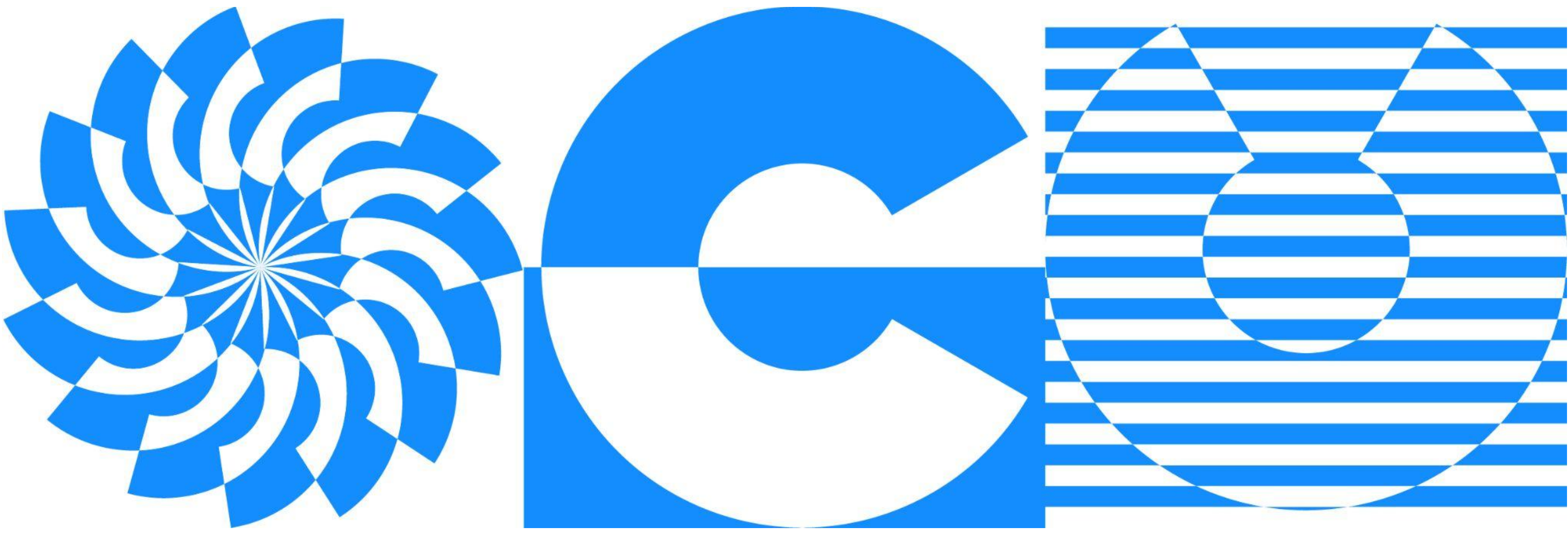
A place where legacy creates future.



Program

CVDL Master

Detailed Curriculum

OpenCV University

Index

Course 1 - MOCV

Mastering OpenCV with Python

1. Getting Started with Images

1.1. Image Basics (Basics of Image Formation and Image Formats)
1.2. Reading, Display and Writing Images in OpenCV
1.3. Color space conversion and different color spaces
1.4. Basic image manipulation (resizing, cropping, annotating, creating a Region of Interest)

2. Basic Image Operations

2.1. Mathematical operations on images (brightness and contrast)
2.2. Image thresholding, bitwise operations and masking
2.3. Image blending and the alpha channel
2.4. Creating Digital signatures using alpha blending

3. Histograms and Color Segmentation

3.1. Image histograms and enhancement using Histogram equalization
3.2. Color segmentation on images
3.3. Deforestation analysis using Color Segmentation
3.4. Satellite Imagery analysis using GeoTIFF Images

