Click here to download the source code to this post

PYIMAGESEARCH

DEEP LEARNING (HTTPS://WWW.PYIMAGESEARCH.COM/CATEGORY/DEEP-LEARNING/)

KERAS AND TENSORFLOW (HTTPS://WWW.PYIMAGESEARCH.COM/CATEGORY/KERAS-AND-TENSORFLOW/)

TUTORIALS (HTTPS://WWW.PYIMAGESEARCH.COM/CATEGORY/TUTORIALS/)

Auto-Keras and AutoML: A Getting Started Guide

by **Adrian Rosebrock (https://www.pyimagesearch.com/author/adrian/)** *on* January 7, 2019

In this tutorial, you will learn how to use Auto-Keras, an open source alternative to Google's AutoML, for automated machine learning and deep learning.

Click here to download the source code to this post

When training a neural network on a dataset there are two primary objectives a deep learning practitioner is trying to optimize and balance:

1. **Defining a neural network architecture** that lends itself to the nature of the dataset
2. **Tuning a set of hyperparameters over many experiments** that will lead to a model with high accuracy and ability to generalize to data outside the training and testing sets. Typical hyperparameters that need to be tuned include the optimizer algorithm (SGD, Adam, etc.), learning rate and learning rate scheduling, and regularization, to name a few

Depending on the dataset and problem, it can take a deep learning expert upwards of *tens* to *hundreds* of experiments to find a balance between neural network architecture and hyperparameters.

These experiments can add up to *hundreds* to *thousands* of hours in GPU compute time.

And that's just for *experts* — what about non-deep learning experts?

Enter Auto-Keras and AutoML:

The end goal of both Auto-Keras and AutoML is to reduce the barrier to entry to performing machine learning and deep learning through the use of *automated* Neural Architecture Search (NAS) algorithms.

Auto-Keras and AutoML enable non-deep learning experts to train their own models

Sound too good to be true?

Click here to download the source code to this post

Well, maybe — but you'll need to read the rest of the post first to find out why.

To learn more about AutoML (and how to *automatically* train and tune a neural network with Auto-Keras), *just keep reading!*

Looking for the source code to this post?

JUMP RIGHT TO THE DOWNLOADS SECTION →

Auto-Keras and AutoML: A Getting Started Guide

In the first part of this blog post, we'll discuss **Automated Machine Learning (AutoML)** and **Neural Architecture Search (NAS)**, the algorithm that makes AutoML possible when applied to neural networks and deep learning.

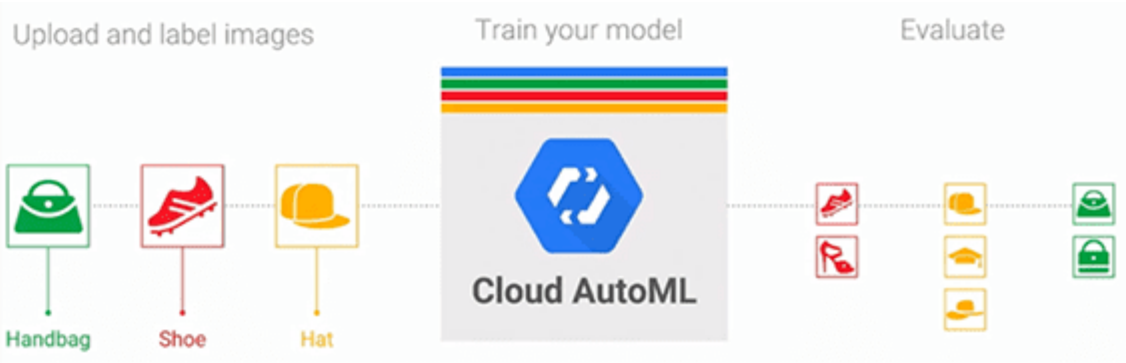
We'll also briefly discuss **Google's AutoML**, a suite of tools and libraries allowing programmers with limited machine learning expertise to train high accuracy models on their own data.

Of course, Google's AutoML is a proprietary algorithm (it's a bit on the expensive side of

Click here to download the source code to this post

I'll then show you **how to *automatically* train a network using Auto-Keras** as well as to evaluate it.

What is Automated Machine Learning (AutoML)?



(https://pyimagesearch.com/wp-content/uploads/2019/01/autokeras_automl.png)

Figure 1: Auto-Keras is an alternative to **Google's AutoML (https://cloud.google.com/automl/)**. These software projects can help you train models automatically with little intervention. They are great options for novice deep learning practitioners or to obtain a baseline to beat later on.

Outside of unsupervised learning (automatically learning patterns from unlabeled data), automated machine learning for non-experts is considered the "holy grail" of machine learning.

Imagine the ability to *automatically* create a machine learning model via:

1. Installing a library/using a web interface
2. Pointing the library/interface to your data
3. Automatically training a model on the data *without* having to tune the parameters/requiring an intimate understanding of the algorithms powering it

experience to automatically train neural networks on their own datasets.

