SCII/00825/2019
MICHAEL ORINA

COMPUTER VISION CAT 2

1. Suppose you have a video/footage titled "cars.mp4", and a classifier titled "cars.xml" write a python script that can be used to detect and draw rectangles on moving cars in the video. (10 marks)

```
import cv2

car_cascade = cv2.CascadeClassifier('cars.xml')

video = cv2.VideoCapture('cars.mp4')

while True:

    ret, frame = video.read()

    if not ret:
        break

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    cars = car_cascade.detectMultiScale(gray, 1.1, 3)

    for (x, y, w, h) in cars:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

    cv2.imshow('Car Detection', frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

video.release()
cv2.destroyAllWindows()
```

2. Suppose you have a cat photo/image titled "cat.jpg", a classifier titled "haarcascade_frontalcatface.xml" , and another classifier titled "haarcascade_eye.xml" , write a python script that can be used to detect and draw rectangles on the eyes of the cat. (10 marks)

```
import cv2

cat_face_cascade = cv2.CascadeClassifier('haarcascade_frontalcatface.xml')
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')

image = cv2.imread('cat.jpg')

gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

```
cat_faces = cat_face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5,
minSize=(30, 30))

for (x, y, w, h) in cat_faces:

    roi_gray = gray[y:y + h, x:x + w]
    roi_color = image[y:y + h, x:x + w]

    eyes = eye_cascade.detectMultiScale(roi_gray)

    for (ex, ey, ew, eh) in eyes:
        cv2.rectangle(roi_color, (ex, ey), (ex + ew, ey + eh), (0, 255, 0), 2)

cv2.imshow('Cat Eye Detection', image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

3. Suppose you have a video titled “pushing_limits.mp4” with human beings within, and a classifier titled “haarcascade_frontalface_default.xml” , write a python script that can be used to detect and draw rectangles on the faces of the human beings in the video. (10 marks)

```
import cv2

face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

video = cv2.VideoCapture('pushing_limits.mp4')

while True:

    ret, frame = video.read()

    if not ret:
        break

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30,
30))

    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

    cv2.imshow('Face Detection', frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

video.release()
cv2.destroyAllWindows()
```

COMPUTER VISION READING ASSIGNMENT

1. Read on how the pin-hole camera works

A pinhole camera is a simple camera that uses a small aperture or pinhole instead of a lens to form an image. Here's how it works:

1. Light enters through the small pinhole: In a pinhole camera, there is a small hole or aperture on one side of a lightproof box. Light from the scene being photographed enters through this tiny opening.

2. Light rays spread out: As the light passes through the pinhole, it spreads out and forms a cone of light rays. The smaller the pinhole, the more focused the cone of light becomes.

3. Image formation: The cone of light rays continues to travel and strikes the opposite side of the box, which is lined with a photosensitive material or film. The light rays intersect at different angles on the film, creating an inverted image of the scene.

4. Image capture: The photosensitive material or film records the pattern of light falling on it, capturing the image formed by the intersecting light rays. The film's chemical composition reacts to light exposure, leading to a latent image.

5. Development: After capturing the image, the film is developed using appropriate chemical processes. The development process converts the latent image into a visible photograph.

6. Viewing the image: Once developed, the photograph can be removed from the camera and viewed by holding it up to light or by scanning it into a digital format.

The pinhole camera's simplicity and lack of a lens result in some unique characteristics. The absence of a lens eliminates distortion and chromatic aberrations, but the small aperture size means the camera has a very small aperture ratio, resulting in long exposure times and reduced light sensitivity. Additionally, pinhole images have a characteristic sharpness throughout the image field, albeit with a certain degree of inherent diffraction due to the small aperture size.

2. Read on the reflection of light and how it is applied in machine vision in addition to understanding the different types of reflection e.g. diffuse reflection

Reflection of light is the phenomenon where light waves bounce off a surface and change direction. It plays a crucial role in machine vision, which is the field of technology that involves capturing and analyzing images for various applications. Here's how reflection of light is applied in machine vision:

1. Surface Inspection: Machine vision systems often use reflection to inspect surfaces for defects or irregularities. By analyzing the reflection pattern, these systems can detect variations in surface texture, color, or shape. For example, in manufacturing industries, machine vision can be used to detect scratches, dents, or other flaws on the surface of products.

2. Object Recognition: Reflections can provide valuable information about the shape and structure of objects. Machine vision algorithms can analyze the reflection patterns to identify and recognize objects based on their unique characteristics. This is commonly used in applications like object sorting, robotic pick-and-place tasks, and quality control.

3. 3D Reconstruction: Reflections can be used to reconstruct a three-dimensional representation of an object or a scene. By analyzing the reflection angles and patterns, machine vision systems can estimate the depth and spatial information of the objects in the captured image. This is useful in applications such as 3D modeling, augmented reality, and robotics.

The different types of reflection:

1. Diffuse Reflection: Diffuse reflection occurs when light strikes a rough or matte surface and scatters in different directions. The reflected light does not form a clear image but instead creates a more even and scattered illumination. Diffuse reflection is commonly seen from surfaces like paper, cloth, or frosted glass. Machine vision systems can analyze diffuse reflection to determine surface properties, such as color or texture.

2. Specular Reflection: Specular reflection occurs when light reflects off a smooth or polished surface at a specific angle, known as the angle of incidence, in a mirror-like manner. The reflected light retains the original angle of incidence, resulting in a clear and well-defined reflection. Specular reflection is used in machine vision for tasks like measuring surface roughness, inspecting reflective coatings, or detecting surface defects that alter the smooth reflection pattern.

3. Retroreflection: Retroreflection involves the reflection of light back to its source. It occurs when light hits a special type of surface that contains small reflective elements, such as corner cubes or microprisms. These surfaces reflect light back along the same path it came from, regardless of the angle of incidence. Retroreflective materials are commonly used in applications like road signs, safety vests, or barcode readers.

Understanding the type of reflection helps machine vision systems interpret and extract valuable information from the captured images, enabling them to perform various tasks efficiently and accurately.

3. Read on the different elements of an image processing system and understand what smoothing is in image processing

An image processing system consists of several essential elements that work together to process and analyze images. Here are some key elements of an image processing system:

1. Image Acquisition: This element involves capturing or acquiring images using cameras, scanners, or other devices. It is the initial step in the image processing pipeline, where raw image data is obtained.

2. Preprocessing: Preprocessing involves applying various techniques to enhance the quality of the acquired images and prepare them for further analysis. This may include operations such as noise reduction, contrast enhancement, image resizing, and color correction.

3. Image Enhancement: Image enhancement techniques aim to improve the visual quality of an image by emphasizing certain features or reducing unwanted artifacts. This can involve techniques like histogram equalization, sharpening, or adjusting brightness and contrast.

4. Smoothing/Filtering: Smoothing, also known as filtering, is a technique used to reduce noise and remove unwanted variations in an image. It involves applying a filter to the image that modifies the pixel values based on their neighboring pixels. Smoothing filters can help in reducing high-frequency noise, blurring edges, or removing unwanted details.

5. Feature Extraction: Feature extraction involves identifying and extracting specific patterns or features from an image that are relevant for further analysis or recognition. This may include techniques like edge detection, corner detection, texture analysis, or object segmentation.

6. Image Analysis: Image analysis focuses on extracting meaningful information and making decisions based on the processed image data. This can involve tasks like object recognition, classification, image segmentation, motion tracking, or measurement.

7. Post-processing: Post-processing involves applying additional operations or transformations to the processed image data, depending on the specific application requirements. This may include tasks such as image stitching, image fusion, image compression, or image display.

Smoothing in image processing:

Smoothing, also known as blurring, is a technique used to reduce noise or unwanted variations in an image. It involves applying a smoothing filter to the image, which modifies the pixel values based on the neighboring pixels. The filter's purpose is to average or smooth out abrupt changes in pixel values, resulting in a more visually pleasing and less noisy image.

The smoothing filter typically operates by replacing each pixel's value with a weighted average of its neighboring pixels' values. The weights assigned to neighboring pixels determine the filter's behavior and can vary based on the specific filter type. Common smoothing filters include the Gaussian filter, mean filter, and median filter.

Smoothing can be beneficial in various image processing tasks. It helps in reducing noise introduced during image acquisition, improving the quality of images for visual inspection, and facilitating subsequent analysis tasks such as edge detection or object recognition. However, excessive smoothing can lead to loss of fine details or blurring of important image features, so choosing an appropriate level of smoothing is important to balance noise reduction and preservation of relevant information in the image.

4. Read on the basics of what a neural network is, the different types of neural networks, the different layers of a neural network, the structure of convolutional neural networks, and how neural networks are used in computer vision

A neural network, also known as an artificial neural network (ANN) or a deep neural network (DNN), is a computational model inspired by the structure and functioning of the human brain. It consists of interconnected nodes called artificial neurons or "units" that work together to process and analyze information.

Neural networks are designed to learn from data and make predictions or perform tasks by adjusting the strengths (weights) of connections between neurons. They are composed of different layers, each serving a specific purpose. Here are the key components and concepts related to neural networks:

1. Neurons/Units: Neurons are the basic building blocks of a neural network. They receive input signals, perform a mathematical operation on the input, and produce an output signal. The output signal is typically passed through an activation function to introduce non-linearity into the network.

2. Layers: A neural network is structured in layers, which are stacked on top of each other. The most common types of layers in a neural network are:

   - Input Layer: The input layer receives the initial data or input features and passes them to the subsequent layers.

   - Hidden Layers: Hidden layers are intermediate layers between the input and output layers. They process the information and learn representations through the adjustment of weights.

   - Output Layer: The output layer provides the final predictions or outputs of the neural network based on the learned representations.

3. Types of Neural Networks: There are various types of neural networks, each designed for specific tasks and data types. Some common types include:

   - Feedforward Neural Network (FNN): This is the simplest type of neural network, where information flows only in one direction, from the input layer to the output layer. They are used for tasks like classification and regression.

   - Convolutional Neural Network (CNN): CNNs are primarily used for computer vision tasks. They consist of convolutional layers that perform localized and shared-weight computations, pooling layers for downsampling, and fully connected layers for classification or regression.

   - Recurrent Neural Network (RNN): RNNs are designed for sequence data, where information flows in loops or feedback connections. They have memory to retain information about past inputs, making them suitable for tasks like natural language processing and speech recognition.

   - Long Short-Term Memory (LSTM): LSTM is a type of RNN that addresses the vanishing gradient problem and can retain information over longer sequences. It is commonly used in tasks involving sequential data analysis.

4. Convolutional Neural Networks (CNNs) in Computer Vision: CNNs are widely used in computer vision tasks due to their ability to effectively analyze visual data. They have a specific structure:

   - Convolutional Layers: These layers consist of filters (kernels) that slide over the input image, applying convolution operations. This helps to detect patterns and features at various spatial locations.

   - Pooling Layers: Pooling layers reduce the spatial size of the convolutional feature maps while retaining the important information. This helps in reducing computation and providing translation invariance.

   - Fully Connected Layers: These layers connect all the neurons from the previous layer to the next layer, enabling classification or regression based on the learned features.

5. Computer Vision Applications: Neural networks are extensively used in computer vision applications. They can perform tasks such as image classification, object detection and tracking, image segmentation, facial recognition, and image generation. By leveraging the hierarchical representation learning capabilities of neural networks, they can extract meaningful features and patterns from images, enabling sophisticated visual analysis and understanding.

Overall, neural networks have revolutionized computer vision by enabling machines to learn from large amounts of visual data and perform complex visual tasks with high accuracy and efficiency.