4. Video Processing and Analysis

4.1. Reading and Writing videos using OpenCV
4.2. Motion Detection analysis using Background Subtraction

5. Contours and Shape Analysis

5.1. Finding and Drawing Contours
5.2. Intruder detection using Contour Analysis

6. Human Computer interaction(HCI) using PyAutoGUI

6.1. Keyboard and Mouse Controls using PyAutoGUI
6.2. Playing Online Games using PyAutoGUI

7. Building and Deploying Apps with Streamlit

7.1. Building a Face detection application using streamlit

OpenCV University

Course 2 - CVIP

Fundamentals of Computer Vision & Image Processing

Module 1

Getting Started with OpenCV

|

Module 2

Video IO and GUI

|

Module 3

Binary Image Processing

|

Module 4

Image Enhancement and Filtering

|

Module 5

Advanced Image Processing and Computational Photography

|

Module 6

Geometric Transforms and Image Features

|

Module 7

Image Segmentation and Recognition

|

Module 8

Video Analysis and Object Tracking

|

Module 9

Deep Learning with OpenCV

1 Getting Started with OpenCV

Assignment 1	**Build a QR Code Detector**

2 Video IO & GUI

2.1 Video IO using HighGUI

2.1.1 Video IO jargon

2.1.2 Read and display video

2.1.3 Properties of video capture

2.1.4 How to write a video

2.2 Callback Functions

2.2.1 What are Callback Functions?

2.3 Keyboard as input device

2.3.1 How to take input from keyboard

Assignment 2	**Perform Image Annotation Using Mouse**
Assignment 3	**Enhance GUI features with Sliders**

3 Binary Image Processing

3.1 Thresholding

3.1.1 What is Thresholding?

3.1.2 Thresholding in OpenCV

3.2 Erosion / Dilation

3.2.1 Overview on Erosion and Dilation

3.2.2 Erosion and Dilation in OpenCV

3.3 Opening and Closing

3.3.1 Overview on Opening and Closing

3.3.2 Opening and Closing on OpenCV

3.4 Connected Component Analysis

3.4.1 What is Connected Component Analysis?

3.4.2 Connected Component Analysis in OpenCV

3.5 Contour Analysis

3.5.1 What are Contours?

3.5.2 Contour Analysis in OpenCV

3.6 Blob detection

3.6.1 Blob detection in OpenCV

Assignment 4	**Implement Different Morphological Operations**
Assignment 5	**Build a Coin Detection application using Contours**

4 Image Enhancement & Filtering

Assignment 6	**Convert Your Images Into Different Color Spaces**
Assignment 7	**Implement Camera Autofocus using Image Analysis**

5 Advanced Image Processing & Computational Photography

5.1 Hough transforms

5.1.1 What is a hough transform?

5.1.2 HoughLine: How to detect a line in an image?

5.1.3 HoughCircle: How to detect a circle in an image?

5.2 High Dynamic Range imaging

5.2.1 What is High Dynamic Range (HDR) imaging?

5.2.2 HDR in OpenCV

5.3 Seamless cloning

5.3.1 What is seamless cloning?

5.3.2 Seamless cloning in OpenCV

5.3.3 Application: Face Blending

5.4 Image inpainting

5.4.1 What is image inpainting?

Project 1	• **Create Your Own Instagram Filter** • **Blemish Removal From Face** • **Chroma Keying**

6 Geometric Transforms & Image Features

6.1 Geometric transforms

6.1.1 Affine transform

6.1.2 Homography

6.1.3 Geometric transforms in OpenCV

6.2 Image features

6.2.1 Image feature: ORB

6.2.2 ORB feature in OpenCV

6.3 Feature matching

6.3.1 Different feature matching algorithms in OpenCV

6.3.2 RANSAC

6.4 Application: Image alignment

6.5 Application: Creating panoramas
6.6 Application: Finding known objects using OpenCV

Assignment 8	**Create Panorama from Multiple Images**
Assignment 9	**Feature Matching-Based Image Alignment**
Project 2	**Implement a Document Scanner Application**