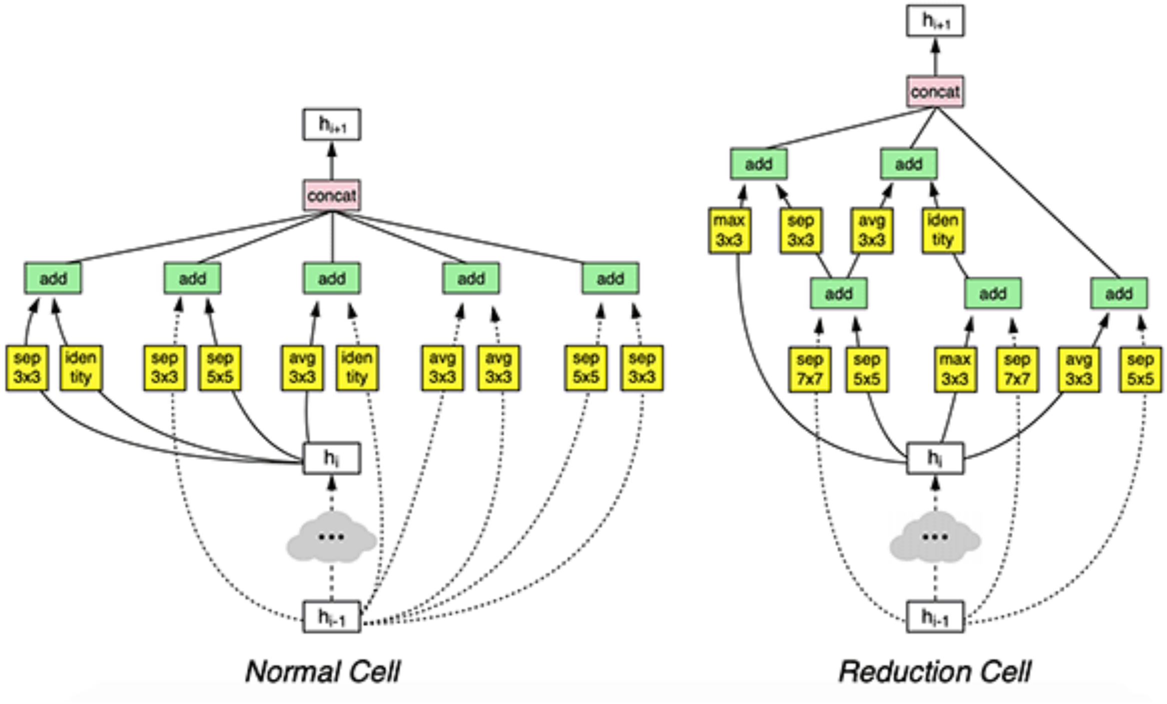
Click here to download the source code to this post

Under the hood Google's AutoML algorithms are iterative:

1. Training a network on a training set
2. Evaluating the network on a testing set
3. Modifying the neural network architecture
4. Tuning hyperparameters
5. Repeating the process

The programmer or engineer using AutoML doesn't need to define their own neural network architecture or tune the hyperparameters — AutoML is doing that for them automatically.

Neural Architecture Search (NAS) makes AutoML possible



(https://pyimagesearch.com/wp-content/uploads/2019/01/autokeras_cells.png)

Figure 2: Neural Architecture Search (NAS) produced a model summarized by these graphs when searching for the best CNN architecture for CIFAR-10 (source: Figure 4 of **Zoph et al. (https://arxiv.org/pdf/1707.07012.pdf)**)

Both Google's AutoML and Auto-Keras are powered by an algorithm called Neural Architecture Search (NAS).

Given your input dataset, a Neural Architecture Search algorithm will *automatically* search for the most optimal architecture and corresponding parameters.

Neural Architecture Search essentially replaces the deep learning engineer/practitioner with a set of algorithms that automatically tunes the model!

In the context of computer vision and image recognition, a Neural Architecture Search algorithm will:

1. Accept an input training dataset
2. Optimize and find *architectural building blocks* called "cells" — these cells are automatically learned and may look similar to inception, residual, or squeeze/fire micro-architectures
3. Continually train and search the "NAS search space" for more optimized cells

If the user of the AutoML system is an experienced deep learning practitioner then they may decide to:

1. Run the NAS on a significantly smaller subset of the training dataset

4. Train the network on the full training set using their own expertise and best practices

Such an approach is a *hybrid* between a fully automated machine learning solution and one that requires an expert deep learning practitioner — often this approach will lead to better accuracy than what the NAS finds on its own.

I would recommend reading ***Neural Architecture Search with Reinforcement Learning* (https://arxiv.org/abs/1611.01578)** (Zoph and Le, 2016) along with ***Learning Transferable Architectures for Scalable Image Recognition* (https://arxiv.org/abs/1707.07012)** (Zoph et al., 2017) for more details on how these algorithms work.

Auto-Keras: An open source alternative to Google's AutoML

(https://pyimagesearch.com/wp-content/uploads/2019/01/autokeras_logo.png)

Figure 3: The **Auto-Keras (https://autokeras.com/)** package was developed by the DATA Lab team at Texas A&M University. Auto-Keras is an open source alternative to **Google's AutoML (https://cloud.google.com/automl/)**.

The **Auto-Keras (https://autokeras.com/)** package, developed by the **DATA Lab team at Texas A&M University (http://people.tamu.edu/~guangzhou92/Data_Lab/)**, is an optimization to guide the network morphism for more efficient neural network search.

Click here to download the source code to this post

You can find the full details of the Auto-Keras framework in Jin et al.'s 2018 publication, ***Auto-Keras: Efficient Neural Architecture Search with Network Morphism* (https://arxiv.org/abs/1806.10282)**.

Project structure

Go ahead and grab the zip from the ***"Downloads"*** section of today's blog post.

From there you should unzip the file and navigate into it using your terminal.

Let's inspect today's project with the `tree` command:

→ **Launch Jupyter Notebook on Google Colab**

Auto-Keras and AutoML: A Getting Started Guide

```
$ tree --dirsfirst
.
├── output
│   ├── 14400.txt
│   ├── 28800.txt
│   ├── 3600.txt
│   ├── 43200.txt
│   ├── 7200.txt
│   └── 86400.txt
└── train_auto_keras.py

1 directory, 7 files
```

Today we're going to be reviewing a single Python script: `train_auto_keras.py` .

Since there will be a lot of output printed to the screen, I've opted to save our classification reports (generated with the help of scikit-learn's `classification_report` tool) as text files to disk. Inspecting the `output/` folder above you can see a handful of the reports that have been generated. Go ahead and print one to your terminal (

Click here to download the source code to this post

(https://pyimagesearch.com/wp-content/uploads/2019/01/autokeras_libraries.png)

Figure 4: The Auto-Keras package depends upon Python 3.6, TensorFlow, and Keras.

As the **Auto-Keras GitHub repository states (https://github.com/jhfjhfj1/autokeras)**, Auto-Keras is in a "pre-release" state — it is *not* an official release.

Secondly, Auto-Keras *requires* Python 3.6 and is *only* compatible with Python 3.6.

If you are using ***any other version of Python other than 3.6*** you will ***not** be able to utilize the Auto-Keras package*.

