

Thị giác máy tính (tiếng Anh: computer vision) là một lĩnh vực bao gồm các phương pháp thu nhận, xử lý ảnh kỹ thuật số, phân tích và nhận dạng các hình ảnh và, nói chung là dữ liệu đa chiều từ thế giới thực để cho ra các thông tin số hoặc biểu tượng, ví dụ trong các dạng quyết định. Việc phát triển lĩnh vực này có bối cảnh từ việc sao chép các khả năng thị giác con người bởi sự nhận diện và hiểu biết một hình ảnh mang tính điện tử. Thị giác máy tính bao gồm các lĩnh vực sau: Xử lý hình ảnh: Phát triển các thuật toán xử lý ảnh như tăng/giảm chất lượng ảnh, lọc nhiễu... Nhận diện mẫu: Giải thích các kỹ thuật khác nhau để phân loại mẫu. Quang trắc: Liên quan đến việc thu thập các số đo chính xác từ hình ảnh. I=image, O=description=>cv, I=O=image=>digital image processing, I=O=desc=>AI, I=des, O=I=> computer graphics

HOG là viết tắt của Histogram of Oriented Gradient - một loại “feature descriptor”. Mục đích của “feature descriptor” là trừu tượng hóa đối tượng bằng cách trích xuất ra những đặc trưng của đối tượng đó và bỏ đi những thông tin không hữu ích. Vì vậy, HOG được sử dụng chủ yếu để mô tả hình dạng và sự xuất hiện của một đối tượng trong ảnh.

- 0 Sử dụng gradient magnitude và gradient orientation để lưu thông tin ảnh.
- 1 Các toán tử HOG được cài đặt bằng cách chia nhỏ một bức ảnh thành các vùng con (cells).
- 2 Mỗi cell, tính toán một histogram về các hướng của gradients cho các điểm nằm trong cell.
- 3 Ghép các histogram lại với nhau ta sẽ có một biểu diễn cho bức ảnh ban đầu. patches scale 1:2 (64 128)

```
from skimage import feature
def compute_hog_train():
    train_features = []
    for img in train_images:
        img = cv2.resize(img, (128, 256))
        (hog, hog_image) = feature.hog(img, orientations=9,
                                       pixels_per_cell=(8, 8), cells_per_block=(2, 2),
                                       block_norm='L2-Hys', visualize=True,
                                       transform_sqrt=True)
        train_features.append(hog)
    return train_features
```

Abrupt change in the intensity of pixels Discontinuity in image brightness or contrast
Usually edges occur on the boundary of two regions.

Cạnh (đường viền) ảnh là một đặc trưng rất quan trọng do hệ thống thị giác của con người dựa trên cạnh để nhận thức được đối tượng./ Việc dò tìm cạnh ảnh có thể dựa vào độ thay đổi của các giá trị pixel, hay còn gọi là gradient của ảnh./ Phương pháp dò cạnh có thể được chia làm 2 loại: phụ thuộc hướng và độc lập hướng. Phương pháp dò cạnh phụ thuộc hướng: sẽ dựa trên việc xác định gradient của ảnh theo các hướng x và y. Edge detection is extensively used in image segmentation when we want to divide the image into areas corresponding to different objects. If we want to extract different objects from

an image, we need Edge detection. Using edge detection, we can: recognition, image comparison, unaccepted object can be removed.

Image Segmentation (phân đoạn ảnh) là lớp bài toán thuộc về Computer Vision trong đó hình ảnh được chia thành nhiều vùng khác nhau, trong đó từng pixel trên ảnh được phân loại thành các nhãn khác nhau. Trong Semantic Segmentation, các pixel thuộc cùng 1 class sẽ được gán nhãn giống nhau, còn với Instance Segmentation thì các pixel thuộc cùng 1 vật thể được gán nhãn giống nhau.

Histogram (lược đồ xám) là biểu đồ tần xuất thống kê số lần xuất hiện các mức sáng trong ảnh.

```
hist = cv.calcHist([img],[0],None,[256],[0,256])
def TinhHist(img):
hist = cv.calcHist([img],[0],None,
[256],[0,256])
size = img.shape[0]*img.shape[1]
hist = hist / size
return hist
def TinhHOG(pathfilename):
img = cv.imread(pathfilename,0)
img =cv.resize(img, (128, 256))
(hog, hog_image) = feature.hog(img, orientations=9,
pixels_per_cell=(8, 8), cells_per_block=(2, 2),
block_norm='L2-Hys', visualize=True, transform_sqrt=True)
return hog
```