7 Image Segmentation & Recognition

7.1 Image segmentation using GrabCut
- 7.1.1 GrabCut theory
- 7.1.2 GrabCut in OpenCV

7.2 Introduction to AI
- 7.2.1 Basic overview of AI

7.3 Image classification
- 7.3.1 Histogram of Oriented Gradients (HOG)
- 7.3.2 Support Vector Machines (SVM)
- 7.3.3 Eyeglass Classifier in OpenCV

7.4 Object detection
- 7.4.1 Pedestrian detection in OpenCV
- 7.4.2 Face detection using HAAR Cascade
- 7.4.3 Face detection in OpenCV

Project 3	**Create Your Own Selfie Application** ● **Skin Smoothing Filter** ● **Sunglass Filter**

8 Video Analysis

8.1 Motion estimation using Optical Flow
- 8.1.1 What is Optical Flow?
- 8.1.2 Lucas-Kanade Optical Flow

8.2 Application: Video stabilization

8.3 Object tracking
- 8.3.1 Different object tracking algorithms

8.4 Different object trackers in OpenCV

8.4.1 Object tracking in OpenCV

8.4.2 Comparison of different trackers

8.5 Multiple object tracking

8.5.1 How to track multiple objects in OpenCV

8.6 Kalman filtering

8.6.1 Kalman Filter Tracker

8.7 MeanShift and CamShift

8.7.1 Tracking using MeanShift and CamShift

Project 4	**Implement fusion of Detection + Tracking**

9 Deep Learning with OpenCV

9.1 Image classification

9.1.1 Image classification using Caffe and TensorFlow

9.2 Object detection

9.2.1 Single Shot Multibox Detector (SSD)

9.2.2 You Only Look Once Detector (YOLO)

9.3 Face detection

9.3.1 SSD-Based Face Detector

9.4 Human pose estimation

9.4.1 OpenPose

Course 3 - DLPT

Deep Learning with PyTorch

Module 1

Getting Started with PyTorch

|

Module 2

Neural Networks

|

Module 3

Convolutional Neural Networks

|

Module 4

Deep Neural Networks

|

Module 5

Best Practices in Deep Learning

|

Module 6

Object Detection

|

Module 7

Single Stage Object Detectors

|

Module 8

Segmentation

|

Module 9

Pose Estimation

|

Module 10

Bonus Module: Generative Adversarial Networks (GANs)

1 Getting Started with PyTorch

Assignment 1	**Implement ReLu, Softmax & Neuron Using PyTorch**
Assignment 2	**Implement Gradient Descent For Two Variables**

2 Neural Networks

Assignment 3	Implement MSE & MAE Loss Functions

Assignment 4	Implement CNN For Image Classification On CIFAR10 Dataset

OpenCV University

OpenCV University

Course 4 - DLTK

Deep Learning with TensorFlow & Keras

Module 1

Getting Started with Tensorflow and Keras

|

Module 2

Neural Networks

|

Module 3

Convolutional Neural Networks

|

Module 4

Best Practices In Deep Learning

|

Module 5

Object Detection

|

Module 6

Image Segmentation

|

Module 7

Pose Estimation

|

Module 8

Bonus: Generative Adversarial Networks (GANS)

1 Getting Started with TensorFlow and Keras

1.1 Introduction to Artificial Intelligence
1.2 AI in Computer Vision
1.3 AI terminology
1.4 NumPy refresher
1.5 Introduction to TensorFlow & Keras
1.6 What is inside an ML algorithm?
1.7 Gradient-based Optimization techniques in ML

Assignment 1	**Implement Leaky ReLu, Softmax & Convolution Using TensorFlow**
Assignment 2	**Implement Gradient Descent For Two Variables**

