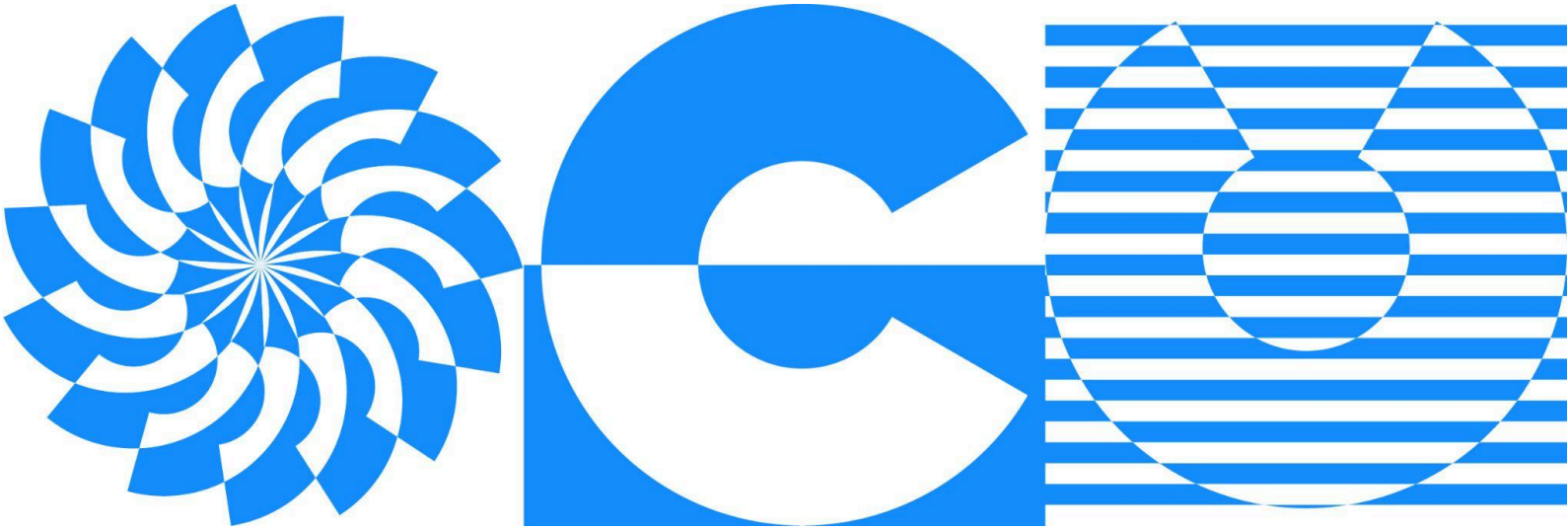
A place where legacy creates future.



Program

CVDL Master

Detailed Curriculum

OpenCV University

Index

Course 1 - MOCV

Mastering OpenCV with Python

Module 1

Getting Started with Images

|

Module 2

Basic Image Operations

|

Module 3

Histogram and Color Segmentation

|

Module 4

Video Processing and Analysis

|

Module 5

Contour and Shape Analysis

|

Module 6

Playing Games Using CV (HCI)

|

Module 7

Building & Deploy Web Apps

|

Module 8

Image Filtering and Enhancement

|

Module 9

Lane Detection using Hough Transforms

|

Module 10

Image Restoration Techniques

|

Module 11

Image Registration Techniques

|

Module 12

ArUco Markers for Augmented Reality

|

Module 13

Deep Learning with OpenCV

|

Module 14

Face and Landmarks Detection

|

Module 15

Object Detection

|

Module 16

Object Tracking

|

Module 17

Human Pose Estimation

|

Module 18

Person Segmentation

|

Module 19

Text Detection and OCR

|

Module 20

Super Resolution

|

Module 21

Barcodes and QR Codes using OpenCV

|

Module 22

Deploying Applications on the Cloud

Module 23

Depth Applications using OpenCV

1. **Getting Started with Images**
   - 1.1. **Image Basics**
     - 1.1.1. Geometry of Image Formation
     - 1.1.2. Digital Image Formation
     - 1.1.3. Image Formats
   - 1.2. **Getting Started with Images**
     - 1.2.1. Reading, Displaying and Saving Images
     - 1.2.2. Color Images
     - 1.2.3. Basic Image Manipulation
     - 1.2.4. Annotating Images

2. **Basic Image Operations**
   - 2.1. **Arithmetic & Logical Operators**
     - 2.1.1. Arithmetic Operations on Images
     - 2.1.2. Thresholding in Images
     - 2.1.3. Logical Operations on Images
   - 2.2. **Alpha Channel**
     - 2.2.1. Understanding and Using the Alpha Channel
   - 2.3. **Applications**
     - 2.3.1. Creating Watermarks
     - 2.3.2. Creating a Digital Signature

3. **Histograms and Color Segmentation**
   - 3.1. **Histograms and Color Segmentation**
     - 3.1.1. Histograms
     - 3.1.2. Color Segmentation
   - 3.2. **Applications**
     - 3.2.1. Deforestation using Color Segmentation
     - 3.2.2. Analyzing Satellite Imagery using GeoTIFF Images

4. **Video Processing and Analysis**
   - 4.1. **Video Processing**
     - 4.1.1. Reading and Streaming Videos
     - 4.1.2. Writing Videos
     - 4.1.3. Motion Detection in Videos

5. **Contour and Shape Analysis**
   - 5.1. **Contour and Shape Analysis**
     - 5.1.1. Finding and Drawing Contours
   - 5.2. **Applications**

