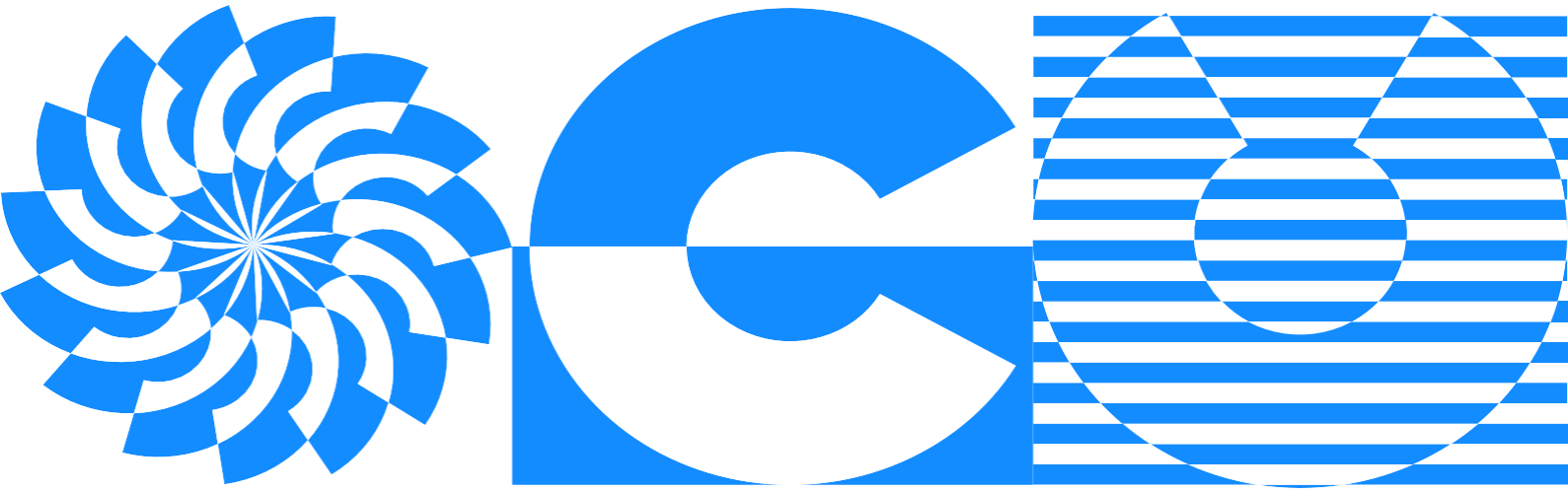
A place where legacy creates future.



DLPT2.X

Deep Learning with PyTorch 2.x

Detailed Curriculum

OpenCV University

DEEP LEARNING WITH PYTORCH 2.X

Index

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-I

1.2.2 Notebook: NumPy Refresher Part-II

1.2.3 Notebook: NumPy Refresher Part-III

Assignment: NumPy

1.3 Introduction to PyTorch

1.3.1 Why PyTorch

1.3.2 Notebook: PyTorch Basics

Assignment: 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 Optimization

Assignment: Gradient Descent

QUIZ 1

2 Neural Networks

2.1 Understanding Neural Networks

2.1.1 Deep Learning Overview

2.1.2 What is Neural Network

2.1.3 Feature Vectors and Normalization

2.1.4 Demystifying Neural Network

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 Notebook: Linear Regression with PyTorch

2.2.4 Notebook: Binary Classification with PyTorch

2.3 Building Blocks of a Neural Network

2.3.1 Notebook: Implementing MLP with 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: Classification Losses

2.3.6 Activation Functions

2.3.7 Notebook: Activation Functions

2.3.8 How does Neural Network Learn

2.4 Multi-Class Classification using PyTorch

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

Assignment: MSE vs MAE

Quiz 2

3 Convolutional Neural Networks

3.1 Image Classification

3.1.1 Notebook: Image Classification using 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

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: Implement a CNN for Image Classification on CIFAR10 Dataset

Quiz 3

4 Practical Considerations for Training 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

4.1.4 Notebook: LR Schedulers in PyTorch

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 Notebook : Check Training Pipeline Hands-on

4.3 Using your own Data

4.3.1 Notebook: DataLoader with ImageFolder

4.4 Model Complexity, Generalization and Handling Overfitting

4.4.1 Bias-Variance Tradeoff

4.4.2 How to Prevent Overfitting

4.4.3 Notebook: Training with Regularization

4.5 Gaining Insights

4.5.1 Notebook: GradCam

Assignment: Adam Optimizer Implementation

Quiz 4

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 Lightning

5.4.1 Notebook: Introduction to PyTorch Lightning

5.4.2 Notebook: Transfer Learning with Lightning

QUIZ 5

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 Dataset 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 in Semantic Segmentation

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

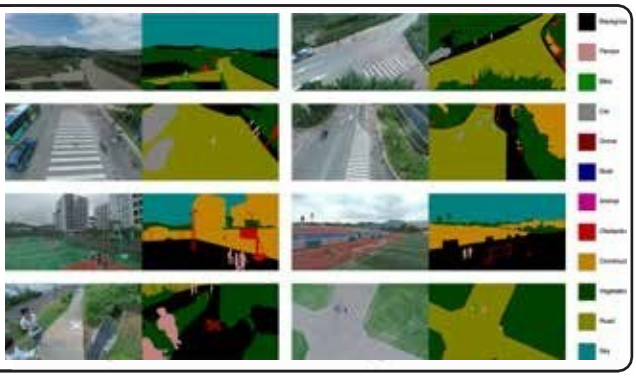
6.8 Instance Segmentation

6.8.1 Instance Segmentation using Mask RCNN

Assignment: UNet Implementation with ResNet50

QUIZ 6

Project 3:
Kaggle Competition: Semantic Segmentation



7 Object Detection Fundamentals

7.1 Introduction 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

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 need an Evaluation Metric

7.5.2 Building Blocks 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 Architecture

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 Detector

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 RetinaNet using Detectron 2

8.3 Ultralytics YOLO

8.3.1 Notebook: Introduction to Ultralytics YOLO

8.3.2 Notebook: YOLOv8 Custom Training

8.4 Create a Custom Object Detector

8.4.1 Notebook: Detector Architecture

8.4.2 Notebook: Anchor Boxes and Label Encoding

8.4.3 Notebook: Anchors Generation using K-Means

8.4.4 Notebook: Loss Function

8.4.5 Notebook: Decode and NMS

8.4.6 Notebook: Create a Custom Dataset Class

8.4.7 Notebook: Training from Scratch

Assignment: Focal Loss Implementation

QUIZ 7

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