2 Neural Networks

2.1 Feature Vectors & Normalization
2.2 Basics of Neural Networks
2.3 Understanding Backpropagation in a Neural Network
2.4 Loss Functions for Recgression and Classification Tasks.
2.5 Mathematical foundation for Regression & Classification
2.6 Implementing a single neuron to model Linear Regression
2.7 Binary Classification using Perceptrons
2.8 Image Classification using Multilayer Perceptron
2.9 Model Complexit, Generalizaation and handling Overfitting
2.10 Understand and implement the building blocks like the different Activation Functions, Loss Functions, Hidden Layers

Assignment 3	**Implement A Multi-Layer Perceptron For Apparel Classification**

3 Convolutional Neural Networks

3.1 The Convolution operation
3.2 CNN building blocks and Layers
3.3 Evaluation metrics for Classification Performance
3.4 Important CNN architectures like VGGNet and ResNet
3.5 Leverage pre-trained models using Transfer Learning and Fine-tuning
3.6 Building custom models from scratch using your own data
3.7 Handling Overfitting through Data augmentations

OpenCV University

Assignment 4	**Conversion from Sequential API to Functional API.**
Assignment 5	**Implement a CNN For Image Classification On CIFAR10/100 Dataset**

Project 1	**Implement An Image Classifier From Scratch**

4 Best Practices in Deep Learning

4.1 Advanced Optimizers in Keras

4.2 Training Deep Neural Networks

4.3 Handling Data in TensorFlow using TF Data, Sequence Class, and TF Records

4.4 Learning Rate Schedulers

4.5 Model Interpretability using GradCAM

Assignment 6	**Implement AdamW Optimizer**

Project 2	**Kaggle Competition For Sports Item Classification**

5 Semantic Segmentation

5.1 Problem formulation, custom layers and Loss Functions associated with Segmentation, like **Dilated & Transposed Convolution, Dice Loss, Pixel Accuracy,** etc.

5.2 Semantic Segmentation Datasets and Architectures

5.3 Evaluation metrics for Semantic Segmentation

5.4 Custom DataLoader for Segementation Datasets

5.5 Fully Convolutional Networks (**FCNs**) and **UNet** training for Binary Segmentation on **Road Data**

5.6 UNet and **DeepLab** v3+ Training on Multi-class segmentation on **CamVid** Data.

5.7 Improving **DeepLab** v3+ on Segmentation of **Underwater Imagery (SUIM)**

Assignment 7	**Implementation of PSPNet**

Project 3	**Kaggle Competition On Semantic Segmentation Of FloodNet Dataset**

6 Object Detection

6.1 Evolution from traditional Object Detection algorithms to state-of-the-art Deep Learning-based methods

6.2 Exploring Object detection Datasets.

6.3 Revisiting Image Classification, and comparing it with Object Detection.

6.4 Two-Stage & Single Stage Object Detectors

6.5 Problem formulation, custom layers and Loss Functions used in Object Detection like Anchors, NMS, IoU, etc.

6.6 Learn about Object Detection models like RCNN, SSD, RetinaNet, YOLO, EfficientDet

6.7 Evaluation metrics for Object Detection: Precision, Recall, Average Precison(AP) and mean average Precison (mAP)

6.8 Using the TensorFlow Object Detection (TFOD) API

6.9 Fine-tuning of Object Detection Models available on TFOD API on a subset of Pascal VOC data.

6.10 Building a Custom SSD Model with FPN and training it on PenFudanPed Dataset

Assignment 8	**Encoding And Decoding Of Ground Truths For Anchor Box Implementation**

Project 4	**Build A Custom Object Detection Model For Detecting Safety Kits In Construction Sites**