To check your Python version just use the following command:

→ **Launch Jupyter Notebook on Google Colab**

Auto-Keras and AutoML: A Getting Started Guide

```
$ python --version
```

Provided you have Python 3.6 you can install Auto-Keras using pip:

```
3. | $ pip install autokeras
```

Click here to download the source code to this post

If you have any issues installing or utilizing Auto-Keras make sure you post on the official **Auto-Keras GitHub Issues page (https://github.com/jhfjhfj1/autokeras/issues)** where the authors will be able to help you.

Implementing our training script with Auto-Keras

Let's go ahead and implement our training script using Auto-Keras. Open up the `train_auto_keras.py` file and insert the following code:

→ **Launch Jupyter Notebook on Google Colab**

Auto-Keras and AutoML: A Getting Started Guide

```
1. | # import the necessary packages
2. | from sklearn.metrics import classification_report
3. | from keras.datasets import cifar10
4. | import autokeras as ak
5. | import os
6. |
7. | def main():
8. |    # initialize the output directory
9. |    OUTPUT_PATH = "output"
```

To begin, we import necessary packages on **Lines 2-5**:

- As previously mentioned, we'll be using scikit-learn's `classification_report` to calculate statistics which we'll save in our output files.
- We're going to use the **CIFAR-10 Dataset (https://www.cs.toronto.edu/~kriz/cifar.html)**, conveniently built into `keras.datasets` .
- Then comes our most notable import, `autokeras` , which I've imported as `ak` for shorthand.

Let's define the `main` function for our script on **Line 7**. We're required to wrap our code in a `main` function due to how Auto-Keras and TensorFlow handle threading. See **this GitHub issue thread (https://github.com/jhfjhfj1/autokeras/issues/311)** for more details.

Click here to download the source code to this post

Our base `OUTPUT_PATH` is defined on **Line 9**.

Now let's initialize a list of training times for Auto-Keras:

→ **Launch Jupyter Notebook on Google Colab**

Auto-Keras and AutoML: A Getting Started Guide

```
11. |     # initialize the list of training times that we'll allow
12. |     # Auto-Keras to train for
13. |     TRAINING_TIMES = [
14. |         60 * 60,        # 1 hour
15. |         60 * 60 * 2,    # 2 hours
16. |         60 * 60 * 4,    # 4 hours
17. |         60 * 60 * 8,    # 8 hours
18. |         60 * 60 * 12,   # 12 hours
19. |         60 * 60 * 24,   # 24 hours
20. |     ]
```

Lines 13-20 define a set of `TRAINING_TIMES` , including `[1, 2, 4, 8, 12, 24]` hours. We'll be exploring the effect of longer training times on accuracy using Auto-Keras today.

Let's load the CIFAR-10 dataset and initialize class names:

→ **Launch Jupyter Notebook on Google Colab**

Auto-Keras and AutoML: A Getting Started Guide

```
22. |     # load the training and testing data, then scale it into the
23. |     # range [0, 1]
24. |     print("[INFO] loading CIFAR-10 data...")
25. |     ((trainX, trainY), (testX, testY)) = cifar10.load_data()
26. |     trainX = trainX.astype("float") / 255.0
27. |     testX = testX.astype("float") / 255.0
28. |
29. |     # initialize the label names for the CIFAR-10 dataset
```

Subsequently, we'll scale this data to the range of *[0, 1]* (**Lines 26 and 27**).

Click here to download the source code to this post

Our class `labelNames` are initialized on **Lines 30 and 31**. These 10 classes are included in CIFAR-10. Take note that order is important here.

And now let's begin looping over our `TRAINING_TIMES`, each time putting Auto-Keras to use:

→ **Launch Jupyter Notebook on Google Colab**

Auto-Keras and AutoML: A Getting Started Guide

```
33.     # loop over the number of seconds to allow the current Auto-Keras
34.     # model to train for
35.     for seconds in TRAINING_TIMES:
36.         # train our Auto-Keras model
37.         print("[INFO] training model for {} seconds max...".format(
38.             seconds))
39.         model = ak.ImageClassifier(verbose=True)
40.         model.fit(trainX, trainY, time_limit=seconds)
41.         model.final_fit(trainX, trainY, testX, testY, retrain=True)
42.
43.         # evaluate the Auto-Keras model
44.         score = model.evaluate(testX, testY)
45.         predictions = model.predict(testX)
46.         report = classification_report(testY, predictions,
47.             target_names=labelNames)
48.
49.         # write the report to disk
50.         p = os.path.sep.join(OUTPUT_PATH, "{}.txt".format(seconds))
51.         f = open(p, "w")
52.         f.write(report)
53.         f.write("\nscore: {}".format(score))
54.         f.close()
```

The code block above is the heart of today's script. On **Line 35** we've defined a loop over each of our `TRAINING_TIMES`, where we:

- Initialize our `model` (`ak.ImageClassifier`) and allow training to start (**Lines 39 and 40**). Notice that we didn't instantiate an object for a particular CNN class as we have in previous tutorials such as **this one (https://pyimagesearch.com/2018/04**

- Once the time limit has been reached, take the best model and parameters Auto-Keras has found + re-train the model (**Line 41**).

Click here to download the source code to this post

- Evaluate and construct the classification report (**Lines 44-47**).
- Write the classification report along with the accuracy score to disk so we can evaluate the effect of longer training times (**Lines 50-54**).

We'll repeat this process for each of our TRAINING_TIMES .

Finally, we'll check for and start the main thread of execution:

→ **Launch Jupyter Notebook on Google Colab**

Auto-Keras and AutoML: A Getting Started Guide

```
56. # if this is the main thread of execution then start the process (our
57. # code must be wrapped like this to avoid threading issues with
58. # TensorFlow)
59. if __name__ == "__main__":
60.     main()
```

Here we've checked to ensure that this is the main thread of execution and then the main function.

Just 60 lines later, we're done writing our Auto-Keras with CIFAR-10 example script. But we're not done yet…

Training a neural network with Auto-Keras

Let's go ahead and train our neural network using Auto-Keras.

Make sure you use the ***"Downloads"*** section of this tutorial to download the source code.

