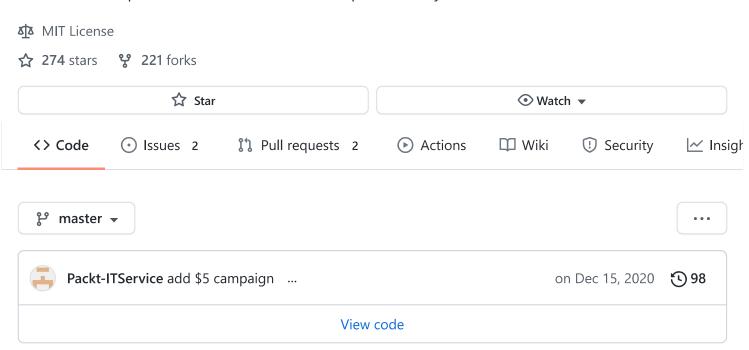
PacktPublishing / Hands-On-Computer-Vision-with-TensorFlow-2

Hands-On Computer Vision with TensorFlow 2, published by Packt



\$5 Tech Unlocked 2021!

Buy and download this Book for only \$5 on PacktPub.com

If you have read this book, please leave a review on Amazon.com. Potential readers can then use your unbiased opinion to help them make purchase decisions. Thank you. The \$5 campaign runs from **December 15th 2020** to **January 13th 2021**.

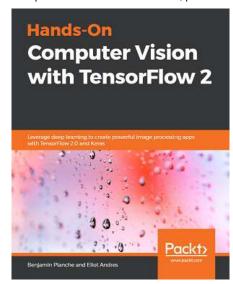
Hands-On Computer Vision with TensorFlow 2

Leverage deep learning to create powerful image processing apps with TensorFlow 2.0 and Keras

This is the code repository for Hands-On Computer Vision with TensorFlow 2 by Benjamin Planche and Eliot Andres, published by Packt.

This book is a practical guide to building high performance systems for object detection, segmentation, video processing, smartphone applications, and more. It is based on TensorFlow 2, the new version of Google's open-source library for machine learning.

This repository offers several notebooks to illustrate each of the chapters, as well as the complete sources for the advanced projects presented in the book. Note that this repository is meant to complement the book. Therefore, we suggest to check out its content for more detailed explanations and advanced tips.





What is this book about?

Computer vision solutions are becoming increasingly common, making their way in fields such as health, automobile, social media, and robotics. This book will help you explore TensorFlow 2, the brand new version of Google's open source framework for machine learning. You will understand how to benefit from using convolutional neural networks (CNNs) for visual tasks.

Hands-On Computer Vision with TensorFlow 2 starts with the fundamentals of computer vision and deep learning, teaching you how to build a neural network from scratch. You will discover the features that have made TensorFlow the most widely used AI library, along with its intuitive Keras interface, and move on to building, training, and deploying CNNs efficiently. Complete with concrete code examples, the book demonstrates how to classify images with modern solutions, such as Inception and ResNet, and extract specific content using You Only Look Once (YOLO), Mask R-CNN, and U-Net. You will also build Generative Adversarial Networks (GANs) and Variational Auto-Encoders (VAEs) to create and edit images, and LSTMs to analyze videos. In the process, you will acquire advanced insights into transfer learning, data augmentation, domain adaptation, and mobile and web deployment, among other key concepts. By the end of the book, you will have both the theoretical understanding and practical skills to solve advanced computer vision problems with TensorFlow 2.0.

This book covers the following exciting features:

- Create your own neural networks from scratch
- Classify images with modern architectures including Inception and ResNet
- Detect and segment objects in images with YOLO, Mask R-CNN, and U-Net

- Tackle problems in developing self-driving cars and facial emotion recognition systems
- Boost your application's performance with transfer learning, GANs, and domain adaptation
- Use recurrent neural networks for video analysis
- Optimize and deploy your networks on mobile devices and in the browser

If you feel this book is for you, get your copy today!



Instructions and Navigation

If you're new to deep learning and have some background in Python programming and image processing, like reading/writing image files and editing pixels, this book is for you. Even if you're an expert curious about the new TensorFlow 2 features, you'll find this book useful. While some theoretical explanations require knowledge in algebra and calculus, the book covers concrete examples for learners focused on practical applications such as visual recognition for self-driving cars and smartphone apps.

