Experiment No: 9

Name: Khanjan Joshi

Roll number: 26

Batch B

Aim: To Creating and Training an Object Detector

Objective: Bag of Words (BOW) in computer vision - Detecting cars in a scene

Theory:

Creating and Training an Object Detector

Object detection is a crucial task in computer vision, with applications ranging from autonomous driving to surveillance systems. In this experiment, we aim to create and train an object detector specifically for detecting cars in a given scene. One of the techniques we'll use for this purpose is the Bag of Words (BOW) model.

Bag of Words (BOW) in Computer Vision:

The Bag of Words model, borrowed from natural language processing, has been adapted for use in computer vision. It is a popular image representation technique used for object detection and image classification.

The BOW model works as follows:

1. Feature Extraction: Extract local features, such as SIFT (Scale-Invariant Feature Transform) or ORB (Oriented FAST and Rotated BRIEF), from a set of training images containing the object of interest (in this case, cars).

2. Create a Vocabulary: Cluster the extracted features into a vocabulary of visual words using clustering techniques like K-means. These visual words represent common patterns or features found in the training images.

3. Histogram Representation: For each image, create a histogram that counts the occurrences of visual words in the image. This histogram is known as the Bag of Words representation.

4. Train a Classifier: Train a machine learning classifier (e.g., SVM or Random Forest) on the Bag of Words representations of the training images, labeling them as positive (contains the object) or negative (does not contain the object).

5. Object Detection: Apply the trained classifier to new, unseen images to detect the object of interest.

Detecting Cars:

In our experiment, we will focus on detecting cars in a scene using the BOW model. We'll use a dataset of images containing cars and background scenes for training.

Example:

Let's take a look at a simplified example of how this process might work in Python:

Code:

```
import cv2 import
numpy as np

# Load training images containing cars
car_images = [cv2.imread(car) for car in os.listdir("car_dataset")]
```

```
# Extract features (e.g., SIFT) from car images

car_features = [extract_features(image) for image in car_images]

# Cluster features to create a vocabulary
vocabulary = create_vocabulary(car_features)

# Create Bag of Words representations for training images
training_data = create_bow_representation(car_images, vocabulary)

# Train a classifier (e.g., SVM)
classifier = train_classifier(training_data, labels)

# Load a test image
test_image = cv2.imread('test_scene.jpg')

# Extract features from the test image
test_features = extract_features(test_image)

# Create a Bag of Words representation for the test image
```

```
test_representation = create_bow_representation([test_image], vocabulary) # Use the trained classifier to detect cars result = classifier.predict(test_representation)

# Display the result
if result == 'car':
    print('Car detected!')
else:
    print('No car detected.')
```

Input Image:

Output:

Car detected!

Conclusion:

In this experiment, we investigated how to build and train a computer vision object detector using the Bag of Words model. Identifying objects is a basic function that has many uses across different fields. Using local features and machine learning classifiers, the BOW model is an efficient representation technique that lets us identify objects in images. The training data's diversity and quality, along with the classifier and feature selections made, all affect the detector's performance. More refinement and optimisation may result in robuster and more accurate object detection systems.