Launch Jupyter Notebook on Google Colab

Auto-Keras and AutoML: A Getting Started Guide

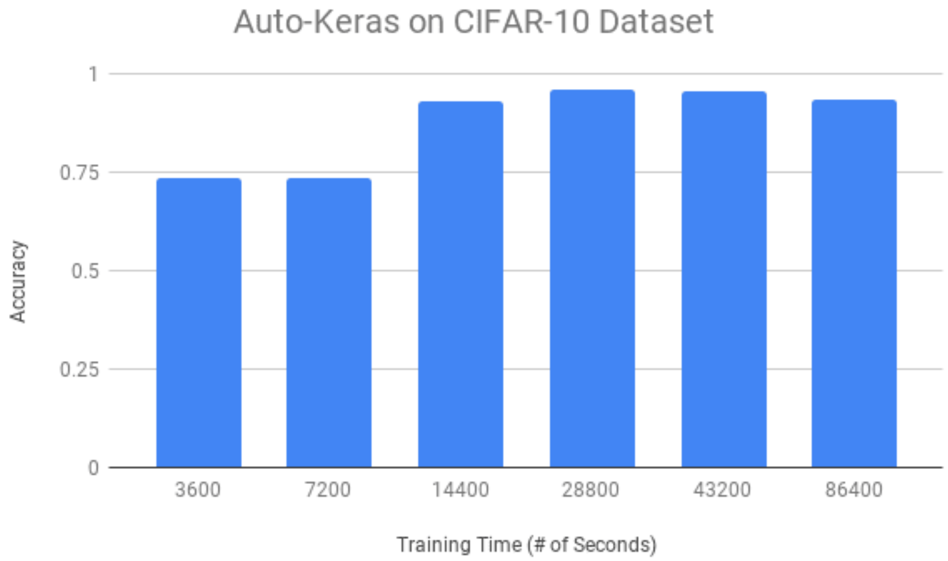
Click here to download the source code to this post

```
$ python train_auto_keras.py
[INFO] training model for 3600 seconds max...
Preprocessing the images.
Preprocessing finished.

Initializing search.
Initialization finished.


+----------------------------------------------+
|               Training model 0               |
+----------------------------------------------+
Using TensorFlow backend.

No loss decrease after 5 epochs.


Saving model.
+--------------------------------------------------------------------------+
|        Model ID        |          Loss          |      Metric Value      |
+--------------------------------------------------------------------------+
|           0            |   4.816269397735596    |         0.5852         |
+--------------------------------------------------------------------------+


+----------------------------------------------+
|               Training model 1               |
+----------------------------------------------+
Using TensorFlow backend.
Epoch-14, Current Metric - 0.83:  28%|███████▋                 | 110/387 [01:02<02:46,  1.67 batch/s]Time is out.
[INFO] training model for 86400 seconds max...
Preprocessing the images.
Preprocessing finished.

Initializing search.
Initialization finished.


+----------------------------------------------+
|               Training model 0               |
+----------------------------------------------+
Using TensorFlow backend.

No loss decrease after 5 epochs.
...
+----------------------------------------------+
|              Training model 21               |
+--------------------------------------------------------------------------+
|    Father Model ID     |                 Added Operation                 |
+--------------------------------------------------------------------------+
|                        |             to_deeper_model 16 ReLU             |
|           16           |              to_wider_model 16 64               |
+--------------------------------------------------------------------------+

Saving model.
+--------------------------------------------------------------------------+
|        Model ID        |          Loss          |      Metric Value      |
+--------------------------------------------------------------------------+
|           21           |   0.8843476831912994   |   0.9316000000000001   |
+--------------------------------------------------------------------------+


+--------------------------------------------------+
|               Training model 22                  |
+--------------------------------------------------+
Using TensorFlow backend.
Epoch-3, Current Metric - 0.9:  80%|████████████████████▌     | 310/387 [03:50<00:58,  1.31 batch/s]Time is out.

No loss decrease after 30 epochs.
```

Here you can see that our script is instructing Auto-Keras to perform six sets of experiments.

The total training time, including the time limits + the time to re-fit the model, was **a little over 3 days** on an NVIDIA K80 GPU.

Auto-Keras results



(https://pyimagesearch.com/wp-content/uploads/2019/01/autokeras_results.png)

Click here to download the source code to this post

Figure 5: Using Auto-Keras usually is a very time-consuming process. Training with Auto-Keras produces the best models for CIFAR-10 in the 8-12 hour range. Past that, Auto-Keras is not able to optimize further.

In **Figure 5** above you can see the effect of the amount of training time (*x*-axis) on overall accuracy (*y*-axis) using Auto-Keras.

Lower training times, namely 1 and 2 hours, lead to ~73% accuracy. Once we train for 4 hours we are able to achieve up to 93% accuracy.

The best accuracy we obtain is in the 8-12 range where we achieve **95% accuracy**.

Training for longer than 8-12 hours does not increase our accuracy, implying that we have reached a saturation point and Auto-Keras is not able to optimize further.

Are Auto-Keras and AutoML worth it?

(https://pyimagesearch.com/wp-content/uploads/2019/01/autokeras_worthit.png)

Click here to download the source code to this post

Outside of unsupervised learning (automatically learning patterns from unlabeled data), automated machine learning for non-experts is considered the "holy grail" of machine learning.

Both Google's AutoML and the open source Auto-Keras package attempt to bring machine learning to the masses, even without significant technical experience.

While Auto-Keras worked reasonably well for CIFAR-10, **I ran a second set of experiments** using my previous post on **deep learning, medical imagery, and malaria detection (https://pyimagesearch.com/2018/12/03/deep-learning-and-medical-image-analysis-with-keras/)**.

In that **previous post, I obtained 97.1% accuracy** using a simplified ResNet architecture which took under one hour to train.

I then let Auto-Keras run for 24 hours on the same dataset — *the result was barely 96% accuracy*, less than my hand-defined architecture.

Both Google's AutoML and Auto-Keras are great steps forward; however, automated machine learning is *nowhere near solved*.