The code is in the form of Jupyter notebooks. Unless specified otherwise, it is running using Python 3.5 (or higher) and TensorFlow 2.0. Installation instructions are presented in the book (we recommend Anaconda to manage the dependencies like numpy, matplotlib, etc.).

As described in the following subsections, the provided Jupyter notebooks can either be studied directly or can be used as code recipes to run and reproduce the experiments presented in the book.

We also provide a PDF file that has color images of the screenshots/diagrams used in this book. Click here to download it.

Study the Jupyter notebooks online

If you simply want to go through the provided code and results, you can directly access them online in the book's GitHub repository. Indeed, GitHub is able to render Jupyter notebooks and to display them as static web pages. However, the GitHub viewer ignores some style formatting and interactive content. For the best online viewing experience, we recommend using instead Jupyter noviewer (https://nbviewer.jupyter.org), an official web platform you can use to read Jupyter notebooks uploaded online. This website can be queried to render notebooks stored in GitHub repositories. Therefore, the Jupyter notebooks provided can also be read at the following address: https://nbviewer.jupyter.org/github/PacktPublishing/Hands-On-Computer-Vision-with-TensorFlow-2.

Run the Jupyter notebooks on your machine

To read or run these documents on your machine, you should first install Jupyter Notebook. For those who already use Anaconda (https://www.anaconda.com) to manage and deploy their Python environments (as we will recommend in this book), Jupyter Notebook should be directly available (as it is installed with Anaconda). For those using other Python distributions and those not familiar with Jupyter Notebook, we recommend having a look at the documentation, which provides installation instructions and tutorials (https://jupyter.org/documentation).

Once Jupyter Notebook is installed on your machine, navigate to the directory containing the book's code files, open a terminal, and execute the following command:

\$ jupyter notebook

The web interface should open in your default browser. From there, you should be able to navigate the directory and open the Jupyter notebooks provided, either to read, execute, or edit them.

Some documents contain advanced experiments that can be extremely compute-intensive (such as the training of recognition algorithms over large datasets). Without the proper acceleration hardware (that is, without compatible NVIDIA GPUs, as explained in Chapter 2, *TensorFlow Basics and Training a Model*), these scripts can take hours or even days (even with compatible GPUs, the most advanced examples can take quite some time).

Run the Jupyter notebooks in Google Colab

مر سمماناسم

For those who wish to run the Jupyter notebooks themselves—or play with new experiments—but do not have access to a powerful enough machine, we recommend using **Google Colab**, also named Colaboratory (https://colab.research.google.com). It is a cloud-based Jupyter environment, provided by Google, for people to run compute-intensive scripts on powerful machines.

Software and Hardware List

Chantar

With the following software and hardware list you can run all code files present in the book (Chapter 1-9).

Coftware required

E README.md			
	1-9	лирутег мотероок	Linux (Any)
	1-9	Python 3.5 and above, NumPy, Matplotlib, Anaconda (Optional)	Windows, Mac OS X, and Linux (Any)
	2-9	TensorFlow, tensorflow-gpu	Windows, Mac OS X, and Linux (Any)
	3	Scikit-Image	Windows, Mac OS X, and Linux (Any)
	4	TensorFlow Hub	Windows, Mac OS X, and Linux (Any)
	6	pydensecrf library	Windows, Mac OS X, and Linux (Any)
	7	Vispy, Plyfile	Windows, Mac OS X, and Linux (Any)
	8	opencv-python, tqdm, scikit-learn	Windows, Mac OS X, and Linux (Any)
	9	Android Studio, Cocoa Pods, Yarn	Windows, Mac OS X, and Linux (Any)

Table of Content

Chapter 1 - Computer Vision and Neural Networks
 1.1 - Building and Training a Neural Network from Scratch