5.2.1. Intruder Detection

6. Playing Games using CV (HCI - Human Computer Interface)

6.1. PyAutoGUI Overview

6.1.1. Introduction to HCI with PyAutoGUI

6.2. Application

6.2.1. Playing Online Games with Faces

7. Building and Deploying Web Apps with Streamlit

7.1. Creating Web Applications using Streamlit

7.1.1. Introduction to Streamlit

7.1.2. Face Detection Web App using Streamlit

7.2. Deploying Streamlit Application to Cloud

7.2.1. App Deployment using Streamlit Share

7.2.2. App Deployment on Heroku

8. Image Filtering and Enhancement

8.1. Introduction to Image Filtering

8.1.1. Basics of Image Filtering

8.1.2. What is Convolution

8.2. Image Smoothing & Sharpening

8.2.1. Image Smoothing and Sharpening using Convolution

8.3. Edge Detection

8.3.1. Edge Detection Methods: Sobel and Canny

8.3.2. Canny Edge Detection Demo

8.4. Applications

8.4.1. Photoshop-like Artistic Filters

8.4.2. Build a Web App for Artistic Filters

9. Lane Detection using Hough Transforms

9.1. Lane Detection using Hough Transform

9.1.1. Detecting Lines using Hough Transform in Images

9.1.2. Lane Detection in Videos

10. Image Restoration Techniques

10.1. Image Restoration using Filtering and Inpainting

10.1.1. Noise Reduction with Median and Bilateral Filtering

10.1.2. Fix Images using Inpainting

10.2. Applications

10.2.1. Image Restoration App

11. Image Registration Techniques

- **11.1. Geometric Transformation and Image Features**
  - 11.1.1. Color Photography and Image Alignment
  - 11.1.2. Affine Transformations
  - 11.1.3. Homography/Perspective Transform
  - 11.1.4. Image Features
- **11.2. Image Registration using Homography and Feature Matching**
  - 11.2.1. Image Alignment Demo
- **11.3. Applications**
  - 11.3.1. Virtual Billboard
  - 11.3.2. Creating Panoramas

12. ArUco Markers for Augmented Reality

- **12.1. Introduction to ArUco Markers**
  - 12.1.1. ArUco Markers Overview
- **12.2. Application**
  - 12.2.1. Augmented Reality using ArUco Markers

13. Deep Learning with OpenCV

- **13.1. OpenCV DNN Module**
  - 13.1.1. Introduction to Deep Learning with OpenCV
  - 13.1.2. Image Classification using OpenCV DNN Module
- **13.2. Application**
  - 13.2.1. Build a Web App for Classifying Images

14. Face and Landmark Detection

- **14.1. Face Detection in OpenCV**
  - 14.1.1. Face Detection using OpenCV DNN Module
- **14.2. Application**
  - 14.2.1. Face Blurring for Privacy Preservation
- **14.3. Detecting Facial Landmarks**
  - 14.3.1. Facial Landmarks Detection using OpenCV
  - 14.3.2. Application: Blink Detection

15. Object Detection

- **15.1. Object Detection in OpenCV**
  - 15.1.1. Object Detection using SSD MobileNet
  - 15.1.2. Object Detection using YOLOv4
  - 15.1.3. Object Detection using YOLOv5
- **15.2. Application**
  - 15.2.1. Social Distance Monitoring

16. Object Tracking

16.1. Object Tracking using OpenCV

16.1.1. Introduction to Object Tracking

16.1.2. Analyzing Object Tracking Methods

17. Human Pose Estimation

17.1. Human Pose Estimation using MediaPipe

17.1.1. Introduction to Mediapipe for Human Pose Estimation

17.2. Application

17.2.1. Sports Analytics: Golf Swing Analysis and Training

18. Person Segmentation

18.1. Person Segmentation using Mediapipe

18.1.1. Implementing Portrait Mode & Color Pop

19. Text Detection and OCR

19.1. Text Detection and Recognition

19.1.1. Text Detection using OpenCV DNN

19.1.2. Text Recognition in Natural Scenes

19.2. Application: Build a Language Translation App

19.2.1. Translate Language in Images

20. Super Resolution

20.1. Image Super-Resolution Techniques in OpenCV

21. Barcodes and QR Codes using OpenCV

21.1. Introduction to QR Code and Bar Code

21.1.1. Introduction to QR Code and Bar Code

21.2. Bar Code

21.2.1. Generating Bar Codes

21.2.2. Detect and Decode Bar Codes

21.3. QR Code

21.3.1. Generating QR Codes

21.3.2. Detect and Decode QR Codes

22. Deploying Web Applications on Cloud Services

22.1. Google Cloud Deployment

22.1.1. GCP Account Setup

22.1.2. Deploy Web App using GCP

22.2. Amazon Web Services

22.2.1. AWS Account Setup

22.2.2. Deploy Web App on AWS

Explore Other Courses

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

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 the 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

4.1 Color spaces

- 4.1.1 RGB color space
- 4.1.2 HSV color space
- 4.1.3 Other color spaces
- 4.1.4 Application: Finding dominant color in an image
- 4.1.5 Application: Desaturation Filter

4.2 Color transforms

- 4.2.1 Histogram Equalization
- 4.2.2 Advanced Histogram Equalization (CLAHE)
- 4.2.3 Color adjustment using curves

4.3 Image filtering

- 4.3.1 Introduction to image filtering
- 4.3.2 What is Convolution?
- 4.3.3 Convolution in OpenCV

4.4 Image smoothing

- 4.4.1 Box Blur
- 4.4.2 Gaussian Blur
- 4.4.3 Median Blur
- 4.4.4 Median Blur in OpenCV
- 4.4.5 Bilateral filtering
- 4.4.6 Bilateral Blur in OpenCV
- 4.4.7 Comparison: Median VS Bilateral

4.5 Image gradients

- 4.5.1 Introduction to image gradients
- 4.5.2 First Order Derivative Filters
- 4.5.3 Why is smoothing important before gradient?
- 4.5.4 Second Order Derivative Filters
- 4.5.5 Application: Sharpening Filter
- 4.5.6 Canny Edge Detection
- 4.5.7 Canny Edge Detection in OpenCV

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 a 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

Summary of Assessments

- **Total Projects: 4**
- **Total Assignments: 9**

Explore Other Courses

Course 3 - DLPT

Deep Learning with PyTorch

Module 1

Getting Started

|

Module 2

Neural Networks

|

Module 3

Convolutional Neural Networks

|

Module 4

Practical Considerations for Training Deep Learning Networks

|

Module 5

Best Practices in Deep Learning

|

Module 6

Segmentation

|

Module 7

Object Detection Fundamentals

|

Module 8

Object Detection Fine-tuning and Training

|

Module 9

Introduction to Generative Adversarial Networks (GANs)

|

Module 10

Pose Estimation

1 Getting Started

1.1 Introduction to Artificial Intelligence

- 1.1.1 History of AI
- 1.1.2 Applications of AI
- 1.1.3 AI in Computer Vision
- 1.1.4 AI Terminology
- 1.1.5 Popularity of Deep Learning
- 1.1.6 Deep Learning Frameworks

1.2 NumPy refresher

- 1.2.1 Notebook: Numpy Refresher Part-1
- 1.2.2 Notebook: Numpy Refresher Part-2
- 1.2.3 Notebook: Numpy Refresher Part-3

Assignment 1	NumPy(Ungraded)

1.3 Introduction to PyTorch

- 1.3.1 Why PyTorch?
- 1.3.2 Notebook: PyTorch Basics

Assignment 2	PyTorch

1.4 What's Inside an ML Algorithm?

- 1.4.1 Machine Learning Pipeline
- 1.4.2 Solving ML Problems
- 1.4.3 Notebook: Gradient Descent for Optimisation

Assignment 3	Gradient Descent

2 Neural Networks

2.1 Understanding Neural Networks

- 2.1.1 Deep Learning Overview
- 2.1.2 What is a Neural Network?
- 2.1.3 Feature Vectors and Normalization
- 2.1.4 Demystifying Neural Networks

2.2 Building Neural Networks in PyTorch

- 2.2.1 Notebook: Introduction to Linear Regression
- 2.2.2 Notebook: Auto-MPG Data Processing
- 2.2.3 Linear FRegression with PyTorch
- 2.2.4 Notebook: Binary Classification with PyTorch

2.3 Building Blocks of a Neural Network

- 2.3.1 Notebook: Intro to PyTorch NN Module
- 2.3.2 Loss Function for Regression
- 2.3.3 Notebook: Regression Loss
- 2.3.4 Classification Loss Functions
- 2.3.5 Notebook: Activation Functions
- 2.3.6 How does the Network Learn?

2.4 Multi-Class Classification using PyTorch

- 2.4.1 Notebook: Classification MNIST digits with a Multi-Layer Perceptron (MLP)

Assignment 4	MSE vs MAE

3 Convolutional Neural Networks

3.1 Image Classification

- 3.1.1 Notebook: Image Classification using Convolutional Neural Network (CNN)

3.2 Convolutional Neural Networks

- 3.2.1 CNN Building Blocks
- 3.2.2 The Convolution Operation
- 3.2.3 Layers in CNN
- 3.2.4 Kernels and Filters in CNNs (contd)

3.3 LeNet in PyTorch

- 3.3.1 Implementing LeNet in PyTorch
- 3.3.2 LeNet in BatchNorm

3.4 Evaluation Metrics for Classification

- 3.4.1 Performance Metrics for Classification
- 3.4.2 Evaluation Metrics using PyTorch
- 3.4.3 Evaluation Metrics using Torchmetrics

3.5 Important CNN Architectures

- 3.5.1 CNN architectures
- 3.5.2 Pretrained Classification Models in TorchVision

Assignment 5	Implement CNN For Image Classification On CIFAR10 Dataset

4 Deep Neural Networks

4.1 Optimizers and LR Schedulers

- 4.1.1 Optimizers
- 4.1.2 Notebook: PyTorch Optimizers
- 4.1.3 Learning Rate Schedulers in P

4.2 Training Deep Networks

- 4.2.1 Overview
- 4.2.2 Step 1: Data Understanding
- 4.2.3 Step 2: Data Preparation
- 4.2.4 Step 3: Check Training Pipeline
- 4.2.5 Step 4: Train the Model
- 4.2.6 Step 5: Improve the Model
- 4.2.7 Check training pipeline Hands-on

4.3 How to add robustness to a model?

- 4.3.1 Bias-variance tradeoff
- 4.3.2 How to prevent Overfitting?
- 4.3.3 Training with Regularization

4.4 Gaining Insights

- 4.4.1 Notebook: GradCam

Assignment 6	Adam Optimizer Implementation

Project 1	Implement An Image Classifier From Scratch

5 Best Practices in Deep Learning

5.1 Troubleshooting training with TensorBoard

- 5.1.1 TensorBoard overview
- 5.1.2 TensorBoard Dashboard
- 5.1.3 Notebook: Logging using TensorBoard
- 5.1.4 Notebook: Model Training using TensorBoard
- 5.1.5 Notebook: Activation Feature Maps Visualization

5.2 Transfer Learning and Fine-tuning in PyTorch

- 5.2.1 Fine-Tuning and Transfer Learning
- 5.2.2 Notebook: Transfer Learning ResNet on Cat-Dog-Panda Dataset
- 5.2.3 Notebook: Transfer Learning on ASL Data
- 5.2.4 Notebook: Fine-Tuning on ASL Data

5.3 Write your own Custom Dataset Class

5.3.1 Notebook: Custom Dataset Class in PyTorch

5.4 PyTorch Lighting

5.4.1 Notebook: Introduction to PyTorch Lightning

5.4.2 Notebook: Transfer Learning with Lighting

Project 2	Kaggle Competition - Classification

6 Segmentation

6.1 Introduction to Semantic Segmentation

6.1.1 Introduction to Semantic Segmentation

6.1.2 Semantic Segmentation Datasets

6.1.3 Semantic Segmentation Architecture

6.2 PyTorch Segmentation Models and Custom Dataset

6.2.1 Notebook: Torchvision Semantic Segmentation Models

6.2.2 Notebook: Introduction to Segmentation Datasets and Custom Dataset Class

6.3 Transposed Convolutions

6.3.1 Transposed Convolutions

6.4 Fully Convolutional Networks

6.4.1 FCN Architecture

6.4.2 Notebook: FCN on Road Data: CE Loss

6.4.3 Evaluation Metrics and Loss Functions

6.4.4 FCN: Custom Metrics and Loss Functions

6.5 UNet

6.5.1 UNet Architecture

6.5.2 UNet on Road Data: CE Loss

6.5.3 UNet on CamVid Data: Dice Loss

6.6 Dilated Convolution

6.6.1 Dilated Convolution

6.7 DeepLabV3

6.7.1 DeepLabv3 Architecture

6.7.2 DeepLabv3 on Road Data: Dice Loss

6.7.3 DeepLabv3 on CamVid Data: Dice Loss

6.7.4 Deeplabv3 on SUIM Data: Dice Loss

6.8 Instance Segmentation

6.8.1 Instance Segmentation using Mask RCNN

Assignment 7	UNet Implementation with ResNet50

Project 3	Kaggle Competition - Semantic Segmentation

7 Object Detection Fundamentals

7.1 Introductions to Object Detection

- 7.1.1 Object Detection Introduction
- 7.1.2 History of Object Detection
- 7.1.3 Object Detection Datasets

7.2 Hands-on with Object Detection

- 7.2.1 Notebook: Inference using Object Detection Models from Torchvision
- 7.2.2 Notebook: Training Object Detection Models

7.3 Classification to Detection

- 7.3.1 Image Classification vs Object Detection
- 7.3.2 Revisiting Classification Pipeline
- 7.3.3 Encoding Bounding Boxes using Anchors
- 7.3.4 Intersection over Union (IoU)
- 7.3.5 Notebook: Iou
- 7.3.6 Encoding of Ground Truth
- 7.3.7 Multiple Anchors

7.4 Non-Maximum Suppression (NMS)

- 7.4.1 Non-Maximum Suppression (NMS)
- 7.4.2 NMS vs Soft NMS
- 7.4.3 Notebook: NMS

7.5 Evaluation Metrics

- 7.5.1 Why do we need an Evaluation metric?
- 7.5.2 Building Block of mAP
- 7.5.3 Precision and Recall
- 7.5.4 Average Precision (AP) and Mean Average Precision (mAP)
- 7.5.5 Notebook: AP and mAP
- 7.5.6 Notebook: mAP using Torchmetrics

7.6 Popular Object Detection Architectures

- 7.6.1 Traditional Object Detectors
- 7.6.2 Two Stage Object Detectors
- 7.6.3 YOLO: You Only Look Once
- 7.6.4 SSD: Single Shot MultiBox Detectors
- 7.6.5 RetinaNet

8 Object Detection Fine-Tuning and Training

8.1 Understanding Faster RCNN

- 8.1.1 Notebook: Understanding Faster RCNN

8.2 Detectron 2

8.2.1 Notebook: Fine-Tuning RatinaNet using Detectron 2

8.3 Ultralytics YOLO

8.3.1 Notebook: Introduction to Ultralutics YOLO

8.3.2 Notebook: YOLOv8 Custom Training

8.4 Create a Custom Object Detector

8.4.1 Notebook: Detectors Architecture

8.4.2 Notebook: Anchors, Boxes and Label Encoding

8.4.3 Notebook: Anchors Generation using K-Means

8.4.4 Notebook: Decode and NMS

8.4.5 Notebook: Create a Custom Dataset Class

8.4.6 Notebook: Training from Scratch

Assignment 8	Focal Loss Implementation

Project 4	Object Detection

9 Introduction to Generative Adversarial Networks (GANs)

9.1 GANs

9.1.1 Notebook: Introduction to GANs

9.1.2 Notebook: Vanilla GAN using Fashion MNIST

9.1.3 Notebook: DCGAN using Flickr Faces

9.1.4 Notebook: CGAN using Fashion MNIST

10 Pose Estimation

10.1 DensePose

10.1.1 Introduction to DensePose

10.1.2 DensePose inference

10.1.3 DensePose Training

10.1.4 Applications: Squat Checker

Summary of Assessments

- **Total Projects: 4**
- **Total Assignments: 7**

Explore Other Courses

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

Advanced Training Concepts

|

Module 5

Semantic Segmentation

|

Module 6

Object Detection

|

Module 7

Introduction to Gans

|

Module 8

Introduction to Mediapipe and Applications

1 Getting Started

1.1 Introduction to Artificial Intelligence

- 1.1.1 History of AI
- 1.1.2 Application of AI
- 1.1.3 AI in Computer Vision
- 1.1.4 AI Terminology
- 1.1.5 Why is Deep Learning so popular?
- 1.1.6 Deep Learning Frameworks

1.2 Numpy Refresher

- 1.2.1 Notebook: Numpy Refresher Part-1
- 1.2.2 Notebook: Numpy Refresher Part-2
- 1.2.3 Notebook: Numpy Refresher Part-3

1.3 Assignment: Numpy (Ungraded assignment)

- 1.3.1 NumPy AssignmeNT

1.4 Introduction to TensorFlow

- 1.4.1 Notebook: TensorFlow Basics

1.5 Assignment A01: Tensorflow Assignment

- 1.5.1 Tensorflow Assignment

1.6 What's inside an ML Algorithm?

- 1.6.1 Machine Learning Pipeline
- 1.6.2 Solving ML Problems
- 1.6.3 Notebook: Gradient Descent for Optimization

1.7 Assignment A02: Gradient Descent Assignment

- 1.7.1 Gradient Descent Assignment

1.8 Quiz 1

- 1.8.1 Instructions & Grading Scheme
- 1.8.2 Quiz 1

2 Neural Networks

2.1 Understanding neural networks

- 2.1.1 Deep Learning Overview
- 2.1.2 What is a Neural Network
- 2.1.3 Feature Vectors and Normalization
- 2.1.4 Demystifying Neural Networks

2.2 Building Neural Networks in Keras

2.2.1 Notebook: Introduction to Linear Regression
2.2.2 Notebook: Auto-MPG Data Processing
2.2.3 Notebook: Linear Regression with Keras
2.2.4 Notebook: Binary Classification with Keras

2.3 Building Blocks of a Neural Network

2.3.1 Loss Function for Regression
2.3.2 Notebook: Regression Loss
2.3.3 Loss Functions for Classification
2.3.4 Notebook: Classification Losses
2.3.5 Types of Activation Functions
2.3.6 Notebook: Activation functions
2.3.7 How does the network learn?
2.3.8 Notebook: Backprop using Gradient Tape

2.4 Multi-Class Classification using Keras

2.4.1 Notebook: Classifying MNIST digits with a Multi-Layer Perceptron (MLP)

2.5 Assignment A03: Applying an MLP on the Fashion MNIST Dataset

2.5.1 MLP Assignment

2.6 Model Complexity, Generalization and Handling OverFitting

2.6.1 Bias-Variance Trade-off
2.6.2 How to Prevent Overfitting

2.7 Quiz 2

Assignment 3	**Applying an MLP on the Fashion MNIST Dataset**

3 Convolutional Neural Networks

3.1 Image Classification

3.1.1 Notebook: Image Classification using Convolutional Neural Network (CNN)

3.2 Convolutional neural networks

3.2.1 CNN Building Blocks
3.2.2 The Convolution Operation
3.2.3 Layers in CNN
3.2.4 Kernels and Filters in CNNs (contd)
3.2.5 Notebook: Implementing LeNet in Keras

3.3 Evaluation Metrics for Classification

3.3.1 Performance Metrics for Classification

3.3.2 Notebook: Evaluation Metrics for Classification

3.4 Assignment A04: Sequential vs Functional API

3.4.1 Assignment: Sequential vs Functional API

Assignment 4	Sequential vs. functional API

3.5 Building Models with Custom Data

3.5.1 Notebook: Keras Image_Dataset_from_Directory

3.5.2 Notebook: Overfitting and Data Augmentation

3.6 Working with pretrained Networks

3.6.1 Important CNN Architectures

3.6.2 Notebook: Pretrained Models from Keras applications

3.6.3 Notebook: Training VGGNet from Scratch on Balls Dataset

3.7 Transfer Learning and Fine-Tuning

3.7.1 Notebook: Transfer Learning with VGGNet as Feature Extractor on Balls Data

3.7.2 Notebook: Transfer Learning with VGGNet as Feature Extractor on ASL Data

3.7.3 Notebook: Fine-Tuning VGGNet using ASL Data

Assignment 5	Image Classification using CNN Assignment

3.8 Quiz 3

Project 1	Implement an Image Classifier from Scratch

4 Advanced Training Concepts

4.1 Optimizers

4.1.1 Optimizers

4.1.2 Notebook: Optimizers

4.2 Assignment A06: AdamW optimizer Implementation Assignment

4.2.1 AdamW Optimizer

Assignment 6	AdamW optimizer Implementation

4.3 Handling Data in Tensorflow

4.3.1 Notebook: Introduction to TF Data

4.3.2 Notebook: Custom Data Loader using Sequence Model

4.3.3 Notebook: TF Records

4.4 LR schedulers

4.4.1 Learning Rate Decay Methods

4.4.2 Notebook: LR Schedulers

4.5 Gaining Insights

4.5.1 Notebook: GradCam

Project 2	Kaggle Competition - Classification

5 Semantic Segmentation

5.1 Introduction to Semantic Segmentation

5.1.1 Introduction to Semantic Segmentation

5.1.2 Semantic Segmentation Datasets

5.1.3 Overview of Semantic Segmentation Architecture

5.2 Custom Data Loader

5.2.1 Notebook: Introduction to Segmentation Datasets and Custom Data Loader

5.3 Transposed Convolutions

5.4 Fully Convolutional Networks

5.4.1 FCN Architecture

5.4.2 FCN on Road Data: CE Loss

5.4.3 Evaluation Metrics in Semantic Segmentation

5.4.4 FCN: Custom Metrics and Loss Functions

5.5 UNet

5.5.1 UNet Architecture

5.5.2 UNet on Road Data: CE Loss

5.5.3 UNet on CamVid Data: Dice Loss

5.6 Dilated Convolution

5.7 DeepLabV3

5.7.1 DeepLabv3 Architecture

5.7.2 DeepLabv3+ on Road Data: CE Loss

5.7.3 DeepLabv3+ on CamVid Data: Dice Loss

5.7.4 DeepLabv3+ on CamVid Data: Best Results

5.7.5 DeepLabv3+ on SUIM Data: CE Loss

5.7.6 DeepLabv3+ on SUIM Data: Best Results

5.8 Segment Anything (SAM)

5.8.1 TF-SAM

Assignment 7	PSPNet

Project 3	Semantic Segmentation

6 Object Detection

6.1 Introduction to Object Detection

6.1.1 Object Detection Introduction

6.1.2 History of Object Detection

6.1.3 Object Detection Datasets

6.2 Hands-on with Object Detection

6.2.1 Inference using Object Detection Models from TensorflowHub

6.3 Classification to Detection

6.3.1 Image Classification vs Object Detection

6.3.2 Revisiting Classification Pipeline

6.3.3 Encoding Bounding Boxes using Anchors

6.3.4 IoU

6.3.5 Notebook: IoU

6.3.6 Encoding of Ground Truth

6.3.7 Multiple Anchors

6.4 Non-Maximum Suppression (NMS)

6.4.1 Non-Maximum Suppression (NMS)

6.4.2 NMS vs Soft NMS

6.4.3 Notebook: NMS

6.5 Evaluation Metrics

6.5.1 Why need an Evaluation metric?

6.5.2 Building Blocks of mAP

6.5.3 Precision and Recall

6.5.4 Average Precision (AP) and Mean Average Precision (mAP)

6.5.5 Notebook: AP and mAP

6.6 Popular Object Detection Architectures

6.6.1 Traditional Object Detectors

6.6.2 Two-Stage Object Detectors

6.6.3 YOLO: You Only Look Once

6.6.4 SSD: Single Shot Multi

6.6.5 RetinaNet

Assignment 8	Encoding of Ground Truths

6.7 TensorFlow Object Detection API

6.7.1 Installation of TFOD

6.7.2 Introduction and Inference using TFOD Pretrained Models

6.7.3 Data Preparation in TFOD

6.7.4 Pipeline Configuration in TFOD

6.7.5 Download Pretrained model and fine-tune

6.7.6 Inference with Fine-tuned Model

6.8 Object Detection KerasCV

6.8.1 KerasCV Introduction

6.8.2 Finetuning YOLOv8 KerasCV

6.9 Create a Custom Object Detector

6.9.1 Detector Architecture

6.9.2 Anchor Boxes and Label Encoding

6.9.3 Anchors Generation using K-Means

6.9.4 Loss Function

6.9.5 Decode and NMS

6.9.6 Evaluator in the Pipeline

6.9.7 Create a Custom Dataset Loader

6.9.8 Training from Scratch

Project 4	Object Detection

7 Introduction to Generative Adversarial Networks (GANs)

7.1 GANs

7.1.1 Notebook: Introduction to GANs

7.1.2 Notebook: Vanilla GAN using Fashion MNIST

7.1.3 Notebook: DCGAN using Flickr Faces

7.1.4 Notebook: CGAN using Fashion MNIST

8 Introduction to Mediapipe and Applications

8.1 Applications

8.1.1 Introduction to Mediapipe

8.1.2 Posture Analysis using Mediapipe

8.1.3 Drowsy Driver Detection using Mediapipe

Summary of Assessments

- **Total Projects: 4**
- **Total Assignments: 8**

Explore Other Courses

Course 5 - TxAP

Advanced Vision Applications with Deep Learning & Transformers

Module 1

Neural Networks and Classification

|

Module 2

Object Detection

|

Module 3

Text Detection & Recognition (OCR)

|

Module 4

Segmentation

|

Module 5

Tracking

|

Module 6

Keypoint Estimation

|

Module 7

Face Recognition and Applications

|

Bonus Module

CLIP & Vision Language Models

1. Neural Networks and Classification

1.1. Introduction to Neural Networks

1.1.1. What is a Neural Network?

1.1.2. Deep Learning Frameworks

1.1.3. How Does a Neural Network Learn?

1.2. Introduction to PyTorch

1.2.1. What is PyTorch and Why Choose PyTorch?

1.2.2. PyTorch Basics and Tensor Operators

1.2.3. Training Neural Networks: Overview

1.2.4. Binary Classification with PyTorch

1.2.5. Dataloaders in PyTorch: Classification

1.3. Feedforward Neural Networks

1.3.1. Why do we need Hidden layers?

1.3.2. Training a Neural Network using Backpropagation

1.3.3. Image Classification using MLP

1.3.4. Example: Image Classification using MLP

1.4. Convolution Neural Networks

1.4.1. The Convolution Operation

1.4.2. Example: Image Classification using CNNs

1.5. Logging using MLOps Tools

1.5.1. Introduction to Weights & Biases

1.6. Transfer Learning and Fine-Tuning using Pre-Trained Models

1.6.1. Transfer Learning Notebook

1.6.2. Train Image Classifier to Fine-Tune on a Difficult Dataset

1.7. Vision Transformers

1.7.1. Introduction to Vit

1.7.2. Vit Attention Map

1.7.3. Fine-Tuning Vit for Birds Classification

Assignment 1	**Binary Classification with CNN**

2. Object Detection

2.1. Introduction

2.1.1. Object Detection Overview

2.1.2. Traditional Object Detection Pipeline

2.2. Single-Stage Detectors

2.2.1. SSD

2.2.2. RetinaNet

2.3. Object Detection using YOLO

2.3.1. Introduction to YOLO

2.3.2. Object Detection Inference

2.3.3. Fine-Tuning for Aerial Images

2.3.4. Tiled Object Detection using YOLOv8

2.4. Detection Transformers

2.4.1. Object Detection using RTDETR

Case Study	**Global Wheat Detection Part 1 - Training** **Global Wheat Detection Part 2 - Practical Improvements**

Assignment 2	**Annotation Conversion**

Project 1	**Object Detection**

3. Text Detection & Recognition (a.k.a OCR)

3.1. Graphic Text Recognition using Tesseract

3.1.1. What is Tesseract?

3.1.2. Notebook: Introduction to OCR using Tesseract

3.1.3. Notebook: Tesseract OCR Failure Cases

3.1.4. Notebook: Improving Tesseract OCR Failures

3.2. Transformer OCR

3.2.1. Introduction to TrOCR

3.2.2. TrOCR Inference with Cropped Image

3.2.3. TrOCR Inference with Text Detection

3.2.4. Fine-Tuning using TrOCR on Captcha

3.3. Application - ALPR

3.3.1. Introduction to ALPR

3.3.2. YOLOv10 License Plate Detection Training

3.3.3. Fine-Tuning TrOCR on License Plate Text

3.3.4. ALPR Inference - Combined Detection + OCR

Assignment 3	**TrOCR Invoice**

Project 2	TrOCR

4. Segmentation

4.1. Deep Learning-Based Segmentation Model

4.1.1. Introduction to Segmentation using Torchvision Models

4.2. Transformer-Based Segmentation Models

4.2.1. Aerial View Segmentation using SegFormer

4.3. SAM (Segment Anything Model)

4.3.1. Introduction to SAM

4.3.2. Automatic Person Segmentation with YOLOv11 + SAM2

4.3.3. Advanced Segmentation Techniques with SAM2

4.4. Segmentation using Custom Backbone

4.4.1. DINO UNet Road Segmentation

Project 3	Segmentation

5. Tracking

5.1. Introduction

5.1.1. Object Tracking Algorithm

5.2. Ultralytics Tracking

5.2.1. Tracking using Ultralytics YOLOv8

5.2.2. SeaDrone Dataset Detection Fine-Tuning and Tracking

5.2.3. YOLO11 Grocery Cart Tracking and Counting

5.3. Multi-Camera Tracking

5.3.1. Multi-Camera Tracking using OpenVINO

5.4. Re-Identification

5.4.1. Person ReID Fine-Tuning

5.5. Transformer-Based Tracker

5.5.1. Introduction to Point Tracking using CoTracker3

6. Keypoint Estimation

6.1. Introduction

6.1.1. Introduction to Pose Estimation and Landmark Detection

6.2. Exercise Analysis using Key Points

6.2.1. Squats Analysis Pipeline Overview

6.2.2. Squats Analysis Code Explanation

6.3. Keypoint Fine Tuning using YOLOv8

6.3.1. Facial Keypoint Fine Tuning using YOLOv8

6.4. Applications of Keypoint

6.4.1. Warping a Triangle

6.4.2. Delaunay Triangulation

6.4.3. Face Alignment

6.4.4. Face Averaging

6.4.5. Face Morphing

6.4.6. Bug Eyes

6.4.7. Face Swap

6.4.8. Beard Filter

6.4.9. Aging Filter

Assignment 4	Smile Detection

7. Face Recognition and Applications

7.1. Creating an Automated Attendance System using AWS

7.1.1. Introduction to Automated Attendance System

7.1.2. Introduction to AWS Rekognition

7.1.3. How to Set Up AWS with AWS Rekognition

7.1.4. Registration using AWS Rekognition

7.1.5. Deregistration using AWS Rekognition

7.1.6. Attendance Using AWS Rekognition

7.1.7. Gradio App Setup Instruction

7.2. Custom Face Verification System

7.2.1. Introduction and Setup

7.2.2. Face Verification Pipeline

Assignment 5	Doppelganger

8. Bonus Module

8.1. Clip

8.1.1. Zero Short Image Classification with CLIP

8.2. Vision Language Models (VLM)

8.2.1. Image Captioning and VAQ with Moondream2

8.2.2. Paligemma2-Mix-VLM

Summary of Assessments

- **Total Projects: 3**
- **Total Assignments: 5**

Explore Other Courses

Course 6 - GENAI

AI Art Generation for Everyone

Module 1

Introduction to Generative AI

|

Module 2

Generative AI with OpenAI DALL-E

|

Module 3

Generating Images with MidJourney

|

Module 4

FREE GPU Credits from RunPod

|

Module 5

Generating Image with Stable Diffusion

|

Module 6

Diving Deeper into Stable Diffusion

|

Module 7

Advanced Image Editing Techniques

1. **Introduction to Generative AI**
   - 1.1. **What is Generative AI?**
     - 1.1.1. What is Generative AI?
     - 1.1.2. Generative AI Landscape
     - 1.1.3. Evolution of Generative AI
     - 1.1.4. Ethics and Guidelines
2. **Facial Landmark Detection using Dlib**
   - 2.1. **Generative AI with OpenAI DALL-E**
     - 2.1.1. OpenAI Intro
     - 2.1.2. DALLE2 and ChatGPT
     - 2.1.3. Image Editing using DALLE2
     - 2.1.4. DALLE2 Alternative
3. **Generating Images with midJourney**
   - 3.1. **Generating Images with Stable Diffusion**
     - 3.1.1. Introduction to MidJourney
     - 3.1.2. Setting up MidJourney Discord
     - 3.1.3. Understanding MidJourney parameters for better Prompting
     - 3.1.4. Blending multiple images
     - 3.1.5. Advanced Features in MidJourney
     - 3.1.6. MidJourney updates
   - 3.2. **MidJourney Alternative**
4. **FREE GPU Credits from RunPod**
   - 4.1. **What is RunPod?**
   - 4.2. **Redeem your RunPod Coupon**
5. **Generating Images with Stable Diffusion**
   - 5.1. **Introduction to Stable Diffusion**
     - 5.1.1. Introduction to Stable Diffusion
     - 5.1.2. Demystifying Stable Diffusion Model
     - 5.1.3. Getting Started with Stable Diffusion
   - 5.2. **Generating Images with DreamStudio**
     - 5.2.1. Generating Images Using DreamStudio
     - 5.2.2. Editing Images with DreamStudio
   - 5.3. **Setting up Automatic1111 Stable Diffusion WebUI**
     - 5.3.1. What is the Stable Diffusion WebUI?
     - 5.3.2. Using Automatic1111 WebUI on RunPod
     - 5.3.3. Using Automatic1111 WebUI on Kaggle
   - 5.4. **Generating Images with Stable Diffusion WebUI**
     - 5.4.1. Generating Images with Stable Diffusion WebUI
     - 5.4.2. Prompt Engineering with Stable Diffusion

6. Diving Deeper into Stable Diffusion

6.1. Advanced Controls in SD WebUI

- 6.1.1. Analyzing Prompts with Prompts Matrix in WebUI
- 6.1.2. Prompt Strength and Weights
- 6.1.3. Prompt Editing and Blending
- 6.1.4. Using the XYZ Plots in Stable Diffusion WebUI

6.2. Editing Image using img2img

- 6.2.1. Image Editing with img2img in WebUI
- 6.2.2. Image Inpainting in WebUI: Part 1
- 6.2.3. Image Inpainting in WebUI: Part 2
- 6.2.4. Image Inpainting img2img sketch
- 6.2.5. Image Editing with Inpaint Sketch

6.3. Where to find Models?

- 6.3.1. Using Fine-Tuned models from CIVITAI

6.4. Using Extensions with WebUI

- 6.4.1. How to Install Extensions
- 6.4.2. WebUI Extensions - Part II

7. Advanced Image Editing Techniques

7.1. Controlled Editing with InstructPix2Pix

- 7.1.1. InstructPix2Pix Introduction & Setup
- 7.1.2. InstructPix2Pix Demo - Part 1 (Virtual Try-On)
- 7.1.3. InstructPix2Pix Demo - Part 2

7.2. ControlNet: Controlling Stable Diffusion Models

- 7.2.1. Introduction to ControlNet
- 7.2.2. ControlNet installation & Setup
- 7.2.3. Generating Image Variation with ControlNet Models
- 7.2.4. ControlNet OpenPose

7.3. UpScalling Image Like a Pro!

- 7.3.1. UpScaling Image with Ultimate SD UpScale + ControlNet

Explore Other Courses

Advanced AI Art Generation

Module 1

DreamBooth

|

Module 2

Stable Diffusion WebUI Practical Tips for Efficient Workflow

|

Module 3

Controlled Image Editing

|

Module 4

Textual Inversion

|

Module 5

LoRA

|

Module 6

Creating videos using Prompt

|

Module 7

Prompt Generating using ChatGPT

|

Module 8

Image Generation Using FLUX

1. **DreamBooth**
   - **1.1. Fine-Tuning SD Models with DreamBooth**
     - 1.1.1. Introduction to DreamBooth
     - 1.1.2. RunPod Configuration for DreamBooth
     - 1.1.3. Fine-tuning SD with DreamBooth
     - 1.1.4. Fine-tuning SD with DreamBooth + Captions
2. **Stable Diffusion WebUI Practical Tips for Efficient Workflow**
   - **2.1. Stable Diffusion WebUI Installation**
     - 2.1.1. Stable Diffusion WebUI MacOS Installation
     - 2.1.2. Stable Diffusion WebUI Windows Installation
     - 2.1.3. Stable Diffusion WebUI Linux Installation
   - 2.2. **How to Recreate Existing Images Using Stable Diffusion**
   - **2.3. Stable Diffusion WebUI Key Settings**
   - **2.4. Understanding XYZ Plots in Stable Diffusion WebUI**
   - **2.5. Understanding Variational Strength Feature in Stable Diffusion**
3. **Controlled Image Editing**
   - **3.1. Asset placement using Paint By Example**
     - 3.1.1. Paint By Example
     - 3.1.2. Setup Paint By Example
     - 3.1.3. Understanding MidJourney parameters for better Prompting
     - 3.1.4. Creating Training Data for Deepfakes
     - 3.1.5. Using the Trained Model
     - 3.1.6. Tips and Tricks for better Results
4. **Textual Inversion**
   - **4.1. Extending SD with Embeddings**
     - 4.1.1. Stable Diffusion Architecture
     - 4.1.2. Textual Inversion Architecture
     - 4.1.3. Using Pretrained Embeddings Part-I
     - 4.1.4. Using Pretrained Embeddings Part-II
     - 4.1.5. Training Custom Style Embeddings
     - 4.1.6. Evaluating Custom Style Embeddings
     - 4.1.7. Using Custom Style Embeddings
5. **LoRA**
   - **5.1. Fine-Tuning SD Models with LoRA**
     - 5.1.1. Kohya Installation on RunPod
     - 5.1.2. LoRA Training in Kohya GUI
     - 5.1.3. LoRA Training & Execution
     - 5.1.4. Using LoRA Models
6. **Creating videos using Prompt**

6.1.1. Introduction to Video Generation
6.1.2. An Overview of TEXT2VIDEO Methods
6.1.3. IMG2VIDEO using RunwayML GEN-2
6.1.4. Methods to perform Video2Video
6.1.5. MOV2MOV using ControlNet - using Automatic 1111

7. Prompt Generation using ChatGPT

7.1. Training ChatGPT for Prompt Generation

8. Image Generation using FLUX

8.1. FLux Image Generation

Explore Other Courses