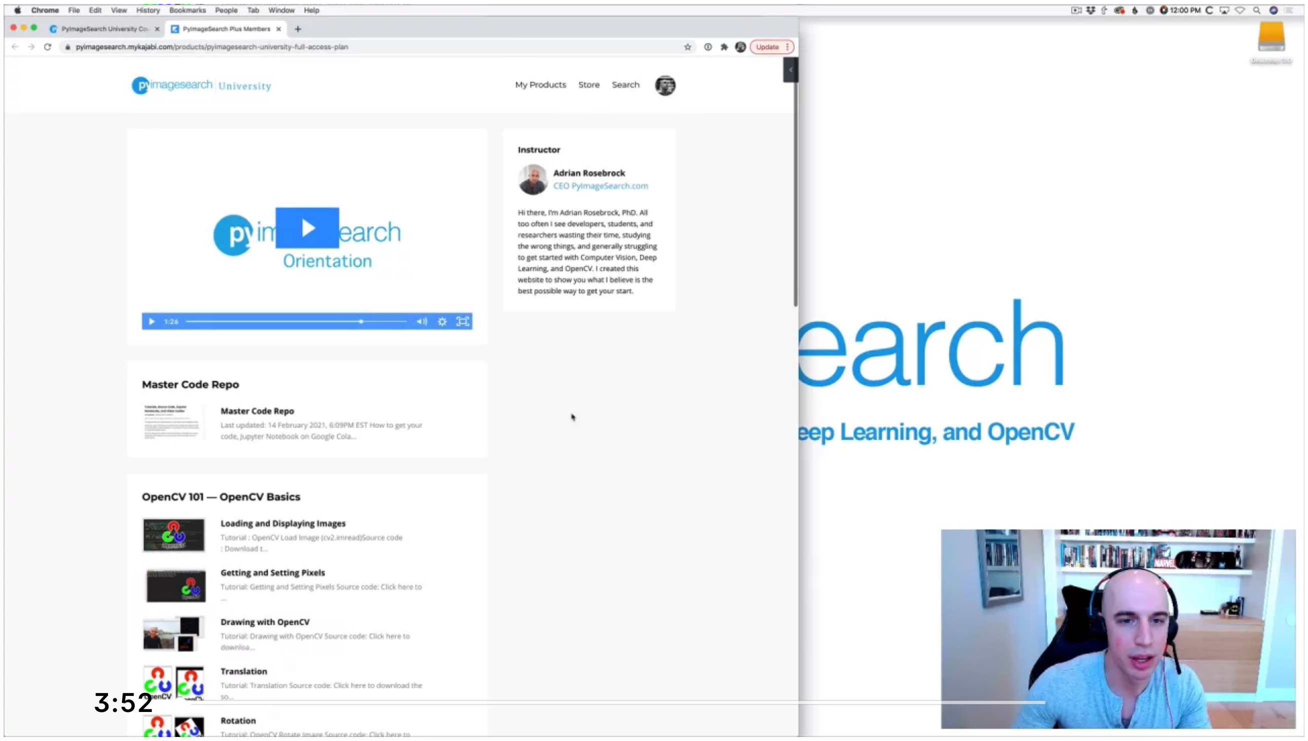
Automatic machine learning (currently) does not beat having expertise in deep learning — domain expertise, specifically in the data you are working with, is *absolutely critical* to obtain a higher accuracy model.

My suggestion is to invest in your own knowledge (https://pyimagesearch.com/deep-learning-computer-vision-python-book/), don't rely on automated algorithms.

To be a successful deep learning practitioner and engineer you need to bring the right tool to the job. Use AutoML and Auto-Keras for what they are, *tools*, and then continue

What's next? I recommend PyImageSearch University (https://www.pyimagesearch.com/pyimagesearch-university/?utm_source=blogPost&utm_medium=bottomBanner&utm_campaign=What%27s%20next%3F%20I%20recommend).

Click here to download the source code to this post


Chrome File Edit View History Bookmarks People Tab Window Help
PyImageSearch University Co
PyImageSearch Plus Members
pyimagesearch.mykajabi.com/products/pyimagesearch-university-full-access-plan
Update
pyimagesearch University
My Products
Store
Search
pyimagesearch
Orientation
1:26
earch
eep Learning, and OpenCV
3:52

Instructor

Adrian Rosebrock
CEO PyImageSearch.com

Hi there, I'm Adrian Rosebrock, PhD. All too often I see developers, students, and researchers wasting their time, studying the wrong things, and generally struggling to get started with Computer Vision, Deep Learning, and OpenCV. I created this website to show you what I believe is the best possible way to get your start.

Master Code Repo

Master Code Repo
Last updated: 14 February 2021, 6:09PM EST How to get your code, Jupyter Notebook on Google Cola...

OpenCV 101 — OpenCV Basics

Loading and Displaying Images
Tutorial : OpenCV Load Image (cv2.imread)Source code : Download t...

Getting and Setting Pixels
Tutorial: Getting and Setting Pixels Source code: Click here to ...

Drawing with OpenCV
Tutorial: Drawing with OpenCV Source code: Click here to downloa...

Translation
Tutorial: Translation Source code: Click here to download the so...

Rotation
Tutorial: OpenCV Rotate Image Source code: Click here to

Course information:

28 total classes • 39h 44m video • Last updated: 10/2021

★★★★★ 4.84 (128 Ratings) • 3,000+ Students Enrolled

Do you think learning computer vision and deep learning has to be time-consuming, overwhelming, and complicated? Or has to involve complex mathematics and equations? Or requires a degree in computer science?

Click here to download the source code to this post

That’s *not* the case.

All you need to master computer vision and deep learning is for someone to explain things to you in *simple, intuitive* terms. *And that’s exactly what I do*. My mission is to change education and how complex Artificial Intelligence topics are taught.

If you're serious about learning computer vision, your next stop should be PyImageSearch University, the most comprehensive computer vision, deep learning, and OpenCV course online today. Here you’ll learn how to *successfully* and *confidently* apply computer vision to your work, research, and projects. Join me in computer vision mastery.