- Chapter 2 TensorFlow Basics and Training a Model
 - 2.1 Training a model with Keras
- Chapter 3 Modern Neural Networks
 - 3.1 Discovering CNNs' Basic Operations
 - 3.2 Building and Training our First CNN with TensorFlow 2 and Keras
 - 3.3 Experimenting with Advanced Optimizers
 - 3.4 Applying Regularization Methods to CNNs
- Chapter 4 Influential Classification Tools
 - 4.1 Implementing ResNet from Scratch
 - 4.2 Reusing Models from Keras Applications
 - 4.3 Fetching Models from TensorFlow Hub
 - 4.4 Applying Transfer Learning
 - 4.5 (Appendix) Exploring ImageNet and Tiny-ImageNet
- Chapter 5 Object Detection Models
 - 5.1 Running YOLO inference
 - 5.2 (TBD) Training a YOLO model
- Chapter 6 Enhancing and Segmenting Images
 - 6.1 Discovering Auto-Encoders
 - 6.2 Denoising with Auto-Encoders
 - 6.3 Improving Image Quality with Deep Auto-Encoders (Super-Resolution)
 - 6.4 Preparing Data for Smart Car Applications
 - 6.5 Building and Training a FCN-8s Model for Semantic Segmentation
 - 6.6 Building and Training a U-Net Model for Object and Instance Segmentation
 - 6.6 Object and Instance Segmentation for Smart Cars with U-Net
- Chapter 7 Training on Complex and Scarce Datasets
 - 7.1 Setting up Efficient Input Pipelines with tf.data
 - 7.2 Generating and Parsing TFRecords
 - 7.3 Rendering Images from 3D Models
 - 7.4 Training a Segmentation Model on Synthetic Images
 - 7.5 Training a Simple Domain Adversarial Network
 - 7.6 Applying DANN to Train the Segmentation Model on Synthetic Data
 - 7.7 Generating Images with VAEs
 - 7.8 Generating Images with GANs
- Chapter 8 Video and Recurrent Neural Networks
 - 8.1 Action recognition using an LSTM
- Chapter 9 Optimizing Models and Deploying on Mobile Devices

- 9.1 Model profiling
- 9.2 Non-maximum suppression algorithm comparison
- 9.3 Training an emotion detection model and converting it for mobile devices
- iOS app
- Android app
- Tensorflow.js app



Get to Know the Authors

Benjamin Planche is a passionate PhD student at the University of Passau and Siemens Corporate Technology. He has been working in various research labs around the world (LIRIS in France, Mitsubishi Electric in Japan, and Siemens in Germany) in the fields of computer vision and deep learning for more than five years. Benjamin has a double master's degree with first-class honors from INSA-Lyon, France, and the University of Passau, Germany. His research efforts are focused on developing smarter visual systems with less data, targeting industrial applications. Benjamin also shares his knowledge and experience on online platforms, such as StackOverflow, or applies this knowledge to the creation of aesthetic demos.

Eliot Andres is a freelance deep learning and computer vision engineer. He has more than 3 years' experience in the field, applying his skills to a variety of industries, such as banking, health, social media, and video streaming. Eliot has a double master's degree from École des Ponts and Télécom, Paris. His focus is industrialization: delivering value by applying new technologies to business problems. Eliot keeps his knowledge up to date by publishing articles on his blog and by building prototypes using the latest technologies.



Referencing

If you use the code samples in your study/work or want to cite the book, please use:

```
@book{Andres_Planche_HandsOnCVWithTF2,
  author = {Planche, Benjamin and Andres, Eliot},
  title = {Hands-On Computer Vision with TensorFlow 2},
  year = {2019},
  isbn = {978-1788830645},
  publisher = {Packt Publishing Ltd},
}
```

► Other Formats: (Click to View)

Errata

- Page 18: stand of the heart should be state-of-the-art
- Page 24: graphical processing unit should be graphics processing unit
- Page 55: before hand should be beforehand
- Page 76: indiacting should be indicating
- Page 90: depth dimensions into a single vector should be depth dimensions into a single dimension
- Page 178: bceause should be because
- Page 183: smaller than the input and target latent spaces should be smaller than the input and target spaces
- Page 214: cannot only should be can not only
- Page 254: Jupyter Notebooks should be Jupyter notebooks

Related products

- Mastering OpenCV 4 with Python [Packt] [Amazon]
- OpenCV 4 for Secret Agents Second Edition [Packt] [Amazon]

Suggestions and Feedback

Click here if you have any feedback or suggestions.

Releases

No releases published

Packages

No packages published

Contributors 7















Languages

Jupyter Notebook 98.3%

• Python 1.1%

Java 0.3%

• Swift 0.3%

JavaScript 0.0%

• Shell 0.0%