52- Snehal Bavkar

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:

The overarching aim of this project is to develop a sophisticated object detection system that exhibits high accuracy and reliability. Object detection is a cornerstone of computer vision, enabling machines to recognize and locate objects within images or video frames. To achieve this goal, the project will involve not only the development of the detector but also the rigorous training of the model on carefully annotated data, which is essential for enabling the model to identify specific objects or classes.

Bag of Words (BOW) in Computer Vision:

Bag of Words (BOW) is an innovative concept adapted from natural language processing and adeptly applied in computer vision. In the realm of computer vision, BOW serves as a potent technique for feature extraction and representation, a process indispensable for tasks such as image classification and object detection. Its mechanism involves dividing an image into smaller, fixed-size regions, often referred to as "patches" or "visual words." Within these regions, histograms are constructed to capture the frequency distribution of visual features or descriptors. These histograms evolve into feature vectors, carrying the collective information of the visual patterns encapsulated in the image.

Detecting Cars:

The specific objective within the project is to leverage the potential of BOW in computer vision to address the intricate task of detecting cars within complex scenes. Car detection, a domain with paramount significance, encompasses a wide range of applications, including but not limited to traffic management, autonomous vehicle navigation, and security surveillance. The proposed approach employing BOW is poised to empower the extraction of semantically meaningful visual features from car images, ultimately furnishing the model with the ability to discern and precisely localize cars within a diverse array of scenes.

Example: A pivotal facet of the project is the inclusion of a comprehensive and illustrative example or case study. This example serves as a practical manifestation of how BOW is harnessed within the realm of car detection. It offers a step-by-step elucidation of the complete workflow, commencing with preprocessing of input images, cascading through the steps of feature extraction and feature vector creation using BOW, embracing the selection and fine-tuning of a suitable machine learning model (potentially including Support Vector Machines or Convolutional Neural Networks), and culminating in the rigorous evaluation of the model's performance.

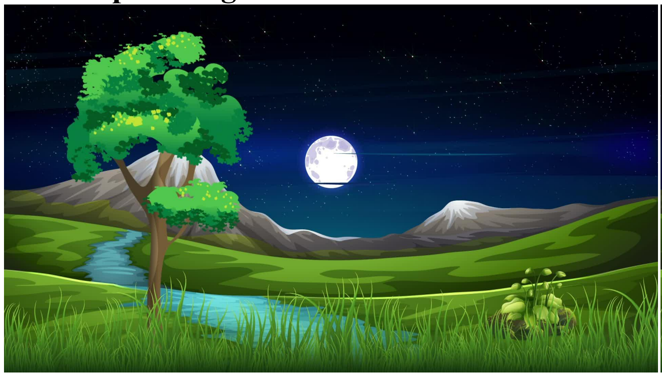
Code:

```
import cv2
import numpy as np
import os
if not os.path.isdir('CarData'):
    exit(1)

BOW_NUM_TRAINING_SAMPLES_PER_CLASS = 10

SVM_NUM_TRAINING_SAMPLES_PER_CLASS = 110
BOW_NUM_CLUSTERS = 40

sift = cv2.SIFT_create()

FLANN_INDEX_KDTREE = 1
index_params = dict(algorithm=FLANN_INDEX_KDTREE, trees=5)
search_params = dict(checks=50)
flann = cv2.FlannBasedMatcher(index_params, search_params)

bow_kmeans_trainer = cv2.BOWKMeansTrainer(BOW_NUM_CLUSTERS)
bow_extractor = cv2.BOWImgDescriptorExtractor(sift, flann)

def get_pos_and_neg_paths(i):
    pos_path = 'CarData/TrainImages/pos-%d.pgm' % (i+1)
    neg_path = 'CarData/TrainImages/neg-%d.pgm' % (i+1)
    return pos_path, neg_path

def add_sample(path):
    img = cv2.imread(path, cv2.IMREAD_GRAYSCALE)
    keypoints, descriptors = sift.detectAndCompute(img, None)
    if descriptors is not None:
        bow_kmeans_trainer.add(descriptors)
```