Inside PyImageSearch University you'll find:

✓ **28 courses** on essential computer vision, deep learning, and OpenCV topics

✓ 28 Certificates of Completion

✓ **39h 44m** on-demand video

✓ **Brand new courses released *every month***, ensuring you can keep up with state-of-the-art techniques

✓ Access to **centralized code repos for *all* 400+ tutorials** on PyImageSearch

Click here to download the source code to this post

✓ **Easy one-click downloads** for code, datasets, pre-trained models, etc.

✓ Access on mobile, laptop, desktop, etc.

CLICK HERE TO JOIN PYIMAGESEARCH UNIVERSITY (HTTPS://WWW.PYIMAGESEARCH.COM/PYIMAGESEARCH-UNIVERSITY/?UTM_SOURCE=BLOGPOST&UTM_MEDIUM=BOTTOMBANNER&UTM_CAMPAIGN=WHAT%27S%20NEXT%3F%20I%20RECOMMEND)

Summary

In today's blog post, we discussed Auto-Keras and AutoML, a set of tools and libraries to perform automated machine learning and deep learning.

The end goal of both Auto-Keras and AutoML is to reduce the barrier to entry to performing machine learning and deep learning through the use of Neural Architecture Search (NAS) algorithms.

NAS algorithms, the backbone of Auto-Keras and AutoML, will *automatically*:

1. Define and optimize a neural network architecture
2. Tune the hyperparameters to the model

The primary benefits include:

- Getting up and running quickly with either a GUI interface or a simple API
- A potentially state-of-the-art performance with little effort

Of course, there is a price to be paid — two prices in fact.

First, Google's AutoML is expensive, approximately $20/hour.

To save funds you could go with Auto-Keras, an open source alternative to Google's AutoML, *but you still need to pay for GPU compute time*.

Replacing an actual deep learning expert with a NAS algorithm will require many hours of computing to search for optimal parameters.

While we achieved a high accuracy model for CIFAR-10 (~96% accuracy), when I applied Auto-Keras to my previous post on **medical deep learning and malaria prediction (https://pyimagesearch.com/2018/12/03/deep-learning-and-medical-image-analysis-with-keras/)**, **Auto-Keras only achieved 96.1% accuracy, a full percentage point lower than my 97% accuracy (and Auto-Keras required *2,300% more compute time!*)**

While Auto-Keras and AutoML may be a step in the right direction in terms of automated machine learning and deep learning, there is still quite a bit of work to be done in this area.

There is no silver bullet for solving machine learning/deep learning with off-the-shelf algorithms. **Instead, I recommend you invest in yourself as a deep learning practitioner and engineer (https://pyimagesearch.com/deep-learning-computer-vision-python-book/).**

The skills you learn today and tomorrow will pay off *tremendously* in the future.

Click here to download the source code to this post

Download the Source Code and FREE 17-page Resource Guide

Enter your email address below to get a .zip of the code and a **FREE 17-page Resource Guide on Computer Vision, OpenCV, and Deep Learning.** Inside you'll find my hand-picked tutorials, books, courses, and libraries to help you master CV and DL!

About the Author

Hi there, I’m Adrian Rosebrock, PhD. All too often I see developers, students, and researchers wasting their time, studying the wrong things, and generally struggling to get started with Computer Vision, Deep Learning, and OpenCV. I created this website to show you what I believe is the best possible way to get your start.

Previous Article:

Keras Conv2D and Convolutional Layers

Click here to download the source code to this post
(https://www.pyimagesearch.com/2019/01/14/machine-learning-in-python/)

39 responses to: Auto-Keras and AutoML: A Getting Started Guide

golemu

January 7, 2019 at 11:50 am (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-495012)

Great tutorial. This post is a great way to start my year! I look forward to augmenting auto-keras to my research and see if it helps improves my models. I look forward to gathering great insight from auto-keras/autoML.

As always, you provide valuable insight to machine learning and deep learning. Looking forward to reading more! Enjoy your 2019!!

–golemu

Adrian Rosebrock

January 7, 2019 at 12:23 pm (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-495015)

Thanks so much, Golemu!

No problem. But I did have a question.

Will Auto-keras work for multi-task learning?

Adrian Rosebrock

January 7, 2019 at 2:22 pm (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-495030)

As far as I understand, no. Most applications with Auto-Keras are used for image and text.

golemu

January 11, 2019 at 2:24 am (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-495813)

Is there a fit_generator in auto-keras? My dataset is too large to fit into memory all at once. Any suggestions?

Adrian Rosebrock

January 11, 2019 at 9:28 am (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-495909)

Good question. As far as I know there is not a fit_generator function in Auto-Keras. I would suggest reaching out to the Auto-Keras developers on GitHub.

Click here to download the source code to this post

Hi Adrian,

Thanks for another great tutorial!!

How do AutoML and Auto-Keras compare to AutoWeka? I've used AutoWeka for a couple of projects and it works reasonably well.

Adrian Rosebrock

January 7, 2019 at 12:23 pm (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-495014)

I've never used AutoWeka before, but the general idea behind AutoWeka is that it's not *just* neural networks and deep learning — it's *all* the algorithms implemented inside Weka along with their associated hyperparameters. AutoWeka seeks to not only search among algorithms but their hyperparameters as well.

brian penn (http://www.legacycorescanning.com)

January 8, 2019 at 6:08 pm (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-495192)

Yes, I had one dataset that AutoWeka recommended an SVM solution. I actually like this idea b/c neural networks don't always provide optimal solutions.

Excellent tutorial with tangible figures! Many thanks for letting us see what is under the hood indeed.

Click here to download the source code to this post

It makes me think that AutoML/AutoKeras is like a car with automatic transmission. It consumes more and is not currently good enough for performance challenges but good for general purpose soft-driving mostly. In formula 1 race (challenge in a short time), you would not use a car with an automatic transmission but a manual transmission car (hyperparameter tuning) with a F1 Pilot (Deep Learning Expert).

Having said that I believe that NAS will evolve in time and will be pervasively used across all domains in parallel to ongoing computing power increases. Just like cars with automatic transmissions override those with manual transmissions today …

Adrian Rosebrock

January 8, 2019 at 6:47 am (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-495119)

Great analogy, Cenk!

