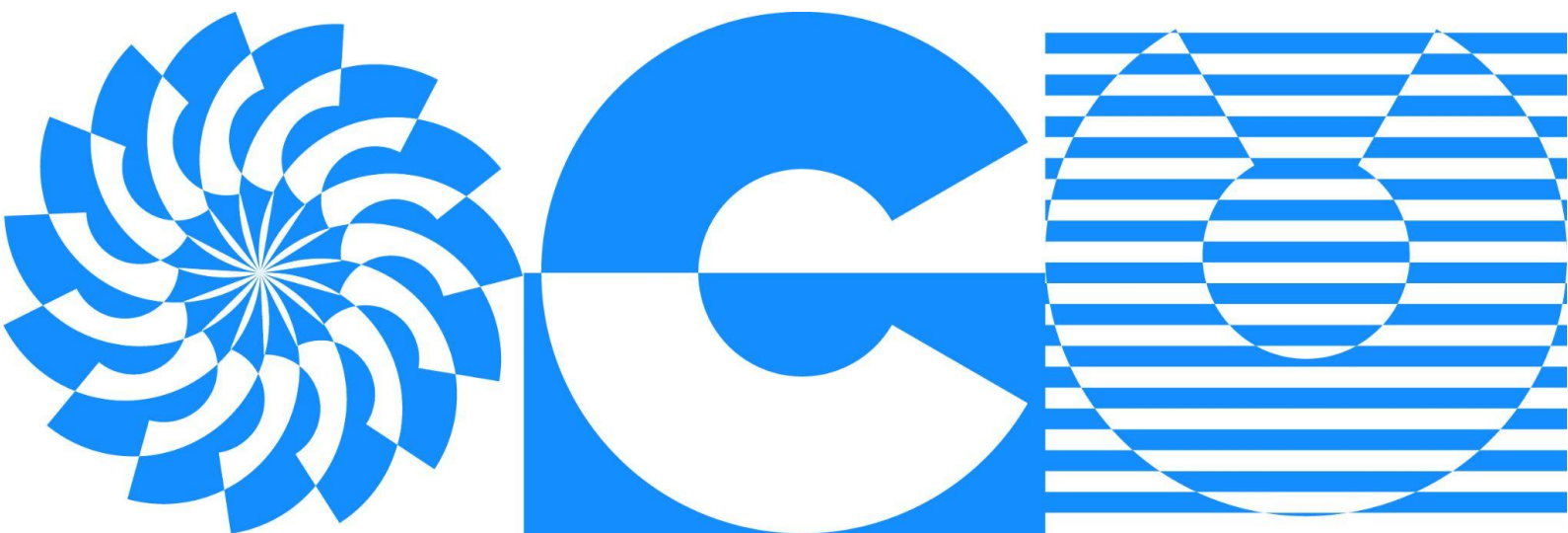


A place where legacy creates future.



DLAP

# **Computer Vision & Deep Learning Applications**

[Detailed Curriculum](#)



## Index

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

The course is being updated and a new version of the course is planned to be released by July 31 2023. You will get access to both the versions of the courses. The syllabus mentions the updates marked in red in the relevant areas.

## **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**

### 3 Snapshot Filters

#### 3.1 Face Swap

- 3.1.1 How can Face Swapping be achieved?
- 3.1.2 Seamless Cloning
- 3.1.3 Seamless Cloning in OpenCV
- 3.1.4 Face Swapping in a video

#### 3.2 Application: Beard Filter

#### 3.3 Application: Aging Filter

#### 3.4 Non-linear Deformations

- 3.4.1 Moving Least Squares
- 3.4.2 Application: Happify and Fatify filters on face

Project 1	Build a Virtual Makeup Application
-----------	------------------------------------

### 4 Face Recognition

- 4.1 Introduction to Face Recognition
- 4.2 Eigen Faces
- 4.3 Fisher Faces
- 4.4 Local Binary Patterns Histograms
- 4.5 Face Recognition API in OpenCV
- 4.6 Deep Learning-based Face Recognition [To be updated with latest models]

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

### 5 Introduction To Deep Learning

#### 5.1 Basics of Neural Networks

- 5.1.1 What is a Neural Network?
- 5.1.2 How do Neural Networks learn?

#### 5.2 Introduction to Keras [To be updated to The PyTorch framework]

- 5.2.1 Deep Learning frameworks
- 5.2.2 The Keras Framework
- 5.2.3 Linear Regression using Keras

#### 5.3 Convolutional Neural Network

- 5.3.1 What is a Convolutional Neural Network (CNN)?
- 5.3.2 Example: Image Classification using CNN
- 5.3.3 Data Augmentation (Python)

#### 5.4 Transfer Learning and Fine-tuning and Logging

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

<b>Assignment 2</b>	<b>Improve CNN Performance during Training</b>
---------------------	--

## 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

<b>Project 3</b>	<b>Train A Face Mask Detector using YOLO</b>
------------------	--

## 7 Text Detection And Recognition

- 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)**



**7.5 Inference using Transformer OCR (TrOCR)**

**7.6 Fine-Tuning Transformer OCR (TrOCR)**

**7.7 Case Study - ALPR (YOLO NAS + TrOCR)**

<b>Assignment 3</b>	<b>Build an app to perform OCR On Invoices</b>
---------------------	--

## **8 Deploy Applications On Cloud**

### **8.1 Create a web application using Flask**

- 8.1.1 What is Flask?
- 8.1.2 A minimal Flask application
- 8.1.3 Using HTML templates
- 8.1.4 A complete Flask application

### **8.2 Deploy a web application on Heroku [Paas]**

- 8.2.1 What is Heroku?
- 8.2.2 How to create an account on Heroku?
- 8.2.3 Prepare application for deployment
- 8.2.4 How to deploy using Heroku CLI?
- 8.2.5 How to deploy using Heroku website?

### **8.3 Deploy a web application on Google GCP [Iaas]**

- 8.3.1 What is Google Cloud Platform (GCP)?
- 8.3.2 Create account on Google Cloud Platform
- 8.3.3 Create and configure a VM instance
- 8.3.4 Setup VM and deploy app
- 8.3.5 Change firewall settings and check deployment

<b>Assignment 4</b>	<b>Deploy Your Web App On Heroku</b>
---------------------	--------------------------------------

---

[Explore Other Courses](#)