# Lab Week 8: Deep Learning for Computer Vision

In this lab, there are three tasks (two main task and an extension task). They require you to work through code examples. Your aim is to understand the examples and look up any procedures that are unclear. Alternatively, you can also speak to your lab tutor.

The libraries that we are using for deep learning in this module are Tensorflow and Keras. However, the workflow with other libraries such as PyTorch is high similar. Therefore, even though we are sticking to the set choice of libraries, if you wish to use another in the future your skills should transfer very easily.

# Preparation

The first task in this lab makes use of the book: **Python Machine Learning by Example, Liu, Y., (2020),** by Packt Publishing.

You can access the digital version of this book by going to the module’s sidebar, clicking on Resource-Lists and then you should be able to see the E-Book version.

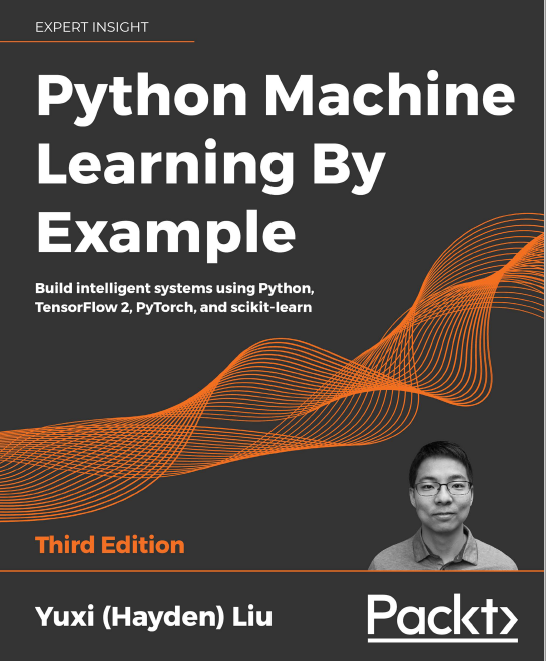


Figure Python Machine Learning by Example: Book

You should then open the book in a new window or tab, as you will be referring back to this.

**Download** all the files for this weeks labs and extract the content of the *.zip* file into the same directory.

You should also ensure that you have (or know how to) installed the required Python libraries in the Anaconda Environment:

* tensorflow (conda install tensorflow)
* albumentations (pip install albumentations)

Alternatively, you can also download and import the lab environment on GCU learn located in the Weeks/Units 🡪 Lab Setup Tutorial Section 🡪 *lab\_environment\_dl.yml*

# Task 1: Categorizing Images of Clothing with Convolutional Neural Networks

The first task is based on the book chapter 12 of the Book: **Python Machine Learning by Example, Liu, Y., (2020)**.

This will cover the basic building blocks of CNNs, how to train them and how you can implement data augmentation.

The accompanying code file is*: “Lab Week 8 Task 1.ipynb”*

***Exercise 1:***

Work through Chapter 12 of the abovenamed book.

The code has been provided in the accompanying code file, although you are still required to work on understanding this code. The book will have explanations but you are required to look up anything unfamiliar to you.

# Task 2: Custom Training Loop on a Horse-Dog Image Dataset and Convolutional Neural Networks



Figure Example Images of the Horse-Dog image dataset

The second task is very similar to the first in that it follows a similar scheme to obtaining the data, preparing the network and applying it with/without augmentation.

There is no book chapter accompanying this. The core-code is located in the file: *“Lab Week 8 Task 2.ipynb”*

The main point of this task is to show you a different way to approach the image classification problem by using a custom training loop, which also illustrates well how TensorFlow’s “*model.fit()”* works.

Additionally, it also shows how you can implement different data augmentation schemes to increase the data available.

***Exercise 2:***

Work through the file “*Lab Week 8 Task 2.ipynb”*similarly to the previous task. Once you understand the code you can try to make some changes such as changing the model architecture or going form a horse-dog dataset to a dataset of “trucks-airplanes” for example.

***Exercise 3:***

Read the guide on how to create an augmentation pipeline from Albumentations [here](https://albumentations.ai/docs/examples/example/). Then add another sample augmentation such as *RandomBrighntessContrast*. If you retrain the models, does this change the performance?

***Exercise 4:***

Try to use a larger model for training. For example, instead of only having a single convolutional layer, try using two with ReLU activations in-between. This offers better feature extraction capabilities. Check if this improves performance.

# Extra Task: Semantic Segmentation using U-Net

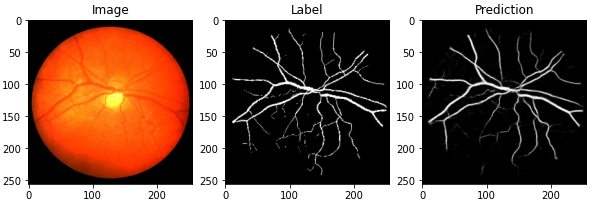


Figure Example Semantic segmentation with an image, the ground truth label and a sample prediction of U-Net

This extra task contains code for a neural network that performs semantic segmentation.

Here, a small medical dataset of retina vessels called [CHASEDB1](https://blogs.kingston.ac.uk/retinal/chasedb1/) is used.

Semantic segmentation is the task of assigning a class label to every pixel in an image. For this use case, we have images of a retina, which is the part of the eye that contains the light-sensitive cells. The goal is to find all the pixels, that belong to blood vessels. This can aid specialists in the diagnosis and monitoring of issues within the eye.

For this extract task, simply run through the code cells, try to understand their content by looking at the comments and execute them.