David Bonn

January 7, 2019 at 4:02 pm (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-495043)

Adrian — this is a very interesting tutorial. I agree with your evaluation that an experienced practitioner in the arts of Deep Learning can easily produce better results in less time, but systems such as this could still be valuable for someone starting down the very steep deep learning curve.

Adrian Rosebrock

January 8, 2019 at 6:45 am (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-495116)

Click here to download the source code to this post

Thanks David!

Mercy Ranjit

January 7, 2019 at 9:06 pm (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-495063)

Hey Adrian,

I greatly benefit from your posts. I would also like to thank you for the books you had released on deep learning(I purchased all the bundles), I used them for my research course work preparation and got up-to speed in no-time. Thank you so much for writing these amazing books.

I wanted to reference the below blog which gives a comparison on all the options available including the one from Microsoft.

https://medium.com/@santiagof/auto-is-the-new-black-google-automl-microsoft-automated-ml-autokeras-and-auto-sklearn-80d1d3c3005c (https://medium.com/@santiagof/auto-is-the-new-black-google-automl-microsoft-automated-ml-autokeras-and-auto-sklearn-80d1d3c3005c)

Thanks,
Mercy

Thanks Mercy, I really appreciate the kind words. Thank you for making PyImageSearch possible by going through my Deep Learning book. I’m so happy to hear that it helped you with your research 🙂

Also, thanks so much for sharing the AutoML comparisons!

Xu Zhang

January 8, 2019 at 7:27 pm (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-495196)

A great article again. Thank you very much.

For each iteration, is it true that the algorithm decides how many epochs it will run?

Adrian Rosebrock

January 11, 2019 at 9:58 am (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-495944)

Correct. The algorithm trains until either accuracy/loss is not improving or the training time has ran out.

Mohanad Abdulsattar

wish to meet you in person to talk more in details about my ideas and imagination in this field. I was really blessed and lucky to find such a smart and generous person like you.

Click here to download the source code to this post

Adrian Rosebrock

January 11, 2019 at 9:55 am (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-495941)

Thank you for the kind words, Mohanad 🙂

kalyanramu

January 9, 2019 at 11:11 am (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-495271)

Is it possible to run AutoML in google colab? Google Colab is free

Basit Aslam

January 14, 2019 at 12:57 am (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-496250)

Dear Adrian Rosebrock,
Really glad to find your blogs which are totally totally gem and upscale for the OpenCv learner. I never worked on such projects in my life and this is my first time that i am going to work on it. I want to start it from the scratch till my final project but

- Project Discription

Be a PHD student my supervisor gave me a task to run a camera and put some chart in front of it. mark some point on the image and according to that with the equal distance from the camera align it parallel with the help of motor to align it.
if you can provide me the email address of yours then i can share some of the image scenarios on that behalf i need to align the camera. That will be much helpful for me.
Thank you so much dear once again.

Adrian Rosebrock
January 16, 2019 at 10:02 am (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-496668)

The best place to reach me is via the **PyImageSearch contact form (https://www.pyimagesearch.com/contact/)** but if you’re having a problem with your final project I would suggest chatting with your supervisor instead — I don’t know the particulars of your project and your supervisor will be better equipped to give you targeted advice for your project. Best of luck with it!

rahman
January 23, 2019 at 11:30 pm (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-497670)

Dear Basit Aslam,
I can understand how do you feel and how hard to do a project you don’t have any experience yet. I was struggling to finish my project on the topic of deep learning which was something not only new but strange for me. Google helps a “bit” but not sufficient to help me out. Thank God I found this blog and invest my future by buying the most amazing book to start deep learning and python. So, I really recommend you to have a look the book and invest in your future:

Trust me, you will thank me later 😊

note: sorry to promote your book Adrian, I only want to share the wonderful knowledge you've done in your amazing book

Adrian Rosebrock

January 25, 2019 at 7:13 am (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-497849)

Thank you Rahman, I really appreciate your kind words — I'm also so happy the book was able to help you with your project, congrats!

rahman

January 23, 2019 at 1:30 am (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-497557)

As always, a great, nice and clean tutorial from the expert one.
Adrian, would you please provide in your next post all about pre-trained network in CNN including what is that, why we need it, and the most important how to use it etc.
I truly believe that your 'beginner' readers will be very happy to learn about the pre-trained CNN from your nice tutorial.
wish you all the best Adrian

Adrian Rosebrock

January 25, 2019 at 7:31 am (https://www.pyimagesearch.com/2019/01/07/auto-keras-

python-book/)? The book is designed to help you understand CNNs, pre-trained CNNs, and train your CNNs from scratch on custom datasets. Even if you are beginner the book will absolutely help you get up to speed. Give it a look, I believe it will really help you.

dergham

March 20, 2019 at 10:06 am (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-508080)

thank you for your very nice tutorial
I have a question
since I am a trader I want to know if auto keras can be used for stock price prediction
most (or all) examples I see Initialize the model (ak.ImageClassifier)
is there other models to deal with numeric data or time series ?

Adrian Rosebrock

March 22, 2019 at 8:57 am (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-508536)

Sorry, I have only used AutoKeras for work in image classification. For non-image questions you should ask the AutoKeras creators.

Click here to download the source code to this post

Nice tutorial. Do you have tutorials where you use your own data?

Adrian Rosebrock

April 18, 2019 at 6:51 am (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-513426)

I have a TON of tutorials that cover how to train your own model using custom data. **This one (https://www.pyimagesearch.com/2018/04/16/keras-and-convolutional-neural-networks-cnns/)** would be a good start as well as **Deep Learning for Computer Vision with Python. (https://www.pyimagesearch.com/deep-learning-computer-vision-python-book/)**

At

May 2, 2019 at 7:57 am (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-515720)

Do we have to just split raw data and feed it to the AutoML or we also need to do feature engineering?

Adrian Rosebrock

May 8, 2019 at 1:44 pm (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-516628)

That really depends on your exact project and dataset. I would run your data as-is

Click here to download the source code to this post

Joseph

July 20, 2019 at 8:54 pm (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-526553)

Hey Adrian! Wonderful article thanks for sharing. I really want to get into deep learning I was wondering if there was any software a non technical person could use to easily train and learn from data? I don't know coding and as a single father of two working for an investment bank, it's impossible for me to find the time.

