc;
  ```

  Displays the total execution time for training and testing.

8. Practical Applications

- Object detection and classification in binary image datasets.
- Evaluating geometric feature-based recognition systems.

---

9. Notes

- Ensure all images in `Kimia99_DB` are compatible binary files.
- Adjust loop parameters (`train_no`, `class_indx`) based on dataset requirements.
- Visualization steps (`imshow`) can be commented out to improve execution speed.

This manual outlines each script component for efficient use in object recognition tasks.