7 Pose Estimation using MediaPipe

7.1 Real-time Posture analysis using MediaPipe Pose

7.2 Drowsy Driver Detection using MediaPipe

8 Generative Adversarial Networks (GANS)

8.1 Introduction to GANs

8.2 Vanilla GAN using Fashion MNIST

8.3 DCGAN using Flickr Faces

8.4 CGAN using Fashion MNIST

Course 5 - DLAP

Computer Vision and Deep Learning Applications

Module 1

Facial Landmark Detection

|

Module 2

Applications of Facial Landmarks

|

Module 3

Snapchat Filters

|

Module 4

Face Recognition

|

Module 5

Introduction to Deep Learning with CNNs

|

Module 6

State-of-the-art Object Detection

|

Module 7

Text Detection And OCR

|

Module 8

Deploy Applications On Cloud

1 Facial Landmark Detection

1.1 Different face processing applications

1.2 Facial Landmark Detection using Dlib

- 1.2.1 Introduction to Dlib
- 1.2.2 Facial Landmark Detection using Dlib

1.3 Application: Face Alignment

1.4 Improving Facial Landmark Detector

- 1.4.1 Improving speed
- 1.4.2 Improving stability
- 1.4.3 Landmark stabilization in OpenCV
- 1.4.4 Optical Flow
- 1.4.5 Lucas-Kanade Optical Flow

1.5 Facial Landmark Detection Theory

- 1.5.1 Machine Learning basics
- 1.5.2 Research paper review

1.6 Train a custom Facial Landmark Detector

- 1.6.1 How to train a custom Landmark Detector?

2 Applications of Facial Landmarks

2.1 Alpha Blending

- 2.1.1 Alpha Blending in OpenCV

2.2 Image Warping

- 2.2.1 Affine Transform
- 2.2.2 Geometric Transform
- 2.2.3 Triangle Warping

2.3 Delaunay Triangulation

- 2.3.1 Theory of Delaunay Triangulation

2.4 Face Averaging

- 2.4.1 Face Averaging using OpenCV

2.5 Face Morphing

- 2.5.1 Face Morphing using OpenCV

2.6 Application: Bug Eyes

- 2.6.1 Create a Big Eye app

2.7 Head Pose Estimation

- 2.7.1 Head Pose Estimation in OpenCV

Assignment 1	**Build an Automatic Smile Detection Application**

Project 1	**Build a Virtual Makeup Application**

Project 2	**DoppelGanger: Find Celebrity Look-Alike**

5.4.1 Weights and Biases Logging
5.4.2 Transfer Learning
5.4.3 Fine-Tuning

Assignment 2	**Improve CNN Performance during Training**

6 Object Detection

6.1 Object Detection overview

6.1.1 What is Object Detection?
6.1.2 Traditional Object Detection pipeline

6.2 YOLOv3 using DarkNet [To be updated to YOLOv8 using Ultralytics]

6.2.1 Inference using YOLOv3
6.2.2 Fine-Tuning using YOLOv3

6.3 YOLO-NAS [Will be added as part of the course update]

6.3.1 Inference using YOLO-NAS
6.3.2 Fine-Tuning using YOLO-NAS

6.4 RT-DETR [Will be added as part of the course update]

6.4.1 Inference using RT-DETR
6.4.2 Fine-Tuning using RT-DETR

Project 3	**Train A Face Mask Detector using YOLO**

7 Text Detection & OCR

7.1 Overview of OCR

7.1.1 What is OCR?
7.1.2 The OCR pipeline
7.1.3 Challenges
7.1.4 Datasets and competitions

7.2 Graphic Text Recognition using Tesseract

7.2.1 What is Tesseract?
7.2.2 Introduction to OCR using Tesseract
7.2.3 Tesseract OCR failure cases
7.2.4 Improving Tesseract OCR failures

7.3 Text Detection

7.3.1 Text Recognition using EAST
7.3.2 Text Recognition using CRAFT (Python)

7.4 Modified pipeline for scene Text Recognition using Tesseract (Python)

Assignment 3	**Build an app to perform OCR On Invoices**

8 Deploy Applications on Cloud

Assignment 4	**Deploy Your Web App On Heroku**

Explore Other Courses