Walid

September 27, 2019 at 10:39 am (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-554932)

Thanks a lot for the great post, I am getting this message over and over

"Current model size is too big. Discontinuing training this model to search for other models".

it is strange as no one posted in comments havimng such issue.

Can you please help?

Walid

Whoa, that is very strange. I haven't encountered that error before. How long did you let the script run for?

Hi Adrian

Thank a million
one question, why do you think we need to retrain the best model we got as instructed in line #41?

All the best,

Walid

Adrian Rosebrock

October 3, 2019 at 12:12 pm (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-557416)

Because the previous call to ".fit" will return whatever the current model is and not the best model.

Walid Ahmed

March 30, 2020 at 10:39 am (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-768398)

Great Article
One question, can I choose which layers can be or can not be in my architecture?
like adding 3d conv layer to structure?

Adrian Rosebrock

April 1, 2020 at 9:30 am (https://www.pyimagesearch.com/2019/01/07/auto-keras-and-automl-a-getting-started-guide/#comment-769536)

Click here to download the source code to this post

I don’t believe so but that would be a good question for the Auto-Keras developers.

Comment section

Hey, Adrian Rosebrock here, author and creator of PyImageSearch. While I love hearing from readers, a couple years ago I made the tough decision to no longer offer 1:1 help over blog post comments.

At the time I was receiving 200+ emails per day and another 100+ blog post comments. I simply did not have the time to moderate and respond to them all, and the sheer volume of requests was taking a toll on me.

Instead, my goal is to *do the most good* for the computer vision, deep learning, and OpenCV community at large by focusing my time on authoring high-quality blog posts, tutorials, and books/courses.

If you need help learning computer vision and deep learning, I suggest you refer to my full catalog of books and courses (https://www.pyimagesearch.com/books-and-courses/) — they have helped tens of thousands of developers, Click here to browse my full catalog. (https://www.pyimagesearch.com/books-and-courses/)

Click here to download the source code to this post

Similar articles

DEEP LEARNING MXNET RESOURCES

How to plot accuracy and loss with mxnet

December 25, 2017

(https://www.pyimagesearch.com/2017/12/25/plot-accuracy-loss-mxnet/)

→

IMAGE PROCESSING TUTORIALS

How-To: OpenCV Load an Image

June 2, 2014

(https://www.pyimagesearch.com/2014/06/02/opencv-load-image/)

→

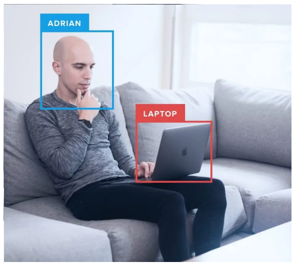
Click here to download the source code to this post

DEEP LEARNING OPENCV TUTORIALS TUTORIALS

How to use OpenCV's "dnn" module with NVIDIA GPUs, CUDA, and cuDNN

February 3, 2020

(https://www.pyimagesearch.com/2020/02/03/how-to-use-opencvs-dnn-module-with-nvidia-gpus-cuda-and-cudnn/)



You can learn Computer Vision, Deep Learning, and OpenCV.

Get your FREE 17 page Computer Vision, OpenCV, and Deep Learning Resource Guide PDF. Inside you'll find our hand-picked tutorials, books, courses, and libraries to help you master CV and DL.

Topics

Deep Learning (https://www.pyimagesearch.com/category

Machine Learning and Computer Vision (https://www.pyimagesearch.com/category/machine-learning-2/)

/deep-learning-2/)

Dlib Library (https://www.pyimagesearch.com /category/dlib/)

Click here to download the source code to this post

Embedded/IoT and Computer Vision (https://www.pyimagesearch.com/category /embedded/)

Face Applications (https://www.pyimagesearch.com/category /faces/)

Image Processing (https://www.pyimagesearch.com/category /image-processing/)

Interviews (https://www.pyimagesearch.com /category/interviews/)

Keras (https://www.pyimagesearch.com /category/keras/)

Medical Computer Vision (https://www.pyimagesearch.com/category /medical/)

Optical Character Recognition (OCR) (https://www.pyimagesearch.com/category /optical-character-recognition-ocr/)

Object Detection (https://www.pyimagesearch.com/category /object-detection/)

Object Tracking (https://www.pyimagesearch.com/category /object-tracking/)

OpenCV Tutorials (https://www.pyimagesearch.com/category /opencv/)

Raspberry Pi (https://www.pyimagesearch.com /category/raspberry-pi/)

Books & Courses

FREE CV, DL, and OpenCV Crash Course (https://www.pyimagesearch.com/free-opencv-computer-vision-deep-learning-crash-course/)

Practical Python and OpenCV (https://www.pyimagesearch.com/practical-python-opencv/)

Deep Learning for Computer Vision with Python (https://www.pyimagesearch.com/deep-learning-computer-vision-python-book/)

PyImageSearch Gurus Course (https://www.pyimagesearch.com /pyimagesearch-gurus/)

Raspberry Pi for Computer Vision

PyImageSearch

Get Started (https://www.pyimagesearch.com /start-here/)

OpenCV Install Guides (https://www.pyimagesearch.com/opencv-tutorials-resources-guides/)

About (https://www.pyimagesearch.com/about/)

FAQ (https://www.pyimagesearch.com/faqs/)

Blog (https://www.pyimagesearch.com/topics/)

Contact (https://www.pyimagesearch.com /contact/)

Privacy Policy (https://www.pyimagesearch.com /privacy-policy/)

(https://www.pyimagesearch.com/raspberry-pi-for-computer-vision/) **Click here to download the source code to this post**

(https://www.facebook.com/pyimagesearch) **(https://twitter.com/PyImageSearch)** **(http://www.linkedin.com/pub/adrian-rosebrock/2a/873/59b)** **(https://www.youtube.com/channel/UCoQK7OVcIVy-nV4m-SMCk_Q/videos)**