Vidyavardhini's College of Engineering & Technology Department of Computer Engineering

---

Aim: To Creating and Training an Object Detector

Objective: Bag of Words BOW in computer version Detecting cars in a scene

Theory:

Creating and Training an object detector

Creating and Training an object detector:- Using built-in features makes it easy to come up with a quick prototype for an application. and we're all very grateful to the OpenCV developers for making great features, such as face detection or people detection readily available (truly, we are). However, whether you are a hobbyist or a computer vision professional, it's unlikely that you will only deal with people and faces:

Bag-of -words

Bag-of-words (BOW) is a concept that was not initially intended for computer vision, rather, we use an evolved version of this concept in the context of computer vision. So, let's first talk about its basic version, which-as you may have guessed-originally belongs to the field of language analysis and information retrieval. BOW is the technique by which we assign a count weight to each word in a series of documents; we then represent these documents with vectors that represent these set of counts. Let's look at an example:

Document 1: like OpenCV and I like Python

Document 2: like C++ and Python

Document 3: don't like artichokes

BOW in Computer Vision

We are by now familiar with the concept of image features. We've used feature extractors, such as SIFT, and SURF, to extract features from images so that we could match these features in another image. We've also familiarized ourselves with the concept of codebook, and we know about SVM, a model that can be fed a set of features and utilizes complex

algorithms to classify train data, and can predict the classification of new data. So, the implementation of a BOW approach will involve the following steps: 1. Take a sample dataset. 2. For each image in the dataset, extract descriptors (with SIFT, SURF, and so on). 3. Add each descriptor to the BOW trainer. 4. Cluster the descriptors to k clusters (okay, this sounds obscure, but bear with me) whose centers (centroids) are our visual words

Detecting Cars

There is no virtual limit to the type of objects you can detect in your images and videos. However, to obtain an acceptable level of accuracy, you need a sufficiently large dataset. containing train images that are identical in size. This would be a time-consuming operation if we were to do it all by ourselves

Example – car detection in a scene

We are now ready to apply all the concepts we learned so far to a real-life example, and create a car detector application that scans an image and draws rectangles around cars.

Let's summarize the process before diving into the code:

1. Obtain a train dataset.

2. Create a BOW trainer and create a visual vocabulary.

3. Train an SVM with the vocabulary.

4. Attempt detection using sliding windows on an image pyramid of a test image.

5. Apply non-maximum suppression to overlapping boxes.

6. Output the result.

Code:

```
import cv2

import os

import numpy as np

car_images = [cv2.imread(car) for car in os.listdir("car_dataset")]

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

vocabulary = create_vocabulary(car_features)

training_data = create_bow_representation(car_images, vocabulary)

classifier = train_classifier(training_data, labels)

test_image = cv2.imread('test_scene.jpg')
```

```
test_features = extract_features(test_image)
test_representation = create_bow_representation([test_image], vocabulary)
result = classifier.predict(test_representation)
if result == 'car':
    print('Car detected!')
else:
   print('No Car detected.')
```

Input:

Output:

Car detected!

Conclusion:

The Bag of Words (BoW) model is a key tool in computer vision for detecting cars. It works by extracting features from images, creating a visual vocabulary through feature clustering, and representing images as histograms of visual word occurrences. BoW is particularly useful for car detection as it can handle variations in car appearance, scale, and orientation. When combined with an SVM classifier, BoW helps identify cars in complex scenes. However, its success relies on feature quality and the training dataset's size and diversity. While more advanced techniques like deep learning are prominent, BoW remains a fundamental concept in computer vision.