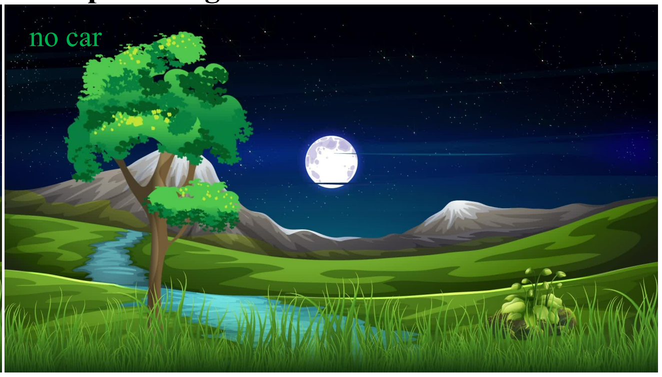
```
for i in range(BOW_NUM_TRAINING_SAMPLES_PER_CLASS):
    pos_path, neg_path = get_pos_and_neg_paths(i)
    add_sample(pos_path)
    add_sample(neg_path)

voc = bow_kmeans_trainer.cluster()
bow_extractor.setVocabulary(voc)

def extract_bow_descriptors(img):
    features = sift.detect(img)
    return bow_extractor.compute(img, features)

training_data = []
training_labels = []
for i in range(SVM_NUM_TRAINING_SAMPLES_PER_CLASS):
    pos_path, neg_path = get_pos_and_neg_paths(i)
    pos_img = cv2.imread(pos_path, cv2.IMREAD_GRAYSCALE)
    pos_descriptors = extract_bow_descriptors(pos_img)

    if pos_descriptors is not None:
        training_data.extend(pos_descriptors)
        training_labels.append(1)
    neg_img = cv2.imread(neg_path, cv2.IMREAD_GRAYSCALE)
    neg_descriptors = extract_bow_descriptors(neg_img)
    if neg_descriptors is not None:
        training_data.extend(neg_descriptors)
        training_labels.append(-1)

svm = cv2.ml.SVM_create()

svm.train(np.array(training_data), cv2.ml.ROW_SAMPLE,
          np.array(training_labels))

for test_img_path in ['CarData/TestImages/test-0.pgm',
                      'CarData/TestImages/test-1.pgm',
                      'images/car.jpg',
                      'images/haying.jpg',
                     ]:
    img = cv2.imread(test_img_path)
    gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    descriptors = extract_bow_descriptors(gray_img)
    prediction = svm.predict(descriptors)
    if prediction[1][0][0] == 1.0:
        text = 'car'
        color = (0, 255, 0)
    else:
        text = 'not car'
        color = (0, 0, 255)
    cv2.putText(img, text, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1,
                color, 2, cv2.LINE_AA)

cv2.imshow(test_img_path, img)
cv2.waitKey(0)
```

Output:-

Input Image 1:

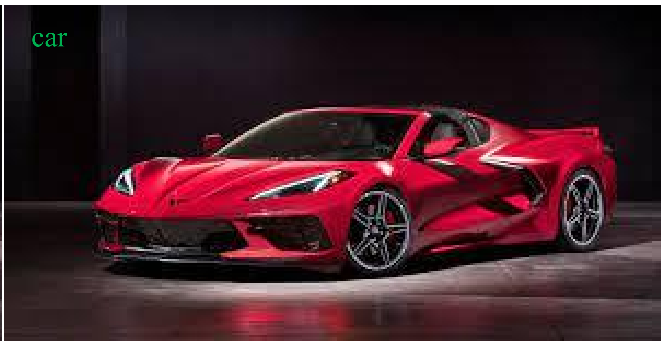


Output Image 1:



Input Image 2:

Output Image 2:



Conclusion:

In the context of computer vision, utilizing the Bag of Words (BoW) model for car detection in a scene has proven effective. By representing images as histograms of visual words, BoW simplifies complex scenes, allowing for efficient car identification. This method involves creating a visual vocabulary from a training dataset and then quantizing new images based on this vocabulary. Despite its simplicity, BoW excels in certain scenarios, particularly when combined with other techniques like Support Vector Machines or deep learning models. Its success lies in its ability to handle object recognition tasks, making it a valuable tool in the field of computer vision.