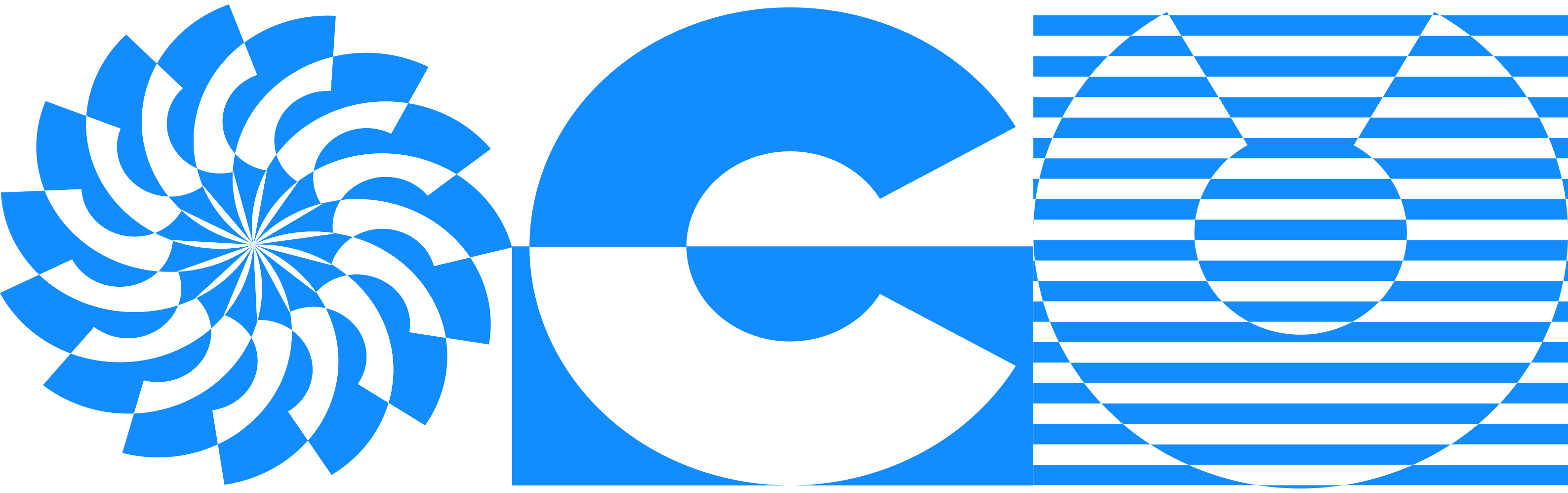
A place where legacy creates future.



DLTK

Deep Learning with TensorFlow & Keras

Detailed Curriculum

OpenCV University

DEEP LEARNING WITH TENSORFLOW & KERAS

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 Why is Deep Learning so popular

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

1.3 Introduction to TensorFlow

1.3.1 Notebook: TensorFlow Basics

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

Assignment1: NumPy

Assignment2: TensorFlow

Assignment3: 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 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 Neural 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)

Assignment4: Applying a MLP on the Fashion MNIST Dataset

2.5 Model Complexity, Generation and Handling OverFitting

2.5.1 Bias Variance Trade-off

2.5.2 How to Prevent Overfitting

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

3.2.5 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 Building Models with Custom Data

3.4.1 Notebook: Keras Image_Dataset_from_Directory

3.4.2 Notebook: Overfitting and Data Augmentation

3.5 Working with pretrained Networks

3.5.1 Important CNN Architectures

3.5.2 Notebook: Pretrained Models from Keras applications

3.5.3 Notebook: Training VGGNet from scratch on Balls Dataset

3.6 Transfer Learning and Fine Tuning

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

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

3.6.3 Notebook: Fine Tuning VGGNet using ASL Data

Assignment5: Sequential vs Functional API

Assignment6: Image Classification using CNN

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 Handling Data in TensorFlow

4.2.1 Notebook: Introduction to TF Data

4.2.2 Notebook: Custom Data Loader using Sequence Model

4.2.3 Notebook: TF Records

4.3 LR schedulers

4.3.1 Learning Rate Decay Methods

4.2.2 Notebook: LR Schedulers

4.4 Gaining Insights

4.4.1 Notebook: GradCam

Assignment7: AdamW Optimizer Implementation

Project 2:

Kaggle Competition:

Classification

5 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

5.3.1 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

5.6.1 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)

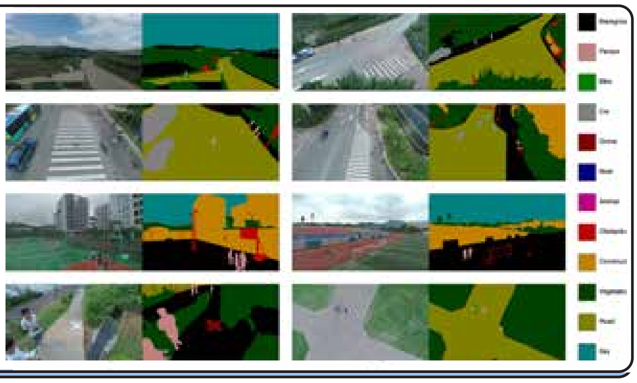
5.8.1 TF-SAM

Assignment8: PSPNet

QUIZ 4

Project 3:

Kaggle Competition: 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 Notebook: 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 Intersection over Union (IoU)

6.3.5 Notebook: IoU

6.3.6 Encoding of Ground Truth

6.3.7 Multiple Enchors

6.4 Non Maximum Supression

6.4.1 Non Maximum Supression (NMS)

6.4.2 NMS vd 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 Architecture

6.6.1 Traditional Object Detectors

6.6.2 Two Stage Objct Detectors

6.6.3 YOLO: You Only Look Once

6.6.4 SSD: Single Shot MultiBox Detector

6.6.5 RetinaNet

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 Finetune

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 Custon Object Detector

6.9.1 Detector Architecture

6.9.2 Anchor Boxes and Label Encoding

6.9.3 Anchor 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

Assignment9: Encoding of Ground Truths

QUIZ 5

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 Meduapipe and Application

8.1 Application

8.1.1 Introduction to Mediapipe

8.1.2 Posture Analysis using Mediapipe

8.1.3 Drowsy Driver Detection using Mediapipe