Updated on 03/12/2024

Sign up

OpenCV training

3 days (21 hours)

Presentation

Our OpenCV training course will enable you to develop image detection and processing applications using Python.

We'll start with an introduction to this development framework, explaining the implications and importance of computer vision. Once OpenCV has been configured, you can familiarize yourself with the main functions.

You'll discover basic and advanced image processing techniques such as binarization, blurring and dilation. You'll learn how to apply detection methods such as SIFT, SURF and ORB.

Finally, you'll deploy a neural network-based machine learning model to take full advantage of OpenCV's power.

As with all our training courses, this one will introduce you to the latest stable version of OpenCV 4.10, coupled with Python 3.12.

Objectives

- Master OpenCV installation and configuration
- Understand and apply basic and advanced image manipulation and processing techniques
- Implementing machine learning models

Target audience

- Data Analyst / Data Scientist / Data Engineer
- Developers, Big Data Architects, Lead Developers

Prerequisites

- Knowledge of Python
- Knowledge of Deep Learning
- Mathematical knowledge (example: gradient)

OUR OPENCV TRAINING PROGRAM

Introduction to OpenCV and image processing

- History, use cases and fundamental concepts
- OpenCV installation and configuration
- Overview of necessary dependencies (Python, NumPy, Matplotlib)
- Fundamentals: pixels, image matrix, color spaces (RGB, HSV, LAB)

Putting it into practice

- Guided installation and configuration of environments
- Loading, displaying and simple image manipulation with OpenCV (cv2.imread, cv2.imshow)
- Basic manipulations: resize, rotate, convert to grayscale

Advanced image manipulation

- Histograms: analysis and equalization
- Filtering techniques: Gaussian blur, bilateral blur, noise reduction
- Geometric transformations: translation, rotation and homography.s

Putting it into practice

- Creating and displaying histograms with OpenCV
- Application of different filters to improve image quality
- Implementing geometric transformations on images
- Webcam project + real-time image processing

Advanced image processing and feature detection

- Feature detection with SIFT, SURF and ORB
  - Explanation of SIFT, SURF and ORB methods.
  - Comparing performance and use cases

- Pattern recognition and object detection
  - Contour detection with Canny and Sobel
  - Detection and tracking of simple shapes (lines, circles, contours)
  - Introduction to Haar Cascades for face detection

Putting it into practice

- Implementation of SIFT and SURF to detect key points on images
- ORB as a fast, patent-free alternative
- Visual comparison of results for the three methods
- Edge detection implementation
- Detection and tracking of simple shapes in images and videos
- Facial detection with Haar Cascades on images and video streams

Introduction to Scikit Learn

- Basic concept: pipeline, training and testing
- Algorithms available: classification, regression, clustering
- Geometric transformations: translation, rotation and homography.

Putting it into practice

- OpenCV + Scikit-Learn integration
  - Demonstration: A complete pipeline for object detection and classification
  - Practical example: Classification of simple objects based on their contours

Machine learning for computer vision

- Introduction to convolutional neural networks (CNN)
  - Fundamental concepts: convolutional layers, pooling, fully connected.
  - CNN applications in computer vision
  - Introduction to TensorFlow and Keras for model development
- Image classification with a CNN
  - Data pre-processing for CNNs
  - Presentation of the MNIST dataset for handwritten digit recognition
- Advanced implementation and deployment
  - Adjustment of CNN model hyperparameters
  - Introduction to fine-tuning pre-trained models
  - Solution deployment and integration
- Deployment and integration
  - Best practices for integrating computer vision models into mobile applications
  - Security and performance concepts for production applications

Putting it into practice

- Creating and training a CNN model with TensorFlow/Keras
- Model test on MNIST dataset (or other if supplied)
- Model performance analysis
- Mini-project: Handwritten digit recognition with a CNN
  - Training and enhancing a CNN model
  - Evaluation on customized data
  - Exporting the model for deployment

Closing and resources

- Question and answer session, QCM validation and satisfaction (30 min)
- Distribution of training materials, code examples and recommendations for further reading

Computer Vision training with Tensorflow

Pytorch training

Tensorflow training

Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer technology, or to acquire specific business knowledge or modern methods.

Positioning on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as registration is finalized, the learner receives a self-assessment questionnaire which enables us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and personal objectives for the training to come, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or security difficulties within the company (intra-company or virtual classroom) which could be problematic for the follow-up and smooth running of the training session.

Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

Organization

The course alternates theoretical inputs from the trainer supported by examples and brainstorming sessions and group work.

Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

Sanction

A certificate will be issued to each